Adama Science and Technology University

Office of Vice President for Research and Technology Transfer

Proceeding of the 2nd Deep Learning Indaba-X Ethiopia Conference 2021

"*Strengthening Awareness and Application of Machine Learning and Artificial Intelligence in Ethiopia*"

January 27 – 29, 2022, Adama, Ethiopia

Disclaimer

Adama Science and Technology University is not responsible for the contents reflected in the articles published in the proceedings of this International Research Conference. The contents of this document are solely the responsibility of the authors.

Inquiries should address to:

Office of Vice President for Research and Technology Transfer
Adama Science and Technology University
P.O.Box 1888, Adama, Ethiopia
Tel: +251-22-110-0017

Members of Organizing Committee

Dr. Alemu Disassa (Chairperson)	Mr. Tadesse Hailu (Member)
Dr. Fedlu Kedir (Secretary)	Dr. Bedasa Abdisa (Member)
Dr. Tilahun Melak (Member)	Dr. Alemgena Belete (Member)
Ms. Genet Shanko (Member)	Dr. Belay Birhanu (Member)
Mr. Kuma Waqtola (Member)	Mr. Mitiku Kinfe (Member)
Mr. Bejiga Yadecha (Member)	Dr. Kassaye Gutema (Member)
Dr. Yadeta Chimdessa (Member)	Dr. Worku Jifara (Member)
Dr. Milkias Berehanu (Member)	Mr. Endris Mohammed (Member)
Dr. Dereje Teklu (Member)	Mr. Tadele Kebebe (Member)
Dr. Demissie Jobir (Member)	Mr. Elias Kebede (Member)
Dr. Tefera Terefe (Member)	Mr. Amare Derbe (Member)

Table of Contents

Preface

The transformation of a nation can basically be achieved through the advancement of science and technology. Ethiopia has long recognized the role of science and technology in bringing about sustainable development. The country has envisioned transforming itself into a middle-income country in 2025. To this end the country has exerted relentless efforts to materialize science and technology in the country. Thus, it has made science and technology the pillar of its top priorities for transformation of the economy.

As one of the universities mandate to spearhead the transformation process, Adama Science and Technology University (ASTU) is looking forward to excel in science and technology. Its goal is to develop highly qualified, capable, competent, and innovative human resource in the field of science and technology so as to transfer relevant scientific knowledge and skills required for nation building. The university also committed to conduct need based problem solving researches for alleviating the problems of the region and the country at large. To this end the university is working in collaboration with industries in its vicinity whereby its staff members are contributing a great deal in alleviating problems. Moreover, ASTU has set centers of excellence as a platform where academia can meet stakeholders.

ASTU's development into a full-fledged science and technology university has helped it to forge strong linkage, cooperation, and partnership with various national and international universities, development sectors, stake-holders, and relevant personalities. To showcase its all-round efforts, ASTU has organized the 2nd Deep Learning Indaba-X Ethiopia Conference 2021 on "*Strengthening Awareness and Application of Machine Learning and Artificial Intelligence in Ethiopia*". This is a broad agenda that is seen as a part of the national plan of transformation of the country. Thus, this research confrence aims to further strengthen the contribution of ASTU in development endeavours of the country at large.

Message from Organizing Committee

Alemu Disassa (PhD)
Vice President for RTT, ASTU
Chair of the Organizing Committee

Honorable Guests, Dear Participants, on behalf of the organizing committee, I would like to welcome you all to the 2nd Deep Learning Indaba X Ethiopia Conference 2021 organized by ASTU in collaboration with Deep learning Indaba on the theme "***Strengthening Awareness and Application of Machine Learning and Artificial Intelligence in Ethiopia***". Since its establishment as full-fledged Science and Technology University, ASTU has been exerting tremendous effort to foster research culture among its staff. Over the last five years, it has initiated and conducted several problem driven applied researches in the area of applied sciences and engineering. Besides, ASTU has provided several community and consultancy services that has curbed development challenges of the country and led to new policy initiatives and generation of development insights. The establishment of eight centers of excellence and the construction state-of-the-art research park in ASTU also witnesses the commitment of the university towards nurturing its research and technology transfer endeavors in the future.

In the effort to disseminate its research outputs and create a platform which allows the academia in ASTU share scientific knowledge and thought with national and international scholars, ASTU had successfully held three international research symposiums so far. The 1st International Research Symposium was held in September 2012 on the theme "Sustainable Development through Science and Technology: Lessons from Emerging Economies", the 2nd in June 2017 on the theme "Ensuring Sustainable Development through Research in Science and Technology" while the 3rd one was held in May 2019 on the theme "Emerging Technologies and Energy for Sustainable Development". These three symposiums have helped a lot in increasing our national and international collaborations besides nurturing better research culture in our university.

The 2nd Deep Learning Indaba X Ethiopia Conference 2021 is also aimed at consolidating our national and international research collaboration and eventually helping ASTU achieve its vision of becoming a national hub for science and technology researches. To be specific, this conference is aimed at enhancing the awareness and application of Machine Learning and Artificial Intelligence in Ethiopia through creating a platform for national and international scholars to share scientific knowledge and skills in the aforementioned thematic area.

Nearly 200 participants, including distinguished researchers from renowned national and international universities, delegates from technology companies, research institutes, industries and relevant government offices are expected to take part in this conference. In this conference, 24 scientific papers are expected to be presented at the plenary and syndicate sessions. Besides, a technical training and panel discussion will be held with higher officials of ASTU on future collaboration. Thus, I am very much confident that the participants can learn a lot from the conference.

Finally, I would like to thank you all for accepting our invitation to share us your scientific knowledge and expertise. I wish you all a fruitful scientific sessions and very pleasant stay in Adama City.

Welcoming Address

Lemi Guta (PhD)
President of ASTU

Dear our Chief Guest – Your Excellency Engineer Worku Gachena - Director General of Artificial Intelligence and Board Chairperson of our University,

Dear Dr. Shumet Gizaw, Director General of the Information Network Security Agency,

Dear Dr. Dereje Engida, President of AASTU,

Dear Dr. Jemal and Dr. Abraham, Vice Presidents of AASTU,

Dear Vice Presidents of our University

Dear Keynote Speakers,

Dear Paper presenters,

Dear Researchers,

Dear Invited Guests,

Dear Invited Media

Dear viewers of this event through YouTube's and our ASTU Official Channel,

Ladies and Gentle Men,

A very warm Good Morning to you all!

Thank you to each and every one of you for being here with us to take part in our 2nd Deep Learning Indaba-X Ethiopia 2021 Conference.

I am very pleased to welcome you all to our university!

As a university, we are proud to be able to host the 2nd Deep Learning Indaba-X Ethiopia 2021 Conference today here at this wonderful place with all of you.

The Deep Learning Indaba-x is a movement that started in 2017 with the aim of strengthening machine learning in Africa. It is meant to build communities and to give training on Machine learning and Artificial intelligence in the African context.

The Deep Learning Indaba-X is the local version of the Deep Indaba Conference. In Deep Learning Indaba Ethiopia 2021, in line with the mission of Deep Learning Indaba, the general objective of this conference is to - strengthen awareness and application of Machine Learning and artificial intelligence in the industrial and academic community in Ethiopia.

Dear Engineer Worku Gachena-

Ladies and Gentlemen,

We are lucky to host the 2nd Deep Learning Indaba-X Ethiopia 2021 Conference here in ASTU. This is relevant in that Adama Science and Technology University (ASTU), as you all know is a special mission

driven science and technology university that is expected to produce qualified graduates in applied science and engineering. The university is expected to facilitate such platforms for the academic community with the mission of creating qualified professionals in engineering and applied sciences.

The 2nd Deep Learning Indaba-X 2021 conference is organized by ASTU in collaboration with Google and Deep learning Indaba 2020/2021 with the theme of "*Strengthening awareness and application of Machine Learning and Artificial Intelligence in Ethiopia*".

The conference will be conducted to improve awareness and enhance the application of Machine learning and Artificial Intelligence in Ethiopia's Academic, industrial and research sphere.

Additionally, the conference's aim is to promote international and national collaborations, creating platform for research community in ASTU and national participants to share best experiences and disseminate their research outputs to the end users.

In the conference, a total of 193 participants from national, international researchers, MSc and PhD students, industry, government and non-government Artificial Intelligence Companies and women in artificial intelligence team are going to participate. The conference intends to improve skill, knowledge and experience of academicians, practitioners and researchers' in the area of computing, engineering and applied natural science.

Dear Engineer Worku Gachena-

Ladies and Gentlemen,

In this conference distinguished researchers and scientists from renowned international universities will share their experiences and create collaboration with ASTU in various research thematic areas. This will create a big opportunity to PhD students and academic staff members in creating good network and sharing experience.

Artificial Intelligence (AI) is everywhere. Possibility is that you are using it in one way or the other and you don't even know about it. One of the popular applications of AI is Machine Learning (ML), in which computers, software, and devices perform via cognition (very similar to the human brain). The value of machine learning technology has been recognized by companies across several industries that deal with huge volumes of data. By leveraging insights obtained from this data, companies are able to work in an efficient manner to control costs as well as get an edge over their competitors.

The application of machine learning awards work efficiency and economic savings in the areas like Financial Services, agriculture, Manufacturing industry, E-commerce, E-Health, E- Governance, Transportation, Security and identification systems. The science and application of

AI and Machine learning can be assessed to be at its early stages in Ethiopia. In order for the country (Ethiopia) to benefit from the application of AI and Machine learning it is must to boost and encourage academicians, industry practitioners and politicians (policy makers) on the awareness and application of AI and Machine learning. Several African countries these days are trying their best to benefit from the science and application of AI and Machine learning.

The Deep Learning Indaba is an organization whose mission is to Strengthen Machine Learning and Artificial Intelligence in Africa. The organization has been working towards the goal of Africans should be active shapers and owners of these technological advances, instead of observers and receivers of the ongoing advances in Machine learning and artificial intelligence.

Ladies and Gentlemen,

The Deep Learning Indaba aims to address two principal concerns: African participation and contribution to the advances in artificial intelligence and machine learning, and diversity in these fields of science.

The implications of these overarching goals is the spreading of technical knowledge at the state-of-the art in the field; the opportunities for new research connections to be made and silos in the research community to be broken; the fostering of a better understanding of the variety of career paths in the field, especially those that are in abundance locally; and through new friendships, perspectives and backgrounds, taking the steps to realizing a more representative, inclusive and multicultural machine learning community.

I would like to express my sincere appreciation to all of you who actively participated to make this event come together to become a success particularly, the Management of the School of Electrical Engineering and Computing and the organizers and Initiators of this conference. We couldn't have done it without you!

Once again a very warm welcome to each and every one of you and I wish that you will have a very fruitful time in the coming days of our International conference.

Following this, I would like to invite our Chief Guest – His Excellency Engineer Worku Gachena - Director General of Artificial Intelligence and Board Chairperson of our University to make an Opening remarks.

I thank you!!

Opening Speech

Engineer Worku Gachena
Director General of Artificial Intelligence
Board Chairperson of ASTU

Honored guests, distinguished researchers, Ladies and gentlemen, good morning. It's a real privilege and an honor for me to address this conference of strengthening awareness and application of artificial intelligence technology in Ethiopia.

We live in exciting times where the change in technology and sciences is disruptive with a vast pervasive impact throughout society. I am delighted to see conferences like this while we are trying to create awareness in the society to strengthen Artificial intelligence in the country. I, on behalf of, Ethiopian artificial intelligence Institute, I would like to extend my sincere appreciation and say thank you to Adama Science and Technology University and the collaborators Google and Deep Learning Indaba for organizing such platform and have the national and international scholars for sharing their scientific knowledge and skills to enhance awareness of Artificial intelligence in the country. Once again, thank you all for coming together to participate on the **2nd Deep learning Indaba-X 2021 international conference**.

In the existing digital world, the competitiveness of a given country largely relies on its effective utilization of information technologies in general and digital technologies such as Artificial Intelligence in particular. Several countries across the world managed to boost their economic growth and emerge as world giants mainly because of effective utilization of digital technologies. The secret behind a leap-frogging economic growth and rapid industrialization of those Asian Tigers: Hong Kong, South Korea, Singapore and Taiwan, is nothing but effective utilization of digital technologies.

Artificial Intelligence is advancing dramatically. It is already transforming our world socially, economically and globally. As the discipline of Artificial Intelligence advances mostly outside the boundaries of Africa, it appears that the entire continent is getting left behind; Even if it is believed to profoundly impact services and productivity in low-and middle-income countries. Artificial intelligence is a very important opportunity for developing countries like Ethiopia. If the challenges associated with AI technology can successfully be navigated and discussed and narrowed with this kind of platform, it can be a driver for growth and development of our nation. The world has already witnessed the great fits of Artificial Intelligence in almost every field out there during the past decade. It has the potential to enhance productivity by expanding opportunities for the country's development in key sectors such as agriculture, manufacturing industries, healthcare, financial services, telecommunication, and other government and public services. To properly function as a country and as an individual in the existing and forthcoming digital world, the use of digital technologies is not a choice that we make, but a necessity that we should meet.

As you might already know, the Planning and Development Commission of Ethiopia has unveiled its 10-years development plan (2013-2022 E.C) under the theme 'Ethiopia: An African Beacon of Prosperity". During the life span of the new plan, Ethiopia's economy is expected to experience a 10.2 percent average

growth annually. And it is almost impossible to think of such fast economic growth without effective utilization of digital technologies. The government of Ethiopia has put due emphasis to the development of information communication technologies sector in its 10-years development plan.

Ethiopia's digital economy is at an early stage of development with few private sector players offering digital services and some government driven digitization initiatives.

Currently, the government is working immensely to leverage the digital space and build a more prosperous society. A couple of years ago, the Ethiopian Artificial Intelligence Institute was established to help Ethiopia "catch up" with the other countries of the globe that have already leaped in AI technology. The institute has a vision, in 2030 to be a state-of-the-art National AI research and Development Center with Excellence and Key Role in Creating Innovative AI-enabled Solutions at national and international levels.

To this end, the institute is currently working on various research areas to mitigate problems that occur in production, service provision, and regular operations of various sectors using Artificial Intelligence applications. It has so far reached the development of several prototypes that possess the potential of providing solutions upon reaching into fully-fledged products for solving problems that the health, agriculture, finance, and transport sectors encounter.

Undoubtedly, the dissatisfaction of students and researchers in data sciences and artificial intelligence for lacking appropriate data centers and AI infrastructure is being voiced. Talented and skillful individuals in these areas seek careers elsewhere. If Ethiopia in particular, and Africa, in general, fail to maintain its best and brightest citizens, its international competitiveness in the area of Artificial Intelligence will suffer and remain disadvantaged. To cope up with this challenge, the Ethiopian Artificial Intelligence institute is working its level best in the provisioning of a cluster with a state-of-the-art data center and AI laboratories at its premises. This has the purpose of bringing together researchers and AI enthusiasts across the country and taking part in innovative endeavors.

Today, Ethiopia is once again at a crossroads which dictates whether to remain a consumer of the technological products by others, as has been in the past or to actively engage in shaping the technology itself. However, I strongly believe that the road ahead is marked by so many opportunities if we all commit to harness, nurture, and enhance the country's capabilities. This will require the establishment of sound AI and digital policy, a comprehensive and coherent strategy, and an objective assessment of the country's skilled manpower in the area.

The designation of the two universities: Adama Science and Technology University(ASTU) and Addis Ababa Science and Technology University as science technology universities with especial mandate of promoting science and technology in Ethiopia also shows the readiness and commitment of the government to build technology or knowledge based economy.

I hope, it is with this major objective that ASTU has organized the **2nd Deep Learning Indaba-X Ethiopia 2021 conference** in collaboration with Deep learning Indaba Limited on the Theme of "***Strengthening Awareness and Application of Artificial Intelligence in Ethiopia***". As I managed to learn from the very theme of the conference and the summary those papers selected for presentation, the main objectives of this conference is improving awareness and enhancing the application of Artificial Intelligence in various

sectors in Ethiopia. Among the major themes covered in the 21 original research papers and five plenary speeches selected for this conference are:

- AI For Sustainable Development
- AI for Good Policies and Standards
- Robotics And Automation
- Natural Language Processing
- AI in Practice

As can be seen from these themes, Artificial Intelligence has broader areas of application. It can be used in the area of manufacturing, transportation, wireless communication, health service provision, space science, climate change monitoring, power an energy among others.

In general, the development in the area of information technology is moving fast from "smart" to "intelligent". Recently, intelligent electronic gadgets are replacing those smart gadgets in various sectors.

Therefore, as a country, we need to move faster to cope up with these ever-changing advancements in information technology. In this regard, higher learning institutions such as ASTU and Ethiopian Artificial Intelligence Institute have a lot to do. In addition to organizing such types of collaborative conferences, ASTU must strengthen its ties with renowned international and national universities and research institutes and strongly engage in AI related research and technology transfer projects.

Institutions such as **Deep Learning Indaba Limited** shall also be encouraged to further expand their collaborations with ASTU, other higher education institutions in Ethiopia and also with Ethiopian Artificial Intelligence Institute as well. In relations to this, I would like to assure you that the government is fully committed to provide you with all the necessary support in your endeavor to promote the application of digital technologies in various sectors.

Before I windup my speech, let me use this opportunity to thank the School of Electrical Engineering and Computing for initiating this conference and the Office of Vice President for Research and Technology Transfer for effectively organizing the conference. I would also like to thank once again the Deep Learning Indaba Limited for sponsoring the conference.

Finally, wishing you a successful and fruitful time, I now declare that **the 2nd Deep Learning Indaba-X Ethiopia 2021 conference** is officially opened.

Thank you!!

Keynote Address

Dr. Yabebal Tadesse Fantaye

African Institute for Mathematical Sciences

Keynote Title: Probalisctic theory and deep learning

Dr. Taye Girma

Deputy Director General, Ethiopian Artificial Intelligence Institute

Keynote Title: AI research challenges and opportunities in Ethiopia

Dr. Girmaw Abebe Taddesse

Research Scientist IBM Research - Africa

Keynote Title: Computer Vision and AI: Advances, Applications and Concerns

Dr. Dawit Assefa

Head, Center of Biomedical Engineering,
Addis Ababa Institute of Technology, AAU

Keynote Title: Biomedical Imaging Applications of Machine Learning

Prof. Jemal H. Abawajy

Director, the Parallel and Distributing Computing Laboratory, Deakin University, Australia

Keynote Title: Towards smart and suistainable household waste management system

Syndicate Session 1

Natural Language Processing

Amharic Semantic Parser Using Deep Learning

Gashaw Demlew

Faculty of Computing and Informatics, Jimma University, Jimma, Ethiopia. E-mail: gashudemman@gmail.com

ABSTRACT

To process and understand natural languages, it is necessary to organize the linguistic structure of the texts at different levels. The semantic level of linguistic analysis is concerned with producing accurate representations of the meaning of utterances that may contain significant conventions. The semantic analysis aims to map natural language discourses to machine-comprehensible meanings. Traditional methods of creating semantic analyzers depend on high-quality lexicons, hand-made grammars, and language features that are limited by the domain or expression applied. This paper presents an Amharic semantic parser developed using an attention-enhanced sequence-to-sequence neural model that encodes utterances and generates their logical forms. The model is constructed to handle different types of sentences from simple to complex, and to accept additional difficulties for the Amharic language such as rich morphology and greater freedom in the composition of sentences. I trained the model with a fully supervised training setting where utterances-logic forms are given and tested in Amharic sentences collected in three different domains. It scored 0.95 for BLEU-4 and 0.97 for LEPOR.

Keywords: *Semantic parsing, Amharic semantic parser, Deep learning, Sequence-to-sequence, Full supervision*

1. INTRODUCTION

In order to process and understand natural languages, the linguistic structures of the text must be organized at different levels. Structured text increases the performance of NLP applications [1] [2]. Semantic parsing is the mapping of natural language utterances into a machine-readable formal representation of their meaning (logical form) with many applications. such as question answering [3] [4], relationship extraction [6], goal-oriented dialogue [7], natural language interfaces, robot control [8], instruction interpretation [9]. Traditional methods of building a semantic parser [10][11] rely on high-quality lexicons, hand-craft grammars, and linguistic features that are applied in a domain-limited manner (i.e., they work on limited domains with a small number of logical predicates) or representation. Since the Amharic language, in particular, is one of the least available and morphologically complex languages, it is difficult to obtain and create such lexicons and features; This motivates me to find another approach that is more suitable for the Amharic language.

The rise of the sequence-to-sequence (Seq2Seq) model [14] provides an alternative way to tackle the mapping problem and reduces the need for domain-specific assumptions, grammar learning, and generally broader feature engineering by considering semantics analysis as a sequential problem. Different types of supervision have been studied to train semantic parsing; full supervision, given a set of input sentences and their corresponding logical forms (it's more effective but more expensive to get [15]); Alternative forms of supervision have been proposed to reduce the burden of annotation, training of utterance-denotation pairs [10][16] or the use of remote supervision [15][17].

In this work I propose to use a domain-independent neural semantic parsing, a machine learning-based semantic parsing particularly deep learning-based, that learns from input sentences paired with meaning representations (full supervision based), and generate a parser that can parse new input sentence. The parser

differs from conventional semantic parser in that it does not require lexicon-level rules to specify the mapping between natural language and logical form tokens. Instead, the parser is designed to handle cases where there the lexicon is missing or incomplete thanks to a neural attention layer that encodes a soft mapping between natural language and logical form tokens. This modeling choice greatly reduces the number of grammar rules used during inference to those only specifying domain-general aspects. Other than the above-mentioned solution to tackle the annotation burden, the proposed solution pool examples from multiple datasets in different domains, each corresponding to a separate knowledge-base (KB), and train a model's overall examples. This is motivated by the observation that while KBs differ in their entities and properties, the structure of language composition repeats across domains.

In this article, I propose to use a domain-independent neural semantic parsing, a machine learning-based semantic analysis that learns from input sentences coupled with meaning representations (based on full supervision), and a parser that can parse new input sentences. The parser differs from the traditional semantic parser in that it does not require lexical-level rules to specify the mapping between natural language and logic form tokens. Instead, the parser is designed to handle cases where the lexicon is missing or incomplete, thanks to a neural attention layer that encodes a smooth mapping between natural language and logic form tokens. This modeling option significantly reduces the number of grammar rules used during inference to those that only specify general aspects of the domain. In addition to the above-mentioned solution to address annotation burden, the proposed solution includes utterances from multiple datasets in different domains, each corresponding to a separate knowledge base (KB), and trains a model with all utterances. This is motivated by the observation that although KBs differ in their entities and properties, language composition structure is repeated across domains.

Various formalisms (logical forms) have been proposed to represent natural language meaning, such as lambda calculus [18], frame semantics [19], abstract meaning representations [20], and functional query language [21]. In this article, I use FunQL, which maps first-order logical form to functional form, resulting in recursive tree-structured representations of meaning. Although FunQL lacks the full expressiveness of lambda calculus, it is easy to annotate sentences in their logical forms.

I adopt the general encoder and decoder framework based on advanced neural networks with an attention mechanism that allows the model to learn soft alignment between utterances and logical forms. There are research papers [22] [23] closer to this work (/learning-based approach) and based on the traditional/grammar-based approach. However, some of the approaches are unable to parse complex sentences due to grammar coverage and some of the parsers are developed for specific languages that cannot be applied to the Amharic language. The intention of this research is to develop an Amharic semantic parser that answers the following research questions:

- ✓ What kind of approach is appropriate to develop a parser for Amharic language? (i.e. grammar based vs learning based)
- ✓ How the parsing approach will handle language specific issues (such as morphological complexity and freedom in sentence compositions)

- ✓ What type of sentences the parser can parse? (I.e. simple sentence vs complex sentence like aggregation, comparison, superlative)

As far as I know, there are no published articles on the semantic parsing done for the Amharic language. Therefore, this article became the first for the Amharic language.

2. STRUCTURE OF AMHARIC LANGUAGE

A) Amharic Language

More than 80 languages are spoken in Ethiopia, which now has a population of more than 110 million. Amharic is the working language of the country's federal government and spoken as a first language by a large portion of the population, and is the most widely studied language in the entire country. As a result, Amharic is the country's lingua franca in modern times [24]. Unlike Arabic, Hebrew, and Syriac (other Semitic languages), Amharic is written using a syllabary system originally developed for the extinct Ethiopian Semitic language Ge'ez and later extended to Amharic and other Ethiopian Semitic languages. As in other abugida systems, each character in the Ge'ez (or Ethiopic) writing system derives its base form from the consonant of the syllable, and the vowel is represented by more or less systematic modifications of these base shapes.

The alphabet of the Amharic language consists of 33 core symbols or Fidel (ፊደል). Each of these core symbols comes in seven different orders; the base character plus six different symbols or orders formed from the base character. There are also 37 additional characters representing labialized variants of consonants followed by specific vowels. The complete system has 268 characters. There are also a set of Ge'ez numbers, but nowadays these tend to be replaced by the Hindu-Arabic numbers used in European languages [25].

B) Grammatical Rules of Amharic Language

Yimam [26] and Amare [27] classified phrase structures of the Amharic language as noun phrases (NP), verb phrases (VP), adjectival phrases (AdjP), adverbial phrases (AdvP), and prepositional phrases (PP). These phrases have principal word classes as heads. For example, an Amharic noun phrase has a noun as its head; an Amharic verb phrase has a verb as its head; etc. Amharic phrases, except prepositional phrases, can be made from a single headword or with a combination of other words.

The Amharic language follows a subject-object-verb (SOV) grammatical pattern as opposed to, for example, the English language, which has an SVO sequence of words [3], [18]. For instance, the Amharic equivalent of the sentence "John ate bread" is written as "ጆን (Jon/John) ዳቦ (dabo/bread) በላ (bäla/ate)". Amharic sentences can be constructed from simple or complex NP and simple or complex VP. Simple sentences are constructed from simple NP followed by simple VP, which contains only a single verb. Complex sentences are sentences that contain at least one complex NP or complex VP or both complex NP and complex VP. Complex NPs are phrases that contain at least one embedded sentence in the phrase construction. The embedded sentence can be complemented.

In addition, since Amharic is a morphologically complex language in which verbs, nouns, and adjectives are marked for various grammatical functions, such agreements must be checked: adjective-noun, adjective-verb, subject-verb, object-verb, and adverb-verb (Amare, 2010) [27].

a) ኢትዮጵያ ውስጥ ስንት ሀይቆች አሉ? (Etyopya wst snt hayqoch alu/ How many Lakes are there in Ethiopia?)
b) Answer (count (A) , ሀይቅ (A) , መገኛቦታ (A , ኢትዮጵያ))

Figure 1: Example of utterance-logic form pairs

3. Design of Semantic Parser

A sequence-to-sequence recurrent neural network (Sec2Sec RNN) approach that considers both encoded input and decoded output as sequences are used to develop a semantic parser that maps Amharic language input sentence q $=x_1 \ldots x_{|q|}$ to a logical form representation of it meaning a $=y_1 \ldots y_{|a|}$ (For example, as depicted in Figure 1, the aim is translating input sentence (a) to its corresponding logic form (b)). Figure 2 shows the proposed architecture of the neural semantic parser.

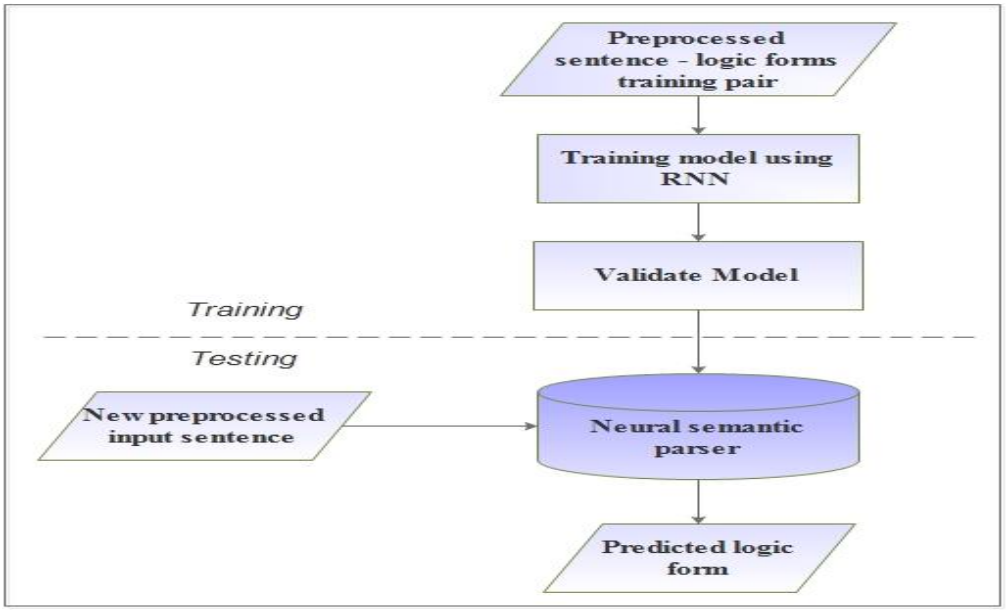


Figure 2: The architecture of the Amharic semantic parser

A) Sentence Pre-processing

It is the module above that takes input sentences and pre-processes them by simple NLP tools like Normalizer and Stemmer (words in a sentence are normalized and stemmed). This process is language-dependent and includes the following activities; Removing unnecessary words, changing characters and words to their common form, and stemming.

- **Removal of extraneous characters:** Words containing numbers like (2nd i.e. ፪ or ኢዜአ103862) were excluded at the first phases of preprocessing. Furthermore, the standard control character; Amharic punctuation marks and symbols borrowed from other languages (?!, "," , " , /,, etc.) were ignored.

- **Normalizations**: In Amharic, some words can be written in different formats. It would cause the same word to be treated in different ways which could reduce the efficiency and accuracy of the parser. So, this activity normalizes these spelling variations by changing the different forms of a character into one common form (Table 1:) shows an example of the character redundancy where more than one symbol is used for the same sound). The other normalization issue is related to shorthand representation of words like ኦ/ኦ, ት/ሚ, and ጽ/ቤት. I have collected and prepared a dictionary of short form-to-long form to convert them existing in the input sentence into their expanded long forms.

Table 1*:* Redundant Amharic characters

Consonants	Other symbols with the same sound
ሀ (hä)	ሃ ሐ ሓ ኃ ኅ ኻ
ሰ (sä)	ሠ
አ (ä)	ኣ ዐ ዓ
ጸ (tsa)	ፀ

- **Stemming:** the process of reducing inflected words to their stem. In this work, it is sufficient that related words map to the same stem, and it provides a means to minimize vocabulary terms and hence increases the performance of the parser (accuracy and efficiency). In this thesis, I employ the stemming algorithm developed in [28].

***B)* Sequence-to-sequence model (seq2seq)**

It is an encoder-decoder model with two different L-layer recurrent neural networks that encode input sequence to vector representation and then the decoder which learns to generate output sequence conditioned on the encoding vector. As I have stated, the training setup of the model is fully supervised using utterance-logic form pairs. Thus, the aim is to learn a model which maps natural language input q =$x_1 \ldots x_{|q|}$ to a logical form representation of it meaning y =$y_1 \ldots y_{|a|}$. The entire model is trained end-to-end by maximizing p (a\q) (1):

$$p(y\backslash q) = \prod_{t=1}^{|a|} p(y_t | y < t, q) \qquad (1)$$

where, y<t =$y_1 \ldots y_{t-1}$.

I also extend the model with long short-term memory (LSTM) neuron cell to handle long-range dependency (i.e. longer sentences) and with bidirectional RNN to embrace the context of input and employ an attention-based copying mechanism to learn soft alignment at each time step of the decoder (as shown in Figure 3, 2-layer RNN). Below I briefly discuss the basic component of the model (encoder, decoder, and attention mechanism).

Encoder: The encoder converts the input (user sentence) sequence $x_1 \ldots x_{|q|}$ into a sequence of context-sensitive embedding $b_1 \ldots b_{|q|}$using a bidirectional RNN. First, a word embedding function $\emptyset^{(in)}$maps each word x_i to a fixed-dimensional vector. These vectors are fed as input to two RNNs: a forward RNN and a backward RNN. The forward RNN starts with an initial hidden state h_0^F, and generates a sequence of hidden states $h_1^F, \ldots, h_{|q|}^F$ by repeatedly applying the recurrence (2**Error! Reference source not found.**:

$$h_i^F = \boldsymbol{LSTM}\left(\emptyset^{(in)}(x_i), h_{i-1}^F\right) \quad (2)$$

where, LSTM refers to the LSTM function being used**.**

The backward RNN (h_i^B) similarly generates hidden states $h_{|q|}^B, \ldots, h_1^B$ by processing the input sequence in reverse order. Finally, for each input position i, I define the context-sensitive embedding b_i to be the concatenation of h_i^F and h_i^B.

Decoder: the decoder is a sequence decoder that generates output tokens one at a time. Once the tokens of the input sequence $x_1 \ldots x_{|q|}$ are encoded into vectors, they are used to initialize the hidden states of the first-time step in the decoder. Next, the hidden vector of the topmost LSTM h_t^L in the sequence, the decoder is used to predict the t-th output token as (3):

$$p(y_t|y < t, q) = \boldsymbol{tanh}(W_0 h_t^L)^T e(y_t) \quad (3)$$

where W_0 is a parameter matrix, $e(y_t)$ a one-hot vector for computing y_t's the probability from the predicted distribution, and tanh is the activation function which is adopted in the output layer to predict the probability of each word's probability.

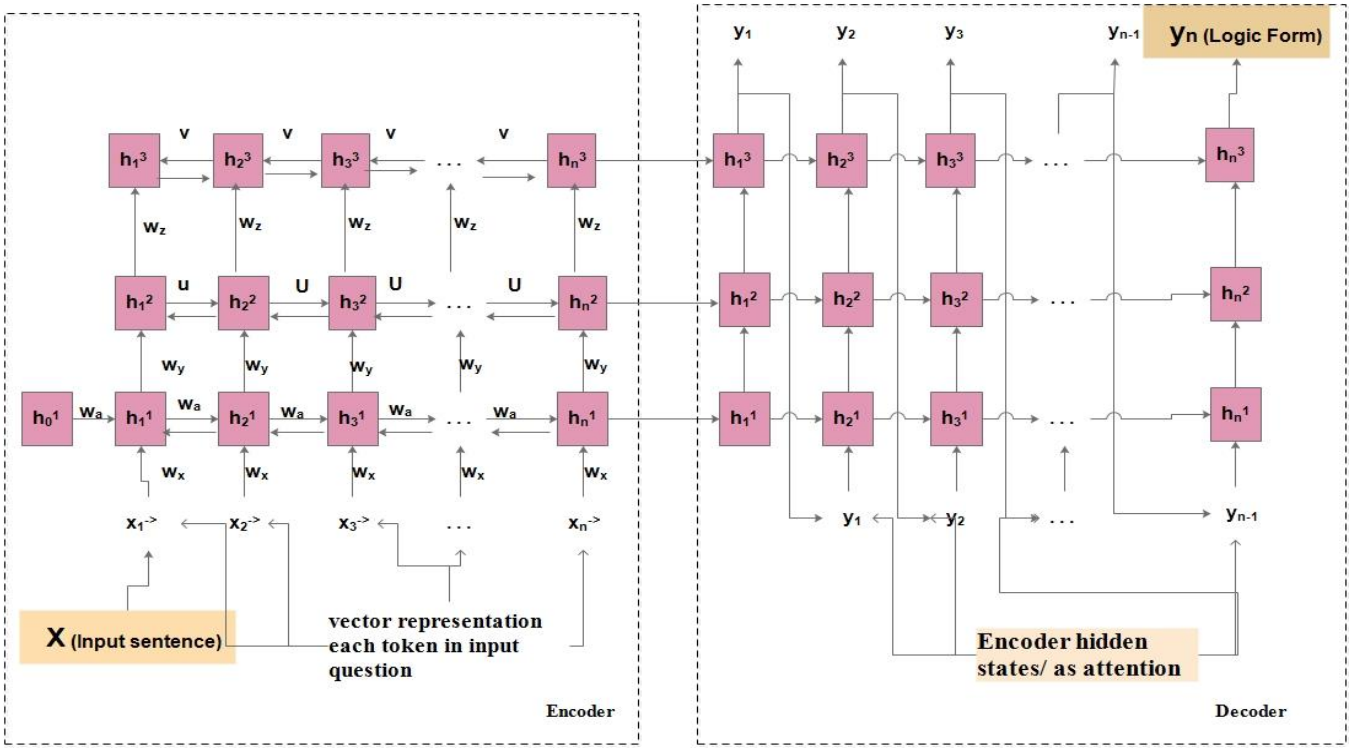


Figure 3: The 3-layer seq2seq RNN model

Attention Mechanism: As shown in equation (1), the hidden vectors of the input sequence are not directly used in the decoding process. However, it makes natural sense to integrate encoder-side information (in the form of a context vector) at each time step of the decoder to better predict the current token. The attention model considers the alignment history (it helps to adjust future attention and guide the decoder towards unprocessed source words) through explicitly modeling the decoding coverage of the source words. To find relevant encoder-side context for the current hidden state h_t^L of decoder, I compute its attention score with the kth hidden state in the encoder (4):

$$s_k^t = \frac{exp\{b_k.\ h_t^L\}}{\sum_j^{|q|} exp\{b_j.\ h_t^L\}} \quad (4)$$

where $b_1 \ldots b_{|q|}$ are the hidden vectors of the top-layer of an encoder. Then, the context vector is the weighted sum of the hidden vectors in the encoder (5):

$$c^t = \sum_{k=1}^{|q|} s_k^t\, b_k \quad (5)$$

Instead of equation (1), I use this context vector, which acts as a summary of the encoder, to compute the probability of generating y_t in the model as (7):

$$h_t^{att} = \boldsymbol{tanh}(W_1 h_t^L + W_2 c^t) \quad (6)$$

$$p(y_t|y < t, q) = \boldsymbol{tanh}(W_o h_t^{att})^T e(y_t) \quad (7)$$

where W_0, W_1, W_2 are the three-parameter matrices and $e(y_t)$ a one-hot vector for computing y_t's the probability from the predicted distribution.

4. EXPERIMENT

A) The corpus

The dataset used to build and test the parser is collected from questions asked online in three different domains (arts, football, and geography) and each question is annotated by its corresponding logical forms. I collected 15453 questions. The data set was divided into a training set of 12,516 (80%), a validation set of 1,391 (10%) and 1,546 (10%) test set.

In this work, I have adopted FunQL (i.e. less expressive than the lambda calculus, but simple) as the semantic formalism of the logical form. I have used similar domain-general functional operators defined by the authors [15] to define the FunQL formalism as shown in Table 2. The dataset consists of different types of questions, such as aggregate, comparison, yes/no, superlative, and factoid/list types.

Table 2: Domain general functional operators to define FUNQL formalisms

Operators	Description	Examples
Entity	creates a unary logical form whose denotation is a singleton set containing that entity	ቀዳማዊ ኃይለሥላሴ
Relation	Acts on an entity and returns as a detonation the set of entities that satisfy the relation.	ልጆች (ኃይለ ሥላሴ, A) corresponds to the question "የ ኃይለ ሥላሴ ልጆች እነማ ናቸው? "
Count	Returns the cardinality of a set of entities.	Count (A, ልጆች (ኃይለ ሥላሴ, A)) corresponds to the question "ኃይለ ሥላሴ ስንት ልጆች አሏቸው?"
argmax or *argmin*	The operator returns a subset of entities whose specific property is maximum or minimum	Argmax (ልጆች (ኃይለ ሥላሴ, A), እድሜ) corresponds to the question "የኃይለ ሥላሴ የመጀመሪያ ልጅ ማነው?"
Filter	Returns a subset of entities where a comparative constrain is satisfied	Filter (ልጆች (ኃይለ ሥላሴ), > (እድሜ, 52)) corresponds to the question "ከ ኃይለ ሥላሴ ልጆች እድሜው ከ52 የሚበልጠው ማነው?"

***B)* Experiment setting**

The model was implemented and trained using a Python programming language with Keras integration [29]. I prepared three different configurations depending on the data I used for training (In short, I developed three parser models with different data configurations to compare the impact of dataset preprocessing on model training); The first configuration is based on a "*normalized but not stemmed question dataset*", the second is based on a "*normalized and stemmed question dataset*" (ie the words in the questions have been stemmed), and the last is a "*stemmed dataset but without an attention mechanism*".

In all settings, I replaced the word vectors in the questions for words that occur only once in the training set with a universal <unk> (unknown) word vector but left all tokens in the logical forms. To train the models I used LSTM with 3 layers, with 200 cells in each layer and a 300-dimensional word embedding. The embedding was initialized with the Skip Gram Word2Vec model (I built it using a collected Amharic corpus and evaluated it with a word relatedness evaluator, an intrinsic evaluator type [30]) and achieved a Spearman correlation score of 0.64 [31]). All the details of the training are given below:

- I initialized all the parameters of an LSTM with the uniform distribution between -0.08 and 0.08
- I used the RMSProp algorithm (with batch size set to 20) to update the parameters. The smoothing constant of RMSProp was 0.95.
- Gradients were clipped at 5 to alleviate the exploding gradient problem.
- The dropout rate was set to 0.5, which computes the softmax activation of the next action or token.
- I trained the model for 100 epochs (with early stopping) with an initial learning rate of 0.1, and halved the learning rate every 5 epochs, starting from epoch 15.
- I used the beam search with beam size 5 to generate logical forms during inference.

***C)* Evaluation**

The evaluation is based on the prepared test data. Each model is evaluated by using the bilingual score understudy score (BLEU [32], which is a metric for evaluating a generated logical form to a reference logical form) and LEPOR (length penalty, precision, N-gram position difference penalty, and Recall [33]) to obtain a quantitative idea of how well the model performed. I calculated various cumulative BLEU values by adjusting the weights; BLEU1 (1, 0, 0, 0), BLEU2 (0.5, 0.5, 0, 0), BLEU3 (0.33, 0.33, 0.33, 0) and BLEU4 (0.25, 0 .25, 0.25, 0.25) and the results from the models are presented in Table 3.

As shown in Table 3, the NSPM (Neural Semantic Parser Model) with Stemmed and Attention Mechanism is superior to other models in all scores. It achieved a 0.95 score of BLEU-4 and a 0.97 score of LEPOR (meaning it offers a cap on what I can expect from this model), beating NSPM Not Stemmed and NSPM with Stemmed and without attention Mechanism. This is to be expected since the hidden vectors of the input sequence were used directly in the decoding process (i.e. it means relevant information from the encoder side is taken into account to better predict the current token) and the different forms of each token encountered in the training questions were changed into their stemmed form.

When comparing my benchmark model (i.e. NSPM with Stemmed) to other deep neural-based semantic parser models developed for other languages, it performs similarly. However, some problems/ inconveniences affect the parser to parse complex sentences; The first is that the stemmer I use is of **lower**

quality; For example, if we accept these two questions taken from the stemmed question sets ሀዋሳ ከተማ የሚለማመዱበት ስታድየ ተመልካች የመያዝ አቅም ስንት ነው" and "አበበ ቢቂላ ስታድየ ተመልካች በመያዝ አቅም አዳማ ስታድየ ይበልጣል", words የመያዝ (from the first question) and በመያዝ (from the second question) stemmed in different forms (but must have the same stemmed form, መያዝ), and are expected to map to the same token (ተመልካች_አቅም) in the corresponding logical forms. Therefore, the probability of each word associated with the token ተመልካች_አቅም is less than the probability of its stemmed form (i.e. መያዝ). In general, the quality of the stemmer has a direct impact on the performance of the model.

Table 3: Evaluation results of different semantic parsers by calculating four cumulative Bleu scores

Scores / Models	BLEU-1(1-gram)	BLEU-2(2-gram)	BLEU-3(3-gram)	BLEU-4(4-gram)	LEPOR
NSPM with Stemmed	0.962	0.957	0.952	0.945	0.970
NSPM not Stemmed	0.921	0.903	0.889	0.867	0.915
NSPM Stemmed No Attention	0.944	0.928	0.917	0.904	0.939

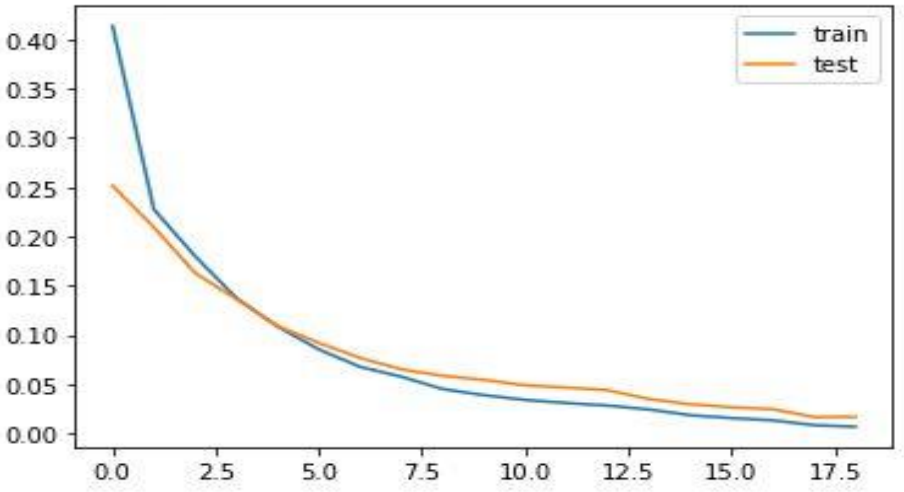


Figure 4: The loss function of the model (NSPM with Stemmed model) in the training and validation dataset where the X-axis refers to epochs and the Y-axis refers to loss values

The second case is **rare words**; Because the training set is relatively small, some question words and semantically light expressions such as the verbs to be and to have, as well as prepositions are rare in the training set, making reliable parameters difficult to estimate (hard to determine dealing with context and meaning). The size of the data also impacted getting good accuracy for each question form of the same question and question types (simple-complex). The last case is **the number of layers**; I used three layers for the encoder and decoder models where they don't provide any further rendering power to the model. This is because the power limit of a computer I used for experimenting is limited (requires GPU) to train the model with >3 hidden layers and more epochs (I will leave this for future work).

Table 4: Sample question sentences from test dataset and their evaluation scores

No	Question sentences	Question type	Actual logic-form (expert annotated)	Predicated logic-form	BLEU-4 score	LAPOR score
1	ፍቃዱ ከበደ ምን ምን ፊልሞችን ሰርቷል	Factoid	answer (A , ተዋናይ (ፍቃዱ_ከበደ) , ፊልም (ፍቃዱ_ከበደ , A))	answer (A , ተዋናይ (ፍቃዱ_ከበደ) , ፊልም (ፍቃዱ_ከበደ , A))	1.0	1.0
2	ሶስት ማእዘን የሚለውን ፊልም ዳይሬክት ያደረገው ማነው	Factoid	answer (A , ዳይሬክተር (A) , ፊልም (A , ሶስት_ማእዘን))	answer (A , ዳይሬክተር (A) , ፊልም (A , ሶስት_ማእዘን))	1.0	1.0
3	በኢትዮጵያ ከሚገኙት ሀይቆች መካከል በጥልቀት ትንሹ ሀይቅ ማነው	Comparison	answer (A , argmin ((ሀይቅ (A), መገኛ_ሀገር (A , ኢትዮጵያ) , ጥልቀት (A , B)) , B))	answer (A , argmin ((ሀይቅ (A), መገኛ_ሀገር (A , ኢትዮጵያ) , ስፋት (A , B))	0.76	0.86
4	ማሞ ወልዴ የ5000 ሜትር እሯጭ ነው	Yes/no	answer (? , የተሳተፈበት_ሩጫ_አይነት (ማሞ_ወልዴ , 5000_ሜትር) , እሯጭ (ማሞ_ወልዴ))	answer (? , የተሳተፈበት_ሩጫ_አይነት (ማሞ_ወልዴ , 5000_ሜትር) , እሯጭ (ማሞ_ወልዴ))	1.0	1.0
5	ቴወድሮስ ታደሰ ስንት አልበሞችን ሰራ	Aggregation	answer (count (A) , አልበም_ስም (ቴወድሮስ_ታደሰ , A))	answer (count (A) , አልበም_ስም (ቴወድሮስ_ታደሰ , A))	1.0	1.0
6	ራስዳሽን ተራራ ኢትዮጵያ ውስጥ ከሚገኙት ተራሮች መካከል በርዝመት ትልቁ ተራራ ነው	Comparison	answer (? , ከፍታ (ራስዳሽን_ተራራ , A) , ተራራ (C) , ከፍታ (C , B) , ይበልጣል (A , B))	answer (? , ከፍታ (ሻሸመኔ_ስምጥ_ሸለቆ , A) , ተራራ (C) , ከፍታ (C , B) , ይበልጣል (	0.73	0.82
7	መግደል መሸነፍ የሚለው ፊልም ላይ ስንት ተዋናዮች ተሳትፈዋል	Aggregation	answer (count (A) , ተዋናይ (A) , ፊልም (A , መግደል_መሸነፍ))	answer (count (A) , ተዋናይ (A) , ፊልም (A , መግደል_መሸነፍ))	1.0	1.0
8	ደራሲ ገብረክርስቶስ ደስታ ዲግሪ ያሰበደው የሚለውን መጸሃፍ ጽፈዋል	Yes/no	answer (? , ደራሲ (ገብረክርስቶስ_ደስታ) , ድርሰት (ገብረክርስቶስ_ደስታ , ዲግሪ_ያሰበደው))	answer (? , ደራሲ (ገብረክርስቶስ_ደስታ) , ድርሰት (ገብረክርስቶስ_ደስታ , ዲግሪ_ያሰበደው))	1.0	1.0
9	ጋምቤላ ክልል ውስጥ በርታ ይነገራል	Yes/no	answer (? , የሚነገር_ቋንቋ (ጋምቤላ_ክልል , በርታ))	answer (? , የሚነገር_ቋንቋ (ጋምቤላ_ክልል , አውንጅ))	0.86	0.93
10	ፍጹም ቃል የሚለው ፊልም ላይ ስንት ተዋናዮች ተሳትፈዋል	Aggregation	answer (count (A) , ተዋናይ (A) , ፊልም (A , ፍጹም_ቃል))	answer (count (A) , ተዋናይ (A) , ፊልም (A , ፍጹም_ቃል))	1.0	1.0
11	መሰረት ደፋር በሜዳሊያ ብዛት ከጥሩነሽ ዲባባ ትበልጣለች	Comparison	answer (? , ሜዳሊያ (መሰረት_ደፋር , A) , ሜዳሊያ (ጥሩነሽ_ዲባባ , B) , ይበልጣል (count (A) , count (B)))	answer (? , ሜዳሊያ (መሰረት_ደፋር , A) , ሜዳሊያ (መሰረት_ደፋር , B) , ይበልጣል (A , B))	0.60	0.80
12	አቡቤሜዳ ተራራ ጭላሎ ተራራ በከፍታ ይበልጣል	Comparison	answer (? , ከፍታ (አቡቤሜዳ_ተራራ , A) , ከፍታ (ጭላሎ_ተራራ , B) , ይበልጣል (A , B))	answer (? , ከፍታ (አቡቤሜዳ_ተራራ , A) , ከፍታ (አቡቤሜዳ_ተራራ , B) , ይበልጣል (A , B))	0.89	0.97

5. CONCLUSION AND FUTURE WORK

The contributions in this paper use the deep learning approach to map the Amharic language sentence into its corresponding logical form and acquire how much the deep learning approach will benefit to develop a parser with minimal knowledge of the domains and knowing what data sets to develop increases the power of deep learning for the Amharic language. In future work, I will look for ways to refine the training monitor to reduce the annotation load and train the model on more than three layers and a larger dataset/domains to give the parser more rendering power.

REFERENCES

[1] S. Abney, "Chunks and Dependencies: Bringing Processing Evidence to Bear on Syntax," in *Computational Linguistics and the Foundations of Linguistic Theory. CSLI*, 1995.

[2] S. K. E. L. E. Bird, "Natural Language Processing with Python," *O'Relly Media Inc*, 2009.

[3] T. L. Z. S. G. a. M. S. Kwiatkowski, "Lexical generalization in CCG grammar induction for semantic parsing", in *In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, 2011.

[4] P. J. a. D. K. Liang, "Learning dependency-based compositional semantics", in *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, 2011.

[5] A. C. R. F. a. P. L. Berant, "Semantic parsing on Freebase from question-answer pairs," in *In Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[6] J. a. T. M. Krishnamurthy, "Weakly supervised training of semantic parsers," in *In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jeju Island, Korea, 2012.

[7] C. N. F. L. Z. L. B. a. D. F. Matuszek, "A joint model of language and perception for grounded attribute learning," in *In Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 2012.

[8] D. L. a. R. J. M. Chen, "Learning to interpret natural language navigation instructions from observations," in *Association for the Advancement of Artificial Intelligence, AAAI*, 2011.

[9] Y. a. L. Z. Artzi, "Weakly supervised learning of semantic parsers for mapping instructions to actions," in *Transactions of the Association for Computational Linguistics*, 2013.

[10] E. C. Y. A. a. L. S. Z. Tom Kwiatkowski, "Scaling semantic parsers with on-the-fly ontology matching," in *In Empirical Methods in Natural Language Processing, (EMNLP)*, 2013.

[11] P. P. a. P. Liang, "Compositional semantic parsing on semi-structured tables," in *Association for Computational Linguistics (ACL)*, 2015.

[12] J. B. a. P. Liang, "Semantic parsing via paraphrasing," in *In Annual Meeting for the Association for Computational Linguistics (ACL)*, 2014.

[13] M. L. a. M. S. Siva Reddy, "Large-scale semantic parsing without question answer pairs," in *Transactions of the Association for Computational Linguistics*, 2014.

[14] O. V. a. Q. V. L. I. Sutskever, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems*, 2014.

[15] S. R. V. S. M. L. Jianpeng Cheng, "Learning an Executable Neural Semantic Parser," arXiv:1711.05066v1 [cs.CL], 2017.

[16] R. J. a. P. Liang, "Data recombination for neural semantic parsing," in *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics,ACL*, Berlin, Germany, 2016.

[17] Y. S. R. J. B. J. H. a. M. S. Bisk, "Evaluating induced CCG parsers on grounded semantic parsing," in *In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, 2016.

[18] R. Montague, "The proper treatment of quantification in ordinary English," *Springer*, vol. 49, p. 221–242, 1973.

[19] C. F. C. J. F. a. J. B. L. Baker, "The berkeley framenet project," in *In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, 1998.

[20] L. C. B. S. C. M. G. K. G. U. H. K. K. P. K. M. P. a. N. S. Banarescu, "Abstract meaning representation for sembanking," in *In Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 2013.

[21] J. M. Zelle, "Using inductive logic programming to automate the construction of natural language parsers," Austin, TX, 1995.

[22] M. L. L. Dong, "Language to logical form with neural attention," in *in Meeting of the Association for Computational Linguistics*, 2016.

[23] J. B. Jonathan Herzig, "Neural Semantic Parsing over Multiple Knowledge-bases," *ArXiv*, vol. abs/1702.01569, 2017.

[24] P. S. F. F. D. Lewis, Ethnologue: Languages of the World, Dallas: SIL International, 2013.

[25] C. W. Isenberg, Grammar of The Amharic Language, London: Richard Watts, 1976.

[26] B. Yimam, የአማርኛ ሰዋስው (Amharic Grammar), Addis Ababa, Ethiopia: Baye Yimam (1994), 2000.

[27] G. Amare, ዘመናዊ የአማርኛ ሰዋስው በቀላል አቀራረብ (Modern Amharic Grammar in a Simple Approach), Addis Ababa, Ethiopia: አዲስ አበባ ንግድ ማተሚያ ድርጅት (Addis Ababa trade publisher plc), 2010.

[28] M. Gasser, "HornMorpho: a System for Morphological Processing of Amharic, Oromo, and Tigrinya," in *Conference on Human Language Technology for Development*, Alexandria, Egypt, 2011.

[29] K. N, "Introduction to Keras," in *Deep Learning with Python*, Apress, Berkeley, 2017.

[30] I. L. a. D. M. Tobias Schnabel, "Evaluation Methods for Unsupervised Word Embeddings," in *in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015.

[31] W. W. Daniel, "Spearman Rank Correlation Coefficient, Applied Nonparametric Statistics," *MA:PWS-KENT*, p. 358–365, 1990.

[32] J. Brownlee, "A Gentle Introduction to Calculating the BLEU Score for Text in Python," in *Deep Learning for Natural Language Processing*, 2017.

[33] A. L. F. a. W. D. F. a. C. L. S. Han, "LEPOR: A Robust Evaluation Metric for Machine Translation with Augmented Factors," in *Proceedings of {COLING} 2012: Posters*, Mumbai, The COLING 2012 Organizing Committee, 2012, pp. 441-450.

[34] S. R. V. S. a. M. L. Jianpeng Cheng, "Learning an Executable Neural Semantic Parser," in *Computational Linguistics*, 2019.

An Architecture for Fake News Classification Using Machine Learning Techniques: In Case of Afaan Oromo Language

Hundaol Bedada[1], Kula Kekeba[2]

[1]*Department of Information Technology, Bule Hora University, Ethiopia*
[2]*Assistant Professor, Department of Software Engineering, Addis Ababa Science and Technology University, Addis Ababa, Ethiopia*

ABSTRACT

The social media proved to be the most popular virtual space where many people interact with each other and share their ideas, opinions using their language of choice. This platform often leads to spreading of fake news or unauthenticated news which creates the attention to identify them. Afaan Oromo is a language spoken by many more peoples in Ethiopia, other foreign country and it is the largest language in Africa. It is very difficult to distinguish the real ones from the fake news. Fake news posted in Afaan Oromo texts news is looked like one of the fundamental problems that face our society and the way it functions. To address this challenge machine learning-based architecture has been proposed which can classify fake news from real news. The proposed architecture encompasses the N-gram model to identify word-level TF, TFIDF features. For testing purposes, about 1263 Afaan Oromo news were collected (623 Afaan Oromo true news from Oromia Broadcasting Network and Fana Broadcasting Corporate, 640 Afaan Oromo false news from different Facebook accounts). Naive Bayes, SVM, and K-NN methods were used to train test the selected datasets. Our experimental results have shown that the Naïve Bayes classifier resulted in highest precision of 96.2% for a Term frequency of 5k, 10k, and for 50k and Unigram size. The SVM learner achieved 60.8% accuracy for TF features with unigram on 5000 feature size and achieve 96.2% accuracy by TFIDF.

Keywords: *Afaan Oromo, Fake News Classification, TF, TFIDF, Naïve Bayes, Support Vector Machine*

1. INTRODUCTION

News is information that is published in newspapers or broadcasted on mass media such as radio and television about certain events or phenomena in the world or some new information posted on social media about a particular event or incident [1] [2]. The false misinterpreted misleading and fabricated news can be identified as fake news. It is also a sort of tabloid or propaganda that consists of deliberate misinformation or hoaxes spread via traditional print and broadcasted through news media or online social media [3] [4].

According to [4] serious fabrications, large-scale hoaxes, and humorous fakes are the three major types of fake news. Serious fabrications are fraudulent reporting feres to the news which not heard previously. The intentional fabrications which extend from simple jokes to more complex which can be understood as authentic are called Large-scale hoaxes. Humorous fakes are creating a parody of the real news which can be accomplished by a professional journalist to alert the audience [4]. According to this categorization there publically available dataset accordingly for the English language but for Afaan Oromo there is no dataset that categorized as such in the above classification. Because of that our studies focused on all classification and we collected the above three types of data from social media like Facebook [4].

Afaan Oromo language is one of the under-resourced languages that have many speakers and containing a very little amount of computational linguistic and works to extract useful results from it. Afaan Oromo fake news detection and classification research is on an early stage, a challenging problem in the Ethiopian

for online social media users. To the best of the researcher's knowledge, there is no previous research conducted for the Afaan Oromo language to solve this problem and there is no dataset prepared to classify social media news. There are many more researchers were conducted on fake news detection with different languages. The conducted research does not work for Afaan Oromo since the language specification like syllable structure, syntax, and semantics of Afaan Oromo is different from other different languages. Various knowledge-based, linguistic, and style based, network analysis based fake news detection methods have been developed for fake news detection for English and other language-based texts and the other. In the case of Afaan Oromo and other morphologically rich languages, fake news detection is a new research area, as very few works have been published.

2. RELATED WORK

The intentional user misleading information is called as fake news [5]. Here we presenting the fake news identification literature focusing on different languages.

Table 1: Summary of Literature Review

Research Work	Fake News Detection Approach	Features Extraction	Machine Learning methods used	Source of Datasets	Accuracy
[5]	Knowledge-Based	Knowledge graphs	Vector space	BBC, Sky, Independent news	0.80 F1 scores
[6]	Linguistic and Style Based	Linguistic features	SVM classifier	Collect manual and crowdsourced	76%
[7]	Linguistic and Style Based	Surface-level linguistic patterns	CNN model	POLITIFACT.COM	27%
[8]	Linguistic and Style Based	n-gram term frequency	SVM classifier	Amazon Mechanical Turk	86%
[9]	Linguistic and Style Based	Lexical similarity and Cosine measure	Linear SVC Classifier	Fake.Br corpus	88%
[10]	Linguistic and Style Based	Lexical and syntactic features	SVM and NB models	Used [8] datasets	84%
[11]	Linguistic and Style Based	Headlines with corresponding article bodies	LR classifier	Fake News Challenge (FNC1)	89.59%
[12]	Linguistic and Style Based	TF &TF-IDF	Linear SVC Classifier	Reuters.com and kaggle.com	92%
[13]	Linguistic and Style Based	Word frequencies	Bernoulli Naive Bayesian	American news articles	89.09%
[14]	Linguistic and Style Based	Elaboration Likelihood Mode	SVM classifier	Buzzfeed, Burfoot and Baldwin	71%
[15]	Network Analysis Based	News content and social context		Facebook messenger chat-bot	81.7%
[16]	Network Analysis Based		Logistic Regression and Boolean crowdsourcing	15,500 Facebook posts	99%
[17]	Network Analysis Based	Contextual approach	Logistic Regression and harmonic Boolean label crowdsourcing	Facebook posts	90%

3. METHODOLOGY

Here we discussed the steps we followed to conduct classification of Afaan Oromo fake news.

3.1. Proposed Model Architecture

The proposed model classification architecture on social media posted components, which are collected the datasets, cleaning the dataset, partitioning the dataset, feature engineering, choosing the right algorithm, mathematical models and methods and results. The following is the proposed model architecture for classification of Afaan Oromo fake news posted on social media.

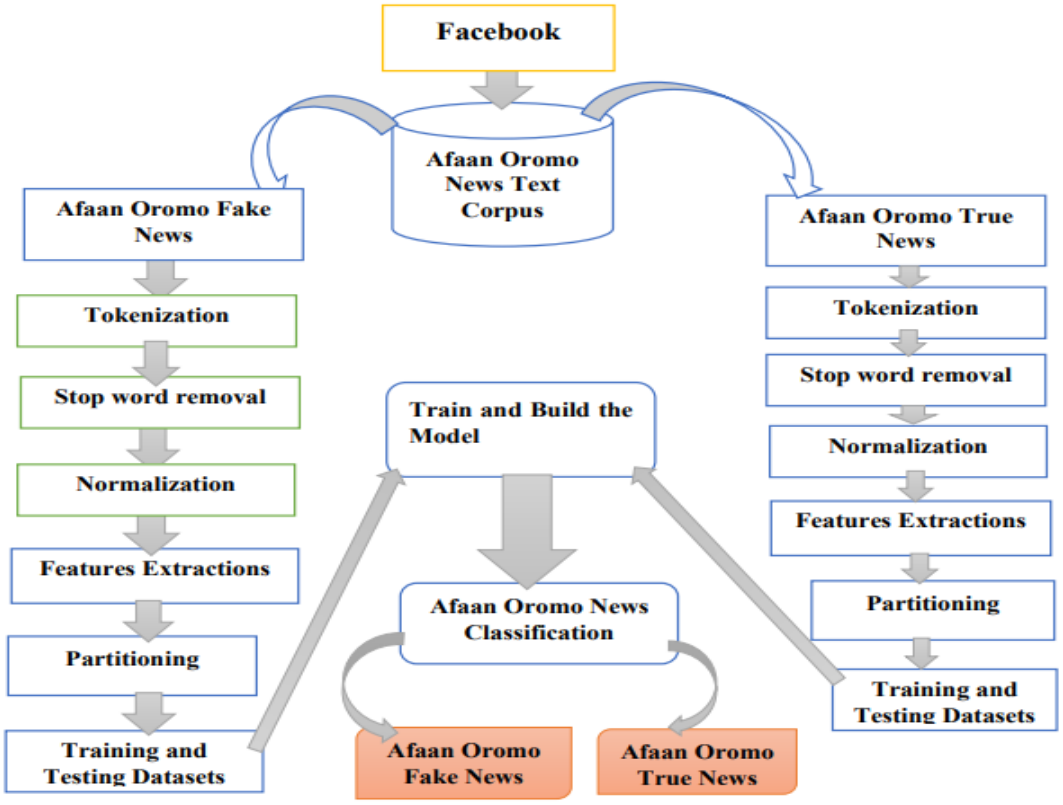


Figure 1: Proposed Model Architecture

3.2. Dataset Collection

To make comprehensive datasets we were collected with language experts, the true news datasets were collected from Ethiopian media which are known to be reputable and have permission, from Oromia Broadcasting Network (OBN) and Fana Broadcasting Corporate (FBC) their Facebook pages. We collected total real news 623, 312 is from OBN and 311 from FBC and fake news around 640 from different Facebook accounts and Facebook pages.

Table 2: Statistics of true news and fake news.

Type of News	Label	Source	Amount
Afaan Oromo True (Dhugaa) news	Dhugaa	OBN, FBC	623
Afaan Oromo False (Soba) News	Soba	From Facebook, different accounts	640

3.3. Data Preprocessing

We have conducted our experiments in two different ways. The first one is after preprocessing the datasets which is Afaan Oromo news text and the second one is experiment before preprocessing the datasets which is Afaan Oromo news text to extract feature engineering like uppercase letter usage, punctuation mark usage and news articles /documents lengths in both datasets.

3.3.1. Experiments After Preprocessing Corpuses

3.3.1.1. Remove Punctuation

The step is started by removing punctuation from all Afaan Oromo news text datasets. To perform it, the python program read sall the datasets sequentially and generates the special characters and punctuations. These can be removed. Then, the cleaned datasets are stored in the "body_clean" variable for next level processing. Figure 2 shows python procedure to remove punctuations that exist in Afaan Oromo news datasets.

```
#Remove punctuation
import string
string.punctuation
def remove_punctuation(txt):
    txt_nopunct = "".join([c for c in txt if c not in string.punctuation])
    return txt_nopunct
df['body_clean'] = df['Body'].apply(lambda x: remove_punctuation(x))
df.head()
```

Figure 2: Python code for removing punctuations

3.3.1.2. Tokenization and Normalization

The next step is tokenization and normalization, the python implementation reads all news in the datasets sequentially and generate each word and numbers. These words are split based on white space. Afaan Oromo uses Latin script for textual need and a whitespace character to separate words from each other in text documents. For example, **ñEjensiin Galmeessa Ragaa Buôuuraa Barnoota Oromiyaa waliin taôuun ragaa galmeessuun ragaa kennuuf ta'uu ibsaò. In this sentence, ñGoaro ñRagaaò, ñBuôuuraaò, ñHawaasummaaò, ñOromiyaa ñwaliinò, ñtaôuunò, ñragaò, ñgalmeessuunò, ñragaa kennuufò, ñta'uu ibsaòa** using a space character and example for normalization; **OROMIYAA KOO / ómy** with **oromiyaa koo. / ómy oromiaô**

The list of words or tokens is stored in the "*body_clean_tokenized*" variable for further processing. In the Afaan Oromo the letters can exist as upper/lower case and/or a mix of the two in text. In this study, all the alphabets in the text are changed to lower case to give common representation through out the datasets. This is accomplished as part of tokenization which can be done by a python function *"body_clean_tokenized. Lower ()"*. Figure 3 shows the code for tokenization and normalize terms in Afaan Oromo news corpus.

```
def tokineze(txt):
    tokens = re.split('\W+', txt)
    return tokens
df['body_clean_tokenized'] = df['body_clean'].apply(lambda x: tokineze(x.lower()))
df.head()
```

Figure 3: Python code for tokenization and normalization

3.3.1.3. Stop Word Removal

The subsequent procedure is to remove the stop words from the tokenized terms list.The stop word list from the file which is saved as "*stopwords.txt*" and gets compared with the tokenized terms with the stop word list. When the tokenized term is same as stop word in the stop word list, it removes from *body_clean_tokenized*. Figure 4 reveals the code to remove stop word from tokenized terms Afaan Oromo news corpuses.

```
sw=open("E:\\4th Semister\\Msc Thesis\\Dataset\\Afaan_Oromo_Stopwords.txt", 'r')
stopwords=sw.read()
def remove_stopwords(txt_tokenized):
    txt_clean = [word for word in txt_tokenized if word not in stopwords]
    txt_clean = " ".join([word for word in txt_tokenized if word not in stopwords])
    return txt_clean
df['body_no_sw'] = df['body_clean_tokenized'].apply(lambda x: remove_stopwords(x))
df.head()
```

Figure 4: Python code for removing stop words

3.4. Feature Extraction

Feature extraction is the construction or extraction of features from the dataset [18]. Feature extraction is the process of mapping from textual data to real-valued vectors. In this paper, we extract feature by using TF and TFIDF to propose a model that classifies Afaan Oromo fake news and N-gram model is used.

3.4.1. Term Frequency

The word count is used to calculate the term frequency by which similrity between documents can be identified. An equal length vector which contains the word count can be used represent the document.Then, the vector gets normalized to lead to the sum of its elements will add to one. Each word count is then converted into the probability of such word existing in the documents [12].

$$\text{TF (t)} = \frac{\text{No. of times term t appears in the document}}{\textit{Total number of terms in the document}} \quad [17]$$

3.4.2. Term Frequency-Inverse Document Frequency

Term frequency-inverse document frequency (TF-IDF) represents the weight value which is used in information retrieval, natural language processing and gives a statistical measure to evaluate the importance of a word in a document collection or a corpus [12] [17].

$$\text{TF (t)} = \frac{\text{No. of times term t appears in the document}}{\textit{Total number of terms in the document}} \quad [17]$$

$$\text{IDF (t)} = \frac{\text{logE (Total number of documents)}}{\textit{Number of documents with term t in it}} \quad [17]$$

$$TF{-}IDF = TF * IDF$$

3.4.3. N-gram Based Model

For document classification tasks, the bag-of-words or the n-gram model is mainly used for extracting features from a text document into a fixed-size vector representation [19]. N-gram is feature identification and analysis modelling used in language modeling and NLP fields [20] [12]. In terms of text classification, n-gram language model proved its success in language and non-language scripts such as music.[20]. N-gram is a sequence of words, bytes, syllables, or characters. An N-gram model captures spatial information by storing the occurrences of n words appearing in sequence in the document [19].

For example, the word-based n-gram for the below sentence is:

"OBN Intarnaashinaal Istuudiyoo magaalaa Mineesotaatti eebbisisee."

Unigram: OBN, Intarnaashinaal, Istuudiyoo, magaalaa, Mineesotaatti, eebbisisee.

Bigram: OBN Intarnaashinaal, Intarnaashinaal Istuudiyoo, Istuudiyoo magaalaa, magaalaa Mineesotaatti, Mineesotaatti eebbisisee.

Trigram: OBN Intarnaashinaal Istuudiyoo, Intarnaashinaal Istuudiyoo magaalaa, Istuudiyoo magaalaa Mineesotaatti, magaalaa Mineesotaatti eebbisisee.

Quad Grams: OBN Intarnaashinaal Istuudiyoo magaalaa, Intarnaashinaal Istuudiyoo magaalaa Mineesotaatti, Istuudiyoo magaalaa Mineesotaatti eebbisisee.

3.5. Classification Techniques

Machine learning algorithms create a mathematical approaches depend on datasets, referred as training data, to form predictions without being explicitly programmed to perform the task. It helps us find patterns in data patterns, we then use to make predictions about new data points. In our studies machine learning helps us find feature and patterns from the news datasets, learn that features and patterns like word occurrence in both datasets, make predictions about new classes. To perform the predictions, the algorithm is trained on the labeled news article dataset and gives the desired output i.e. fake or true news.

3.5.1. Naïve Bayes Algorithm

Naive Bayes (NB) is based on Bayes' probability theorem. Naive Bayes classifier is basically used for high dimensional text categorization. Some examples are email filtration, opinion analysis, and classifying news articles into true news and false news. We define the collected posts data PD and class of data (CDx) which x is real and fake. The probability of post data PD in the class CDx can calculate as follow [21].

$$P(CDx|PD) = \frac{P(PD|CDx) * P(CDx)}{P(PD)} \quad [21]$$

3.5.2. Support Vector Machine Algorithm

A support vector machine (SVM) is a hyper-plane (n-dimensional space) creating mechanism which takes labled training data as input and outputs an optimal hyper-plane which classifies new examples [21] [13] [17]. It can be used for both classification and regression purposes. SVMs are widely used to address classification problems [22].

3.5.3. K-Nearest Neighbors Algorithm

K-Nearest Neighbors (KNN) is known as a simple and effective classifier of text categorization [23]. K-nearest neighbor's algorithm is used for both classifications as well as regression predictive problems [24].

3.6. Evaluation

To evaluat the performance the metrics precision, recall, F-measure and accuracy are used [25]. The precision, recall, F-measure, and accuracy of the classifier are calculated by formula as the following equation [21].

$$\text{Precision} = \frac{True\ Positive}{(True\ Positive + False\ Positive)} \quad [21]$$

$$\text{Recall} = \frac{\text{True Positive}}{(True\ Positive + False\ Negative)} \quad [21]$$

$$\text{F- Measure} = \frac{2 * \text{Precision} * \text{Recall}}{Total\ number\ of\ terms\ in\ the\ document} \quad [21]$$

$$\text{Accuracy} = \frac{\text{True Positive + True Negative}}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad [21]$$

Accuracy calculates the similarity between predicted false and real false. Precision calcualates the fraction of detected fake news that is really fake.Recall is used to measure the sensitivity or the fraction of annotated false news articles that are predicted to be false news. F1 is used to mix precision and recall, to provide a general prediction performance for false news detection. The higher the Precision, F1, Recall, and Accuracy the better the performance [25].

4. RESULTS AND DISCUSSIONS

4.1. Experimental Results

Here we describe the experimental evaluation of our proposed model. Obviously, all texts made up of alphabetic, characters, terms, words, sentences, paragraphs and numeric characters. The Afaan Oromo news also the combination of terms and words, so our algorithm depends on word level to classify whether the news is true or false by using a bag of words, TF, TFIDF features. Afaan Oromo fake news always contain and characterized by emotion words, insult words, hate words and negative words and terms.

4.1.1. Experiment by Naïve Bayesian Classifier

We examined different features extraction methods like a bag of words, TF and TF-IDF for Afaan Oromo fake news classification and diversified the size of the n-gram from n=1-4. The number of features also changed to range from from 5000 to 50,000. The calculated results for different classification algorithms are shown in Tables 3-5.

In this experiment, the term frequency counts words found in both datasets to find the similarity between documents. Each document is represented by a unique length vector which contains the words counts. Then, each vector is normalized to result the sum of its elements will add to one. Each word count is then converted into the probability of such word existing in the documents.

```
Confusion Matrix for NB:----------
[[94  6]
 [ 0 58]]
Classification_report
              precision    recall  f1-score   support

           0       1.00      0.94      0.97       100
           1       0.91      1.00      0.95        58

    accuracy                           0.96       158
   macro avg       0.95      0.97      0.96       158
weighted avg       0.97      0.96      0.96       158

Accuracy of MNB within TF feature extraction with Unigram:---------------- 0.9620253164556962
F1-Score:---------------------------------------------------------------- 0.9508196721311475
```

Figure 5: Classification report for MNB

The following table 3 shows that the naïve Bayesian classifier. In this experiment, the naïve Bayesian algorithms achieve 96.2% good accuracy within all feature size with unigram in term frequency. But in the bag of words information, the structure or order of words in the document is discarded. The model is only concerned with whether known words occur in both datasets or not in the datasets. Term frequency-inverse document frequency gives a statistical measure to evaluate the significance of a word in datasets. In the following tables 3 term frequency-inverse document frequency achieves 94.9% accuracy with unigram at 5000 feature sizes. Increasing the n-gram size lowers the accuracy of the system by using this classifier. The unigram posits that each word occurrence in the corpuses is independent of all other word occurrences make dictionary and vocabulary. As a whole naïve Bayesian classifier achieves promise results within unigrams.

Table 3: Accuracy predicted by Naïve Bayesian classifier. The second row shows the size of the features. Accuracy values are in percent.

N-gram Size	Term Frequency			Term Frequency Inverse Document Frequency		
	5000	10,000	50,000	5000	10,000	50,000
Unigram	96.2	96.2	96.2	94.9	92.4	91.1
Bigram	93.7	93.0	93.0	91.8	87.9	74.1
Trigram	78.5	79.1	81.6	76.6	74.1	65.8
Four gram	71.5	72.2	75.9	70.9	68.4	64.6

4.1.2. Experiment by K-Nearest Neighbor Classifier

The following is accuracy result obtained KNN classifier.

```
Confusion Matrix for KNN:----------
[[94  9]
 [ 0 55]]
Classification_report
              precision    recall  f1-score   support

           0       1.00      0.91      0.95       103
           1       0.86      1.00      0.92        55

    accuracy                           0.94       158
   macro avg       0.93      0.96      0.94       158
weighted avg       0.95      0.94      0.94       158

Accuracy of KNN within TFIDF feature extraction with Unigram:---------------- 0.9430379746835443
F1-Score:-------------------------------------------------------------------- 0.9243697478991597
```

Figure 6: Classification report for KNN

The following table 4 show the k-nearest neighbor classifier. In this experiment, the k-nearest neighbor algorithms achieve 42.4% accuracy with unigram on 5000 and 10,000 feature size and 94.3% with term frequency-inverse document frequency within unigram on all feature sizes. This also, as increasing the n-gram size lowers the accuracy of the system in term frequency-inverse document frequency feature. The unigram posits that each word occurrence in the corpuses is independent of all other word occurrences make dictionary and vocabulary.

Table 4: Accuracy predicted by K-nearest neighbor classifier. The second row shows the size of the features. Accuracy values are in percent.

N-gram Size	Term Frequency			Term Frequency Inverse Document Frequency		
	5000	10,000	50,000	5000	10,000	50,000
Unigram	42.4	42.4	41.8	94.3	94.3	94.3
Bigram	40.5	40.5	40.5	91.1	87.9	89.9
Trigram	40.5	40.5	40.5	40.5	65.8	69.6
Four gram	40.5	40.5	40.5	74.1	59.5	60.1

4.1.3. Experiment by Support Vector Machine Classifier

The below is accuracy obtained by support vector machines classifier.

```
Confusion Matrix for SVM:----------
[[94  6]
 [ 0 58]]
Classification_report
              precision    recall  f1-score   support

           0       1.00      0.94      0.97       100
           1       0.91      1.00      0.95        58

    accuracy                           0.96       158
   macro avg       0.95      0.97      0.96       158
weighted avg       0.97      0.96      0.96       158

Accuracy of SVM within TFIDF feature extraction with Unigram:---------------- 0.9620253164556962
F1-Score:-------------------------------------------------------------------- 0.9508196721311475
```

Figure 7: Classification report for SVM

The following Tables 5 shows the support vector machine classifier. In this experiment, the support vector machine algorithms achieve 60.8% accuracy by term frequency with unigram on 5000 feature size and achieve 96.2% accuracy by term frequency-inverse document frequency feature within all features sizes. This also, as increasing the n-gram size lowers the accuracy of the system in term frequency-inverse document frequency feature. The unigram posits that each word occurrence in the corpuses is independent of all other word occurrences make dictionary and vocabulary.

Table 5: Accuracy predicted by Support Vector Machine classifier. The second row shows the size of the features. Accuracy values are in percent.

N-gram Size	Term Frequency			Term Frequency Inverse Document Frequency		
	5000	10,000	50,000	5000	10,000	50,000
Unigram	60.8	59.5	59.5	96.2	96.2	96.2
Bigram	59.5	59.5	59.5	89.2	86.0	75.9
Trigram	59.5	59.5	59.5	80.4	74.0	67.0
Four gram	59.5	59.5	59.5	74.5	69.6	66.5

4.2. Feature Engineering

4.2.1. Experiments Before Preprocessing Corpuses

We were conducted our experiments before preprocessing which is Afaan Oromo news text extracted by feature engineering like uppercase letter usage, punctuation mark usage and news articles /documents lengths in both datasets.

ly	label	No.Uppercase letter in both corpuses by %
...	Dhugaa	3.450656
...	Soba	7.037037
...	Dhugaa	2.677376
...	Soba	4.285714
...	Dhugaa	5.274725
...	Soba	0.342466
...	Dhugaa	1.696970
...	Soba	7.734807
...	Dhugaa	5.578947
...	Soba	11.046512
...	Dhugaa	4.236006
...	Soba	4.573439
...	Dhugaa	2.071346
...	Soba	6.635071
...	Dhugaa	3.582090
...	Soba	6.027397
...	Dhugaa	4.628633
...	Soba	3.084833
...	Dhugaa	2.207131
...	Soba	4.487179

Figure 8: Number uppercase letter in both corpuses by percent

4.2.2. Uppercase Usage

Afaan Oromo fake news always have lower in quality writing, like all propaganda, they add special character to pursue the people. Capital letter may necessary at start of sentences, name of place and name

of person but the fake news producers capitalize whether first letter for all words in the news or capitalize all parts of that news, this shows fake news and low quality of writing. In the following, we were show by comparing the capital letter usage in both corpuses by percentage and histogram charts.

In the above Figure 8 shows the uppercase usage by percentage for both Afaan Oromo true news and Afaan Oromo false news. The Afaan Oromo false (Soba) news have more percentage than Afaan Oromo true (Dhugaa) news by the usage of uppercase letters.

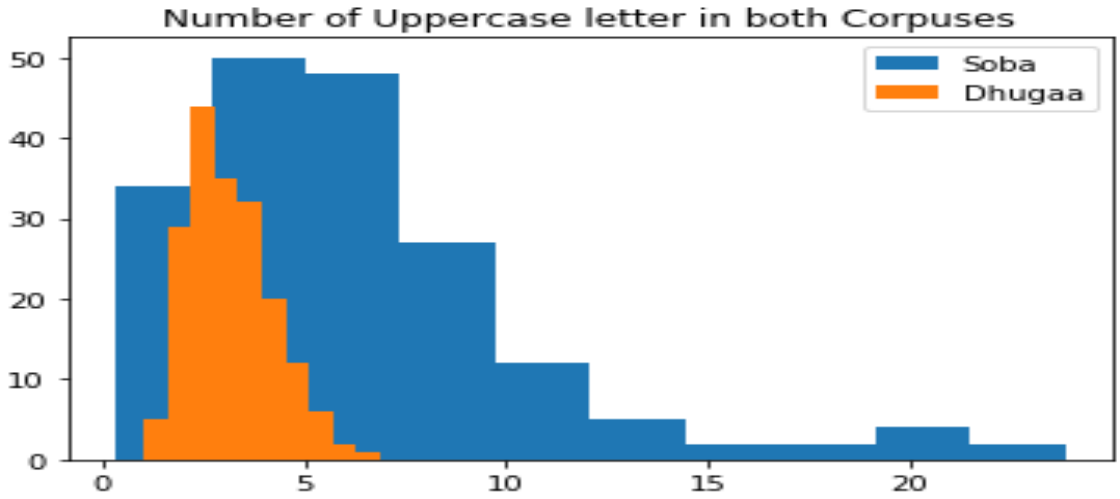


Figure 8: Uppercase letter usage in both corpuses

In the above Figure 9 shows the capital letter usage by histogram charts for both Afaan Oromo true news and Afaan Oromo false news. The fake news writer uses uppercase letter as they want in their news. The above Figures 9 illustrate the Afaan Oromo false (Soba) news uses more capital letter than Afaan Oromo true (Dhugaa) news.

ly	label	No.punctuation in both corpuses by %
...	Dhugaa	0.966184
...	Soba	4.537037
...	Dhugaa	0.803213
...	Soba	1.785714
...	Dhugaa	1.978022
...	Soba	4.452055
...	Dhugaa	2.121212
...	Soba	1.657459
...	Dhugaa	3.684211
...	Soba	4.069767
...	Dhugaa	1.966717
...	Soba	2.198769
...	Dhugaa	1.611047
...	Soba	10.426540
...	Dhugaa	1.940299
...	Soba	0.821918
...	Dhugaa	1.506997
...	Soba	2.570694
...	Dhugaa	1.018676
...	Soba	1.923077

Figure 10: Punctuation usage by percentage in both news

4.2.3. Punctuation Usage

Irregular punctuation mark usage and more frequency of special characters show false news. In fake news there always more special characters and more frequently used punctuation marks. The greater number of punctuation marks always found in fake news and are highly indicative of the presence of fake news. The fake news writer uses more punctuation mark without any limitation and their news is not accurate and clear.

The above Figures 10 shows the punctuation mark usage by percentage for Afaan Oromo true news and Afaan Oromo false news. The Afaan Oromo false news have more percentage than Afaan Oromo true news by the usage of punctuations. The Afaan Oromo true news has small punctuation marks.

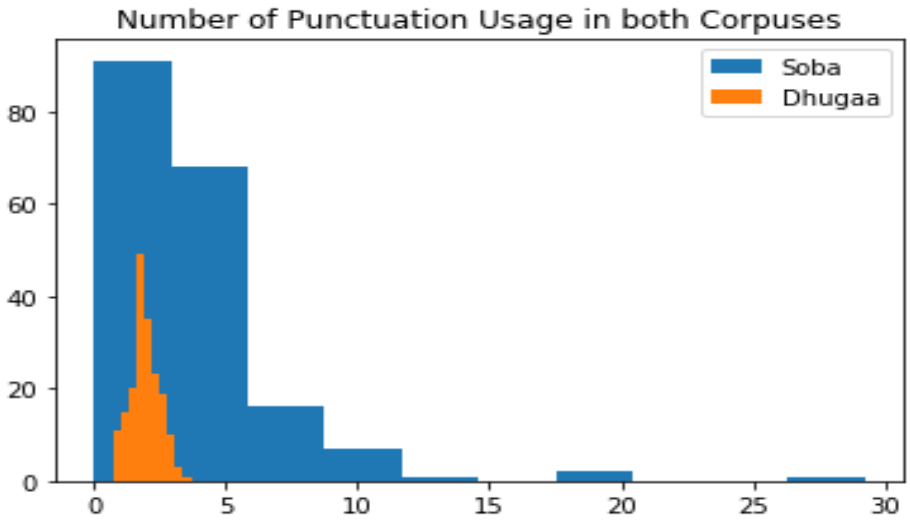


Figure 9: Punctuation usage in both news

The above Figures 11 illustrate the usage of punctuation by histogram charts for Afaan Oromo true news and Afaan Oromo false news. The Afaan Oromo false news uses more punctuation than Afaan Oromo true news. The Afaan Oromo true news uses small punctuation and compiled while produced.

...ty	label	News lengths in both corpuses by characters
...	Dhugaa	1449
...	Soba	1080
...	Dhugaa	1494
...	Soba	280
...	Dhugaa	910
...	Soba	292
...	Dhugaa	1650
...	Soba	181
...	Dhugaa	950
...	Soba	172
...	Dhugaa	661
...	Soba	1137
...	Dhugaa	869
...	Soba	211
...	Dhugaa	670
...	Soba	365
...	Dhugaa	929
...	Soba	389
...	Dhugaa	589

Figure 12: News article lengths in both corpuses

4.2.4. News Length

A long news is a good indicator of quality investigative journalism. To know their lengths, we were counting the number of characters in both Afaan Oromo news and compare with each other.

In the above Figure 12 illustrate the length of news in both datasets. As a general Afaan Oromo true news has greater number of character and words than Afaan Oromo false news.

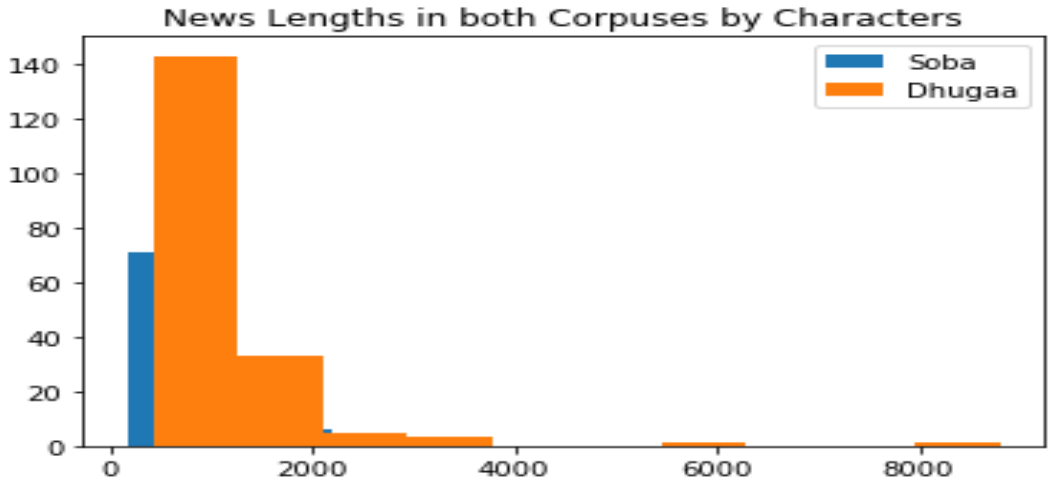


Figure 10: News article lengths in both datasets

In the above Figure 13 illustrate the length of news by histogram charts in both datasets. Afaan Oromo fake news has smaller number character and words than Afaan Oromo true news as shown on the figures. The fake news writer uses small character and words while produce the news to post quickly on Facebook.

As experimented on the above, naïve Bayesian and support vector machine achieved better results. We obtained 96.2% accuracy with naïve Bayesian and support vector machine. It is observed that the resulted best accuracy within unigram analyzer with features. Also, it can be understood that the increasing the n-gram size results in reduced accuracy with all the classifiers. In naïve Bayesian classifier we obtained more accuracy with term frequency feature extraction with unigram level. In the bag of words information, the order of words is ignored. The model concerns with the occurance of wrods in either both data sets or none of them. The term frequency inverse document frequency achieved good results in support vector machines, this illustrate that the support vector machine good classifier than naïve Bayesian algorithm. TF performed better than TF-IDF in naïve Bayesian classifier on all n-gram with all feature values, TF-IDF performed better than TF in support vector machine classifier on unigram with all feature values.

In support vector machine the term frequency scores 60.8% on unigram. The left one scores 59.5% within all feature sizes with term frequency feature. The lowest accuracy of 40.5% was achieved using k-nearest neighbor with all N-gram words except unigram with term frequency features on all feature values.

5. CONCLUSIONS AND RECOMMENDATIONS

Text classification is that the task of assigning predefined classes to free-text documents depend on their content. Classification of Afaan Oromo fake news is a type of text classification tasks. Afaan Oromo news is national and inter national news that is generated about business, about economic, opinions, political, sports news and television listings. These Afaan Oromo news categorized into Afaan Oromo true (Dhugaa)

news and Afaan Oromo fake (Soba) news and posted on social media like Facebook. Checking news reliability is a process of determining whether a particular news report is truthful or false. Afaan Oromo fake news is news that are intentionally prepared by fake news producers to get tangible and intangible benefits. This paper study focused on Afaan Oromo fake news classification on social media by using a machine learning approach. In this paper, we focused on the problem of classifying Afaan Oromo fake news that posted of Facebook by machine learning approach. The news are the combination of multimedia, terms and words, our studies focused on word level to classify whether the news true or false by using a bag of words, TF, TFIDF features within n-gram analysis. On the other hand, the absence of standard sets corpus and evaluation tool for Afaan Oromo language was a limitation, hence we were collected small posts from Facebook for experimentation. For this experiment, we have collected 623 true news from OBN and FBC Facebook pages and 640 false news from different Facebook accounts. Finally, to demonstrate the effectiveness of the system accuracy and F1-score evaluation metrics were conducted and F1-score of 95.0% was obtained from the experiment and as a total 96.2% accuracy achieved within naïve Bayesian classifier and support vector machines. Afaan Oromo fake news classification requires standard data set to make experimentation. But such is not yet developed which address the future wotk emphasis and the present work focuses only news texts only and the others such as video, audio, graphics, pictures needs to be studied in a great manner.

REFERENCES

[1] www.wikipedia.org, "wiki," November 2019. [Online]. Available: https://en.wikipedia.org/wiki/Fake_news.

[2] B. Martens, L. Aguiar, E. Gomez-Herrera and F. Mueller-Langer, "The digital transformation of news media and the rise of disinformation and fake news - An economic perspective; Digital Economy Working Paper," Research Gate, European Commission, Seville, Spain:, 2018.

[3] Abeselom and D. Kiros, "The impacts of fake news on peace and development in the world: the case study of Ethiopia," *International Journal of Current Research*, vol. 10, no. 07, pp. 71356-71365, 2018.

[4] E. M. Okoro, B. A. Abara, A. O. Umagba, A. A. Ajonye and Z. S. Isa, "A Hybrid Approach to Fake News Detection on Social Media," *Nigerian Journal of Technology*, vol. 37, no. 2, p. 454 – 462, 2018.

[5] J. Z.Pan, S. Pavlova, C. Li, N. Li, Y. Li and J. Liu, "Content Based Fake News Detection Using Knowledge Graphs," in *International Semantic Web Conference*, UK and China, 2018

[6] V. P´erez-Rosas, B. Kleinberg, A. Lefevre and R. Mihalcea, "Automatic Detection of Fake News," in *International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA, 2018.

[7] W. Y. Wang, *"Liar, Liar Pants on Fire": A New Bench mark Dataset for Fake News Detection*, Santa Barbar: University of California, 2017.

[8] M. Ott, Y. Choi, C. Cardie and J. T. Hancock, "Finding Deceptive Opinion Spam by Any Stretch of the Imagination," Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Oregon, 2011.

[9] R. A. Monteiro, R. L. S. Santos, T. A. S. Pardo, T. A. d. Almeida, E. E. S. Ruiz and O. A. Vale, "Contributions to the Study of Fake News in Portuguese: New Corpus and Automatic Detection Results," in *Springer*, Brazil, 2018.

[10] S. Shojaee, M. A. A. Murad, A. B. Azman, N. M. Sharef and S. Nadali, "Detecting Deceptive Reviews Using Lexical and Syntactic Features," in *2013 13th International Conference on Intellient Systems Design and Applications*, Bangi, Malaysia, 2013.

[11] P. Bourgonje, J. M. Schneider and G. Rehm, "From Clickbait to Fake News Detection: An Approach based on Detecting the Stance of Headlines to Articles," *Proceedings of the 2017 EMNLP Workshop on Natural Language Processing meets Journalism*, vol. Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism, p. 84–89, 2017.

[12] H. Ahmed, I. Traore and S. Saad, "Detection of Online Fake News Using NGram Analysis and Machine Learning Techniques," in *Springer International Publishing AG*, Victoria, Windsor, 2017.

[13] P. Rumman and M. Svärd, *Combating Disinformation Detecting fake news with linguistic models and classification algorithms*, Philip: Rumman, Philip, 2017.

[14] B. D.Horne and S. Adalı, "This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News," in *(Eleventh International AAAI Conference on Web and Social Media*, 110 8th Street, Troy, New York, USA, 2017

[15] M. L. D. Vedova, E. Tacchini, S. Moret, G. Ballarin, M. DiPierro and L. d. Alfaro, "Automatic Online Fake News Detection Combining Content and Social Signals," in *2018 22nd Conference of Open Innovations Association (FRUCT)*, Jyvaskyla, Finland, 2018.

[16] E. Tacchini, G. Ballarin, M. L. D. Vedova, S. Moret and L. d. Alfaro, "Some Like it Hoax: Automated Fake News Detection in Social Networks," School of Engineering, University of California, Santa Cruz, 2017.

[17] T. Granskogen, *Automatic Detection of Fake News in Social Media: using Contextual Information*, Norwegian University of Science: NTNU, 2018.

[18] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N.Ritcher and H. A. Najada, "Survey of review spam detection using machine learning techniques," *Springer, Journal of Big Data*, vol. 23, no. 3, pp. 1-24, 2015.

[19] A. Ågren and C. Ågren, *Combating Fake News with Stance Detection Using Recurrent Neural Networks*, Gothenburg, Sweden 2018: UNIVERSITY OF GOTHENBURG, 2018.

[20] H. Ahmed, "Detecting Opinion Spam and Fake News Using N-gram Analysis and Semantic Similarity," University of Victoria Library, Victoria, 2018.

[21] S. Aphiwongsophon and P. Chongstitvatana, "Detecting Fake News with Machine Learning Method," in *IEEE*, Chiang Rai, Thailand, Thailand, 2018.

[22] S. N. V. A. N. D. D. H. Kushal Agarwalla, "Fake News Detection using Machine Learning and Natural Language Processing," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7, no. 6, pp. 844-847, 2019.

[23] A. H. Aliwy and E. H. A. Ameer, "Comparative Study of Five Text Classification Algorithms with their Improvements," *International Journal of Applied Engineering Research*, vol. 12, no. 14, pp. 4309-4319, 2017.

[24] www.tutorialspoint.com, "machine learning with python knn algorithm finding nearest neighbors," 2019. [Online]. Available: https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_knn_algorithm_finding_nearest_neighbors.htm.

[25] K. Shu, A. Sliva, S. Wang, J. Tang and H. Liu, "Fake News Detection on Social Media: A Data Mining Perspective," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22-36, 2017

Argument Mining from Amharic Argumentative Texts using Machine Learning Approach

Alemu Kumilachew[1]*, Mikru Lake[2], Debela Tesfaye[2]

[1]*Bahir Dar University, Institute of Technology, Bahir Dar, Ethiopia*
[2]*Jimma University, Jimma, Ethiopia*
**Corresponding author, e-mail: alemupilatose@gmail.com*

ABSTRACT

In the world of information overload, extracting argument relation through a manual method is time-consuming, knowledge-intensive, and prone to bias. Argument mining is an emerging science that deals with the automatic identification and extraction of arguments along with their relation from large, unstructured data that are useful for reasoning engines and computational models. Argument mining incorporates two main argument pipelines: Argument component extraction and relation predictions. There have been many efforts tried on argument relation prediction for English and other European languages. In this study, we design and implement an argument relation prediction for Amharic language using MLP, Naïve Bayes, and SVM algorithms. The study used 815 argumentative sentences to evaluate argument relation prediction. The evaluation of an experiment that is conducted using discourse marker, propositional semantic similarity, and a combination of these approaches results in the highest weighted average F-score of 68%, 84%, and 88% respectively using SVM. This shows that a combining approach and SVM classifier is preferable for the Amharic argument relation prediction task.

Keywords: *Argument, Argument mining, Argument Relation Prediction, Word2vec, Discourse Marker, Proposition Semantic Similarity*

1. INTRODUCTION

In today's digital era, world activities are turned into the Internet. Social Media, weblogs, and discussion forums occupy a majority of user activities. It is believed that such platforms enabled many people to read different comments and express their opinions through a series of reasoning that generates ideas and claims to impact the political decisions; policymakers make publicize their claims and the government to find claims to make genuine decisions and design effective policy. As a result, an abundance amount of comments and feedback is found in the form of fact or opinion that holds arguments to collect and analyze from different genres. Argumentation is a process, in which arguments are constructed, evaluated, proved, or disproved [1]. It is a process of producing and evaluating arguments in the context of a discussion, dialogue, or conversation. Arguments are different from other propositions in that arguments must consist of one conclusion (claim) and one or more premises. Whereas, propositions can be either a conclusion (statements which are either true or false until proof by the ground truth or other pieces of evidence) or premises (a proposition that expresses the justification for support or against the claim).

Argument Mining (AM) is interchangeably used as argumentation mining and is also related to computational argumentation or debating technologies [2]. It is the subfields of text mining that automatically extract natural language argument components along with their relationship from a large resource of unstructured natural language texts to furnish machine-readable structured data for a

computational model of argument and reasoning engines [3]. It is considered as an extension of opinion mining that analysis people's attitudes [4]. Sentiment (opining) mining is focused on extracting or mining people's attitudes (positive, negative, or neutral) towards products, events, or people. However, Argument mining is an automatic extracting of arguments with their relationship that describes the reasons behind the views expressed. It includes people's attitudes with justification or reasons.

Argument mining pipeline has two main stages: argument component extraction (classification) and argument relation prediction [5], where each subtask is interrelated but not fully relied upon one subtask to the other subtask. This means that it is possible to perform argument relation prediction without doing argument components extraction. This study is tried to tackle the subtask of argument relation prediction for Amharic argumentative texts. It is used documents that are published in Amharic language (text) in both electronic and printed format.

2. LITERATURE REVIEW

2.1. Argument mining and Argumentation model

Argument (argumentation) mining becomes an interesting research field in computational linguistics, philosophy, psychology, and artificial intelligence. Argument mining (AM) can be interchangeably used as argumentation mining, computational argumentation, or debate technology. In spite of the fact that there are many interpretations of Argument mining [6], we have focused on more flavor definitions to computational argumentation - the processes of extracting argument components along with their relation.

The study of argumentation has long been ancient since 1960 in philosophy and dialects. Also, it spans diverse knowledge areas such as logic, linguistics, rhetorical, computer science, psychology, and argumentation theory [7]. In the area of artificial intelligence, the argumentation model is classified as Monological, Dialogical and Rhetorical model [8].

The monological model (micro-level model) is one of the widely used taxonomies of argumentation modeling for designing and implementing an argument mining system. The underlying focus of the Monological model is on the internal structure of argument components that an argument is constructed. Among the most influential on these modeling, structures include [9], Toulmin [10], Freeman [11]. On another hand in the dialogical model, argumentation is emphasized in the way arguments are connected in dialogical structures that expressed one more party or authors in dialogue form. Unlike the monological model, the dialogical model connects a set of arguments that focused on the external or macrostructure of arguments. Whereas, Rhetorical model is somehow different from the above two argumentation models in the structuring of arguments. Rather than an internal or external structure of the argument, the rhetorical argumentation model deals rhetorical pattern or scheme of the texts based on audience and persuasion intention. This means that it considers audience persuasion of argument when the structures are constructed. In this model, the argument is evaluated by judgments rather than the truth of a proposition.

2.2. Subtasks of Argument Mining

Argument mining is performed either in the form of Argument component extraction or Argument relation prediction. Argument component extraction or classification is detecting or extracting arguments based on the given target of granularity. This subtask may be included other subtasks like argumentative

sentences classification, argument component classification, and argument bounder detections [5]. Whereas, Argument relation prediction is the subtask that is used to predict or classify what relationships exist between arguments or argument components. That is support, attack, or neutral [12] [13]. This subtask is more complex and challenging because it requires a high level of knowledge understanding and reasoning issues. It uses graphic structure to represent arguments such as trees or graphs. The task is also seen as multiple relation predictions [7], this means that one more argument supports or attacks the single argument.

2.3. Approaches to Argument Mining

The most common approaches in argument mining are Rule-based and Machine learning approaches. In the rule-based approach, argument mining tasks have relied on handcrafted rules constructed by knowledge engineered, who is an expert on that domain area [14]. The rule-based approach used some predefined rules as a knowledge base or mostly relied on natural languages processing methods such as part of speech tagging, syntactic and semantic analyses, and parsing techniques that compiled with rules that follow [1].

Machine learning approach is concerned with the development of design and development of algorithms that allows a computer to learn from its experience by performing tasks on a set of examples [15]. The main objective of machine learning is to build a model that generalizes behaviors relying on data seen. These approaches have been classified into supervised, semi-supervised, and unsupervised learning approaches. This study used a machine learning approach as the data examples are found unstructured and rule-based approaches are not feasible to use in the current problem.

2.4. Common Machine Learning Algorithms

This study utilized supervised machine learning algorithms such as Support Vector Machine (SVM), Naïve Bayes Classifier, Multi-Layer Perceptron (MLP), and Artificial Neural Network (ANN).

SVM is applicable for both linearly separable and non-linearly separable problems. For non-linearly separable data problems, SVM creates a non-linear mapping function that transfer (converts) the original data points into high dimensional space and make them easily linearly separable by hyperplanes [16]. SVM uses kernel functions such as linear, polynomial, Sigmoid, and Radial Basis Function (RBF) kernels to map the given data into different spaces in which the linear hyperplane cannot do the separation. Moreover, in case the problem is not a two-class problem, SVM can apply multi-class SVM in which it has two forms: One-Verses One (OVO) and One Verse All (OVA) [16].

Naïve Bayes class classifier is a probabilistic supervised machine learning algorithm that is mainly used for clustering and classification purpose. This classifier is working on the principle of Bayes theorem which assumes with strong and naïve independent assumption [1]. Naive Bayes classifier predicts the most likely class for the given features because the classifier works on conditional probability principle and computes the posterior probability for the given events.

Artificial Neural Network (ANN) is a collection of highly interconnected processing neurons that is motivated from the concept of a biological neuron of human brains ANN computes the processing with

three basic processing layers: the input layer, the hidden layer, and the output layer where each layer has its own role [17].

MLP is one class of Artificial Neural Network, in which the flow of information is in one direction (feed-forward network). In this network architecture, there is no cyclic connection in the network. MLP is used for supervised learning problems by implementing a back-propagation algorithm [17]. It is applicable for the classification task in which there is labeled data. .

2.5. Overview of Amharic Language

Amharic language is a member of the Semitic language family and it is the second most spoken language next to the Arabic language. It is an official working language of the Federal Democratic Republic of Ethiopia [18] and is dominantly spoken in the northern and central parts of Ethiopia. It has more than 22 million first-language speakers and 4 million second speakers worldwide[1]. Even though it has a great number of speakers, resources, and applicability, the language is still categorized as an "under-resourced" language [19][20]. This study used unstructured texts in Amharic language.

3. METHODOLOGY

To classify argument relations, Amharic argument relation predictor accepts a pair of argumentative sentences and generates argumentative relation between those pair of sentences as support, attack (contradict), or neutral. To predict the relation between argument internal components, argument relation prediction passes several steps. The general system architecture of Amharic argumentative text relation prediction is described in Figure 1.

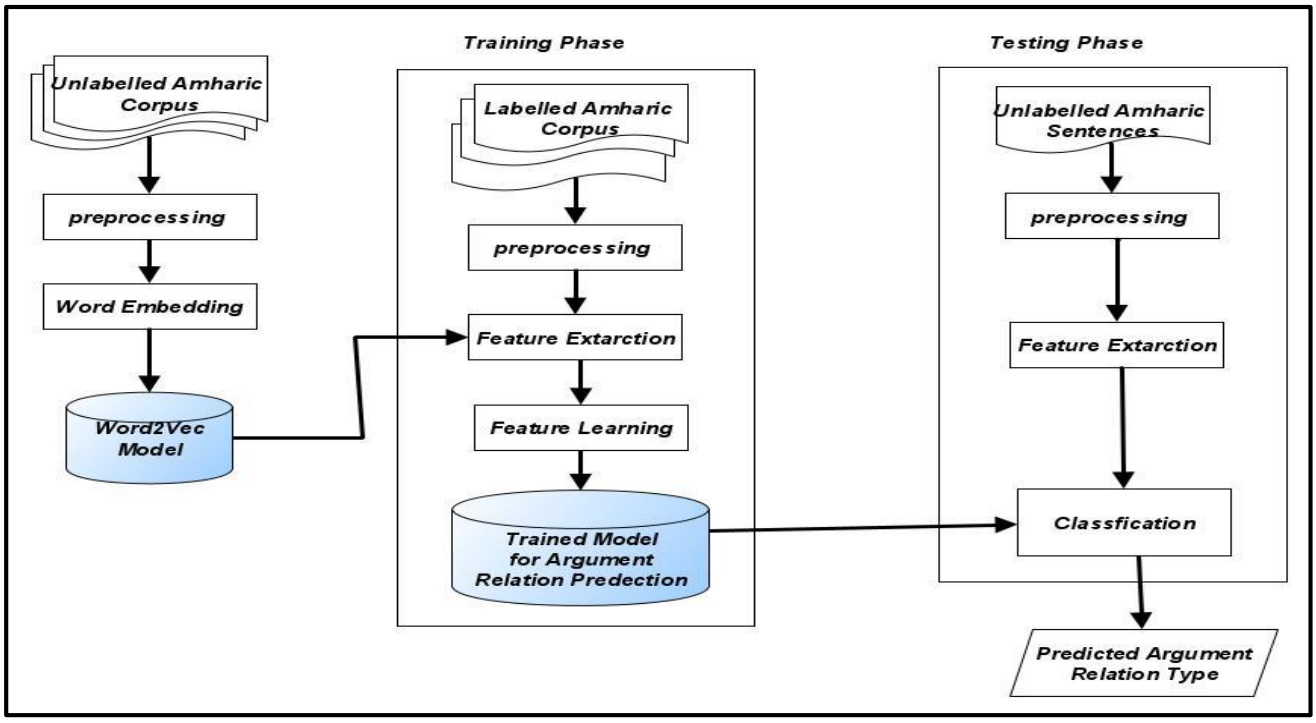


Figure 1: A proposed system architecture

[1] https://dictionary.abyssinica.com/amharic.aspx

Data collection: For successful use of a machine-learning algorithm, a good training corpus is crucial to generate a model for argument relation prediction tasks. To tackle the problem the study used two different datasets. The first data set is argumentative sentences that are used for argument relation prediction tasks. For this task, a total of 5012 argumentative Amharic corpora from a politically focused source such as Amharic Newspapers, weblogs, Facebook, and other social media, is collected. Of which 815 argumentative sentences are selected and annotated with the help of domain experts.

The second one is large unlabeled raw texts which are compiled by a former researcher in the area. Out of 724,608 sentences, 704,608 sentences are directly adopted from former researchers and 20,000 sentences are collected by the researchers. Overall, a total of 724,608 sentences (11,419,195 tokens) were collected and used in this study. This data set is mainly used for the automatic feature generation module (to develop the Word2Vec model) that is used as a look-up model for our argument relation tasks. Even though, the embedding data set is small as compared to other languages like English (e.g. Google has one billion tokens), we have got a promising result from the embedding model and perform very well for extracting propositional similarity and discourse marker features.

Text Annotation: as there is no readily available annotated data, we involved domain experts and OVA (Online Visualization of Argument) analysis tool [21] for the annotation. An OVA is an online annotation tool used to make a series of argument maps graphically, capture this graphical structure, and show it in the form of argument interchange format (AIF). To complete the annotation processes, we have used the left and right sides interfaces of the OVA analysis tool. We select the left side interface of the OVA and insert the argumentative text (containing conclusion and premises) into the rectangular box. And then the argument nodes (support or conflict between two nodes) are created on the right side of the OVA interface. Then, we saved the corresponding argument map in the Argument Interchange Format Database (AIFDB) from which the final corpus is later extracted and used in this study. Finally, the final corpus containing a total of 120 argument maps is created and made available online in the name of "ክርክር በአማርኛ"[2].

Preprocessing: In this step of preprocessing, tokenization, normalization, and morphological analysis have been performed using language-specific data preprocessing. In the case of tokenization, preprocessing has been made for both corpora. As the domain language – Amharic – is a morphologically rich language that suffers from a variation of homophone writing styles, grammatical morpheme, lexical morphemes, and so on, morphological analysis is done to normalize and reduce its various forms into its single base form. In this study, we have used a morphological analyzer developed by Gasser, named as HorneMorpho 2.5[22].

Feature Generation: In this study, we exploit propositional semantic similarity and discourse marker features to our argument relation prediction task. For this task, a Word2Vec model is selected to process a total of 12 million tokens. The model is selected as it is best for similarity task analysis for words in the given sentences or documents [23]. Word2Vec can be implemented in the form of a continuous bag of words or continuous Skip-gram. In this study, we employed word2Vec with a skip-gram model with the following parameters; Context of window size (10), size of diminutions (150), minimum count (3), number

[2] http://corpora.aifdb.org/MIK1

of iteration (10), and number of workers (4). After the training process is finalized, each word with its corresponding feature vector is stored as an output model.

***Feature Extraction*:** An automatic feature generator, Word2Vec, generates a feature vector for each word form from word embedding. In this stage, the model generated by Word2Vec is used as a lookup table – from which discourse marker and semantic similarity feature extraction have been made.

Discourse Marker Feature Extraction: Discourse markers that exist between argumentative propositions as support, attack, and neutral are constructed. First, we extract mostly used discourse markers like *ስለዚህ, ለዚህም* for support and *ቢሆንም, ነገርግን* for attack relation are manually extracted. Then, we expanded discourse marker extraction using an automatic method to find more discourse markers that are similar to (*ስለዚህ, ለዚህም*) and (*ቢሆንም, ነገርግን*) using word2vec model based on the similarity rank. Of these, the top 60 rank similar discourse markers from the word2vec model for both support and attack are extracted separately. And then, we scan again the occurrence of each discourse marker from our argumentative corpus. If the discourse marker has occurrence from our argumentative corpus, we add the discourse marker to one of the discourse marker lists (support or attack). If the discourse marker does not exist in our argumentative texts, we simply discard it. In this study, a total of 35 discourse markers are extracted for our task. Out of these, the Boolean feature and semantic feature of each discourse marker are again extracted. We used both discourse markers features for our argument relation prediction task.

Semantic Similarity Feature Extraction: For semantic similarity of argument proposition, we remove the previous discourse markers that are used as conjunctions. This can be done by performing propositional similarity using the argument map level since the argument relationship is performed on each argument level and each argument map level is composed of the same topic. To perform propositional semantic similarity, first, we load our trained Word2Vec model and then calculate each words vector by extracting the words vector from the trained model and storing these words vectors to a new wordlist. Then we compute the semantic value of each proposition by using cosine similarity for one proposition to the other proposition. Finally, the similarity score is used to train our classifier to predict argument relation tasks for our Amharic argumentative texts.

4. EXPERIMENTAL RESULTS AND DISCUSSION

A total of nine experiments have been performed using three supervised machine learning algorithms on different data (feature) sets. The first experiment is performed using discourse markers features. The second one is conducted using propositional semantic similarity features. And, the third is done by combining both propositional semantic similarity and discourse marker features along with different learning algorithms. At each experimentation, a comparison of the result of each learning algorithm has been made and the overall weighted average f-score is calculated for the learning algorithm. Finally, the result is presented using a suitable representation.

4.1. Experiment One: Argument mining using discourse features

This experiment is conducted using discourse features as input for the classifiers. For experimenting, the data set is randomly split into 70/30 splitting ratios for training and testing data sets respectively. As it is shown in Figure 2, using three learning algorithms, a SVM classifier shows better accuracy in argument relation prediction for Amharic Argumentative text.

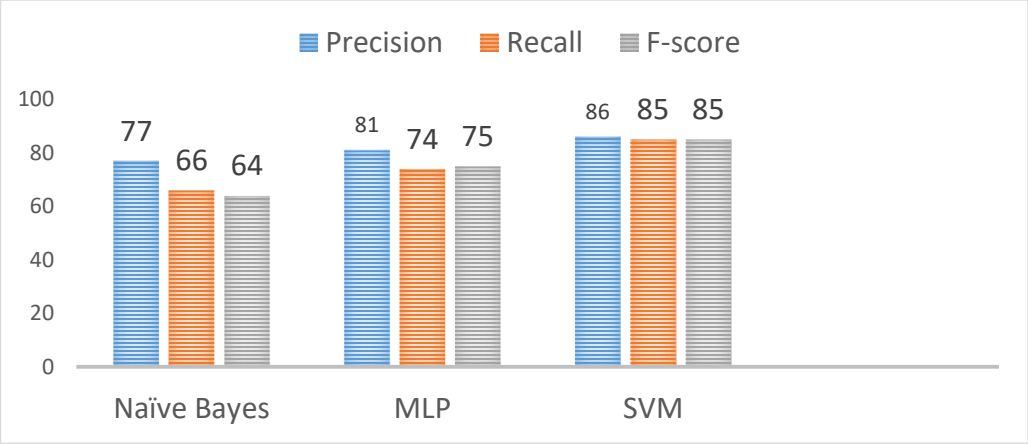


Figure 2: Prediction performances of Naive Bayes, MLP, and SVM Classifiers using discourse features

4.2. Experiment Two: Argument mining using proposition semantic similarity features

This experiment is conducted using propositional semantic similarity to see the effect of the features. The propositional semantics similarity is used to overcome implicit discourse markers.

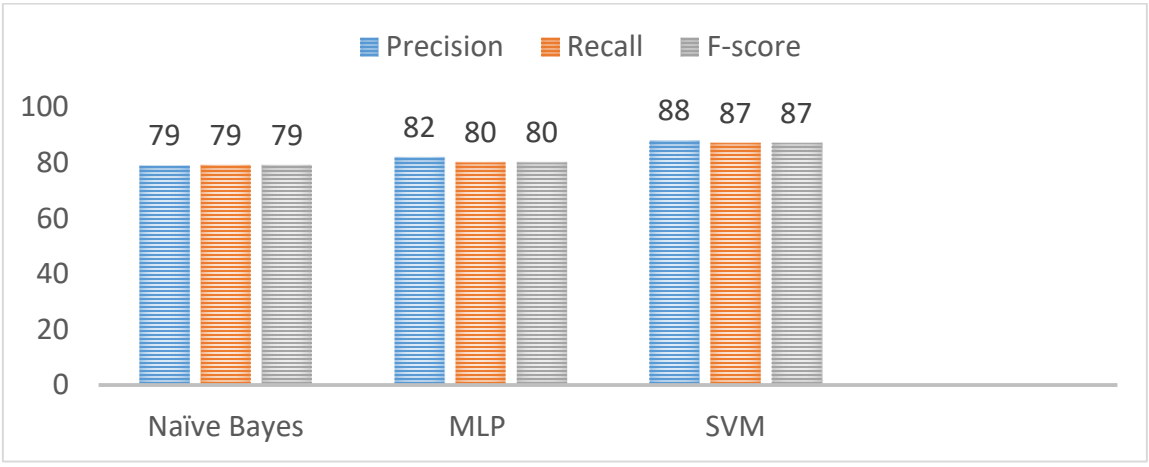


Figure 3: Prediction performances of Naive Bayes, MLP and SVM Classifiers using Proposition Semantic Similarity Features

4.3. Experiment Three: Argument mining using Combined Features

The last experiment performed by using the combination of propositional semantic similarity with discourse markers is to take the advantage of both feature sets. As a result, the performance is increased when combined both feature sets together.

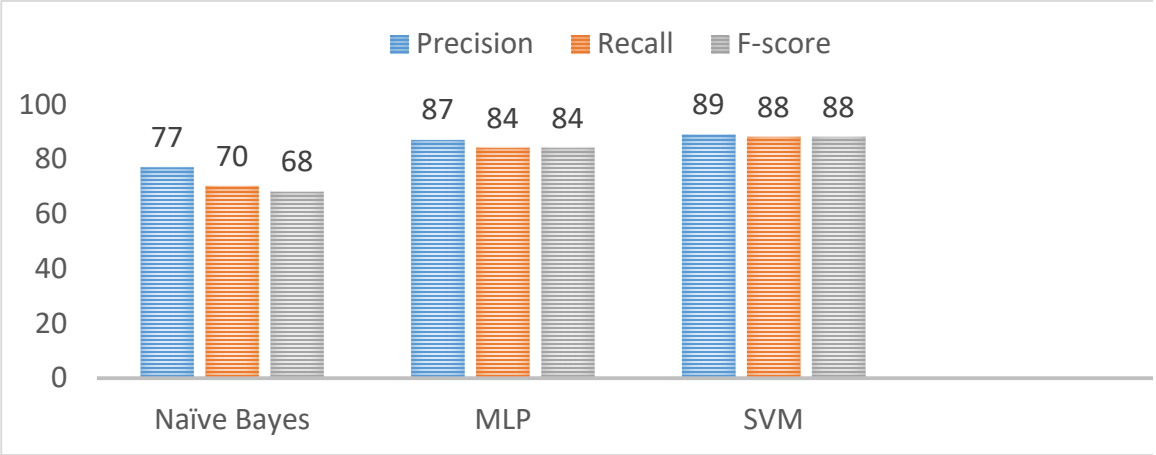


Figure 4: Prediction performances of Naive Bayes, MLP and SVM Classifier using combined features

4.4. Discussion of the Experiment

In this study, argument relation prediction model is designed and developed for Amharic argumentative texts using supervised machine learning algorithms. A propositional semantic similarity and discourse marker features are used to train the model. The model is evaluated using evaluation metrics of precision, recall, and F-score. A total of nine experiments have been conducted on a total of 815 argumentative sentences with three classes (support, attack, and neutral), three basic features (discourse marker, propositional semantic similarity, and combined features), and in separate experimental setups.

The first experiment is conducted using discourse markers as input features for which weighted Averages of F-score 64%, 75%, and 85% are obtained using Naïve Bayes, MLP, and SVM respectively. In the second experiment, the weighted Averages of F-score 79%, 80%, and 87% are obtained using Naïve Bayes, MLP, and SVM respectively. The performance improvement in the second experiment is due to the employment of Word2Vec features for our propositional semantic similarity tasks that capture better semantic relation tarring data sets. Finally, the last experiment is conducted by combining both discourse features and propositional semantic similarity features, and the weighted Averages of F-score of 68%, 84%, and 88% are gained using Naïve Bayes, MLP, and SVM respectively. This result shows there is a significant performance improvement is gained while combining both features as an input set. In general, this study reveals using combining features of both discourse marker and propositional semantic similarity and an SVM classifier is better for Amharic argument relation prediction task.

5. CONCLUSION AND FUTURE WORK

Of many languages, Amharic is one of the most widely used African languages through which several documents, containing argumentative texts are published in both electronic and printed document format. As a result, an abundance amount of comments and feedback is found in the form of fact or opinion that holds arguments to collect and analyze from different genres. Argumentation (Argument mining) is a recent phenomenon in the current NLP application that is used to detect and identify the argumentative structure in texts. This study focused on designing and developing Amharic arguments relation prediction model

which is used to predict or classify the relation that exists between arguments or argument components as support, attack, or neutral. A total of 815 Amharic argumentative text corpus is prepared and of these, propositional semantic similarity, discourse marker features, and a combination of both these features are extracted and used for model training. A total of nine experiments have been conducted to evaluate the model and the best algorithm for Amharic argument relation prediction task is selected. The experiment is classified into three major steps using three learning algorithms (MLP, Naïve Bayes, and SVM) and three different data sets based on input data features. In this experiment, the highest weighted average F-score of 85%, 87%, and 88% are registered using discourse marker features, propositional semantic similarity features, and a combination of these features respectively. In all of the experiments, SVM shows the best classifier among the three machine learning algorithms. Finally, there are many open rooms for improvement in argument relation tasks for the Amharic language. Based on the experimental result achieved and the conclusion remarked, the following recommendation and future works are forwarded by the researchers.

- Building a large and quality argumentative corpus for Amharic argument relation task to increase the performance of the system.
- In this study, we focused only argument relation prediction task for Amharic argumentative texts. Using a different feature performing the full argument mining pipeline can be future work for other researchers.
- In this study, we performed argument relation prediction using supervised machine learning. For future research, we recommend using deep neural network learning algorithms and including additional features to increase the performance of the system.
- In this study, we have to exploit well-known discourse markers manually and from the word2vec model. Developing a standard discourse marker Treebank is a recommended task for future work.

6. MAJOR CONTRIBUTIONS

- In this study, the authors tackle the problem of argument relation prediction (a subtask of argument mining) for the Amharic language. This topic has been studied extensively for English and other European languages, but not for African languages, so the study is an important contribution to extending the field to African languages.
- There are a number of datasets specifically created for relation prediction in argument mining for English, but none for Amharic. Particularly, a total of 815 Amharic argumentative sentences (120 argument maps) is annotated and made publicly available for future research work. The introduction of this new dataset is therefore an important contribution that needs to be emphasized in this paper.

REFERENCES

[1]. Mochales Raquel, and Marie-Francine Moens. "*Argumentation mining." Artificial Intelligence and Law 19,* no. 1 (2011): 1-22.

[2]. Budzynska Katarzyna, and Serena Villata. "*Argument Mining." IEEE Intelligent Informatics Bulletin 17, no. 1 (2016): 1-6.*

[3]. Mayer Tobias, Elena Cabrio, Marco Lippi, Paolo Torroni, and Serena Villata. "*Argument Mining on Clinical Trials."* In COMMA, pp. 137-148. 2018

[4]. Lawrence John, and Chris Reed. "*Mining argumentative structure from natural language text using automatically generated premise-conclusion topic models*." In Proceedings of the 4th Workshop on Argument Mining, pp. 39-48. 2017.

[5]. Lippi Marco, and Paolo Torroni. "*Argumentation mining: State of the art and emerging trends."* ACM Transactions on Internet Technology (TOIT) 16, no. 2 (2016):

[6]. Habernal Ivan, Judith Eckle-Kohler, and Iryna Gurevych. *"Argumentation Mining on the Web from Information Seeking Perspective.*" In ArgNLP. 2014.

[7]. Llewellyn Clare. "*Using Argument Analysis to Define a Structure for User Generated Content.*" In Joint Conference on Digital Libraries. 2012.

[8]. Bentahar Jamal, Bernard Moulin, and Micheline Bélanger. "*A taxonomy of argumentation models used for knowledge representation."* Artificial Intelligence Review 33, no. 3 (2010): 211-259.

[9]. Walton Douglas, Christopher Reed, and Fabrizio Macagno. *Argumentation schemes*. Cambridge University Press, 2008.

[10]. Toulmin Stephen E. *The uses of argument*. Cambridge university press, 2003

[11]. Peldszus Andreas, and Manfred Stede. "*From argument diagrams to argumentation mining in texts: A survey.*" International Journal of Cognitive Informatics and Natural Intelligence (IJCINI) 7, no. 1 (2013): 1-31.

[12]. Lippi Marco, and Paolo Torroni. *"Argument mining: A machine learning perspective."* In International Workshop on Theory and Applications of Formal Argumentation, pp. 163-176. Springer, Cham, 2015

[13]. Cabrio Elena, and Serena Villata. "*Five Years of Argument Mining: a Data-driven Analysis*." In IJCAI, pp. 5427-5433. 2018.

[14]. Green Nancy L. "*Argumentation Mining in Scientific Discourse*." In CMNA@ ICAIL, pp. 7-13. 2017.

[15]. Dey Ayon. "*Machine learning algorithms: a review*." International Journal of Computer Science and Information Technologies 7, no. 3 (2016): 1174-1179.

[16]. Ahuja Yashima, and Sumit Kumar Yadav. "*Multiclass classification and support vector machine.*" Global Journal of Computer Science and Technology Interdisciplinary 12, no. 11 (2012): 14-20.

[17]. Sathya R., and Manama Abraham. "*Comparison of supervised and unsupervised learning algorithms for pattern classification."* International Journal of Advanced Research in Artificial Intelligence 2, no. 2 (2013): 34-38.

[18]. Gobena Markos Kassa. *"implementing an open source Amharic resource grammar in gf*." (2010).

[19]. Pellegrini Thomas, and Lori Lamel. "*Automatic word decompounding for asr in a morphologically rich language: Application to Amharic*." IEEE transactions on audio, speech, and language processing 17, no. 5 (2009): 863-873.

[20]. Fekede Menuta," *Over-differentiation in Amharic orthography and attitude towards reform*". Ethiop.j.soc.lang.stud. 3(1), 3-32. eISSN: 2408-9532; pISSN: 2412-5180. e · July 2016

[21]. Jurafsky Dan, and James H. Martin. "*Speech and language processing".* Vol. 3. (2014).

[22]. Gasser Michael. "*HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya*." In Conference on Human Language Technology for Development, Alexandria, Egypt. 2011.

[23]. Mikolov Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "*Efficient estimation of word representations in vector spac*e." arXiv preprint arXiv:1301.3781 (2013).

Offline Handwritten Amharic Digit and Punctuation Mark Script Recognition using Deep learning

Mahlet Agegnehu[1, 2,]*, Getahun Tigistu[1], and Mesay Samuel[1]

[1]*Arba Minch University, Faculty of Computing and Software Engineering, Arba Minch, Ethiopia.*
[2]*Dilla University, School of Computing and Informatics, Dilla, Ethiopia*
**Corresponding author, e-mail: magegnehu.1221@gmail.com*

ABSTRACT

Amharic is an indigenous Ethiopic script that follows a unique syllabic writing system adopted from an ancient Geez script. The Ethiopic script used by Amharic has about 317 different symbols of which 238 basic characters, 50 labialized, 20 numeric, and 9 punctuation marks. Recently Optical Character Recognition for the Amharic Script has become an area of research interest because there is a bulk of handwritten Amharic documents available in libraries, information centers, museums, and offices. Digitization of these documents enables to harness already available language technologies to local information needs and developments. Converting these documents will have a lot of advantages such as preserving and transferring the history of the country, saving storage space, proper handling of documents, and enhancing retrieval of information through the internet and other applications. Few research works have been made for handwriting character recognition of Amharic scripts but most of them use a dataset that is composed of text characters only, not including digit and punctuation mark scripts. A fully handwriting character dataset for Amharic scripts which include all text, digit, and punctuation marks is not available. As a result, doing complete handwriting character recognition at this level is very challenging and time-consuming. So, this research concerns on Handwriting digit and punctuation mark scripts only. The main objective of this research is to develop a model for recognizing digit and punctuation mark script so that, future researchers can integrate this research with previously done handwriting text character recognition and generate the complete handwriting character recognition for the Amharic language. Using a convolutional neural network and by performing a grid search optimization on the hyper-parameters of the network, the researcher attained an accuracy of 0.8693 for training, 0.7102 for validation, and 0.7004 for the testing dataset..

Keywords: *CNN, HCR, Amharic language scripts, digit, and punctuation mark*

1. INTRODUCTION

Amharic, which belongs to the Semitic language is the official and working language of Ethiopia and the second most often learned language throughout the country next to English[1]. Amharic has a syllabic writing system that is derived from an ancient Geez script, is widely used in Ethiopia's official and non-government sectors to this day. Amharic incorporated all of Geez's symbols as well as some new ones that indicate sounds that aren't available in Geez. Amharic writing system uses about 317 different symbols of which 238 basic characters, 50 labialized, 20 numeric, and 9 punctuation marks which are written and read, as in English, from left to right[2].

A mass of Amharic handwritten documents is available in libraries, information centers, museums, and offices addressing various significant issues including science, religion, social rules, cultures, and artworks which are very reach indigenous knowledge[3]. Nowadays, having information available in digital format is becoming increasingly important for increased efficiency in data storage and retrieval, converting these

documents into electronic format is essential to preserve historical documents, save storage space, and enhance retrieval of relevant information via the Internet. This enables to harness existing information technologies to local information needs and developments.

Optical character recognition (OCR) is a technology that converts images of typed, handwritten, or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene photo, or subtitle text placed on an image[4]. A handwritten character recognition (HCR) is one part of the OCR system that aims at transforming a large number of handwritten documents into machine-encoded text. Technically, this system includes procedures like scanning documents or taking photographs using a scanner or imaging device respectively, pre-processing to clear unwanted pixels, segmenting each character from a scanned or imaged document, extracting features of a character, and finally recognizing characters by training the network using the prepared dataset[5]. OCR contributes immensely to the advancement of the automation process of information storage, processing, and search or retrieval, and improves the interface between man and machine in numerous applications, which include, reading aid for the blind, bank checks, and conversion of any handwritten document into structural text form [5]

For decades, OCR applications have been widely used and implemented to digitize a wide range of historical and modern documents, such as books, newspapers, magazines, and cultural and religious archives written in a variety of scripts. Multiple intensive works for multiple scripts have been done in the area of document image analysis with a better recognition accuracy; most of the scripts now even have commercial off-the-shelf OCR applications. In such a way, many researchers think that the OCR challenge is solved. However, OCR gives better results only for very specific use cases and there are still multiple indigenous scripts, like Amharic, which are underrepresented in the area of natural language processing (NLP) and document image analysis[6]. So, Amharic character recognition is still an area that requires the contribution of many research works.

Several algorithms have been proposed for handwritten character recognition such as support vector machine, hidden Markov model, and neural network. Neural networks are biologically-inspired programming paradigms that enable a computer to learn from observational data. In recent years, neural networks are becoming popular to solve problems of classification with many features[5]. Deep learning, a powerful set of techniques for learning in neural networks, is about learning multiple levels of representation and abstraction that help to make sense of data such as images, sound, and text. Recent studies on convolutional neural networks (CNN) [7] have shown their power in document recognition tasks. Even if few research works have been made for handwriting character recognition of Amharic scripts, due to lack of dataset and a large set of Amharic scripts, most of them are focused on text characters only, not including digit and punctuation marks scripts. So, this work tried to fill this gap by developing a model that can recognize all 20 digit and 9 punctuation mark scripts. The characters are shown in the figure below.

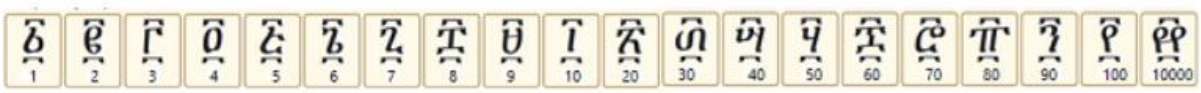


Figure 1: Amharic number representation[8]

Figure 2: Amharic Punctuation marks [9]

2. RELATED WORK

Although a lot of research works have been done for handwritten character recognition for different languages like English and Asian languages such as Japanese, Chinese and Korean, a few works have been reported in the scientific literature related to the recognition of Amharic printed and handwritten documents recognition. In recent literature survey [10] reported that out of selected 172 studies in the domain of handwritten character recognition, the English language has the highest contribution of 53 studies, 44 studies related to the Arabic language, 37 studies are on the Indian scripts, 23 on the Chinese language, 18 on the Urdu language, while 14 studies were conducted on the Persian language and from other approaches, CNN has reported great success in character recognition task and has been widely used for classification and recognition almost for all the languages.

To exploit the application of OCR techniques to Amharic texts, the first research was conducted on printed Amharic text scripts in 1997 [11]. Since then, many studies had been performed and various preprocessing techniques such as segmentation, thinning, underline removal, image restoration, size normalization, feature extraction, and slant correction algorithms had been adopted. Attempts were made not only for machine-printed document recognition but also for typewritten and handwritten Amharic documents.

A work by Million Meshesha and Jawahar [1] proposes a novel feature extraction scheme for OCR of Amharic scripts using principal component and linear discriminant analysis, followed by a decision-directed acyclic graph-based on support vector machine classifier, and the average 96.95% accuracy is obtained. But most attempted printed character recognition of Amharic scripts are dependent on the size and font.

Yaregal Assabie[12] explore possibilities of developing a versatile OCR system that is independent of sizes of Amharic characters by using the most commonly used pattern recognition techniques template matching, statistical, syntactic/structural, and neural network and discover a hybrid system of syntactic/structural and neural network approaches to take their advantage. The syntactic/structural approach enables the developed OCR system to extract primitive structures of characters and generate a unique pattern for each character to be used by the neural network. The neural network enables the developed OCR system to classify/recognize the patterns generated and it can also predict for new cases.

Assabie and Bigun [13] have implemented offline handwritten Amharic character recognition based on the characteristics of primitive strokes that make up characters using Hidden Markov Models. They also develop an Amharic handwritten character dataset[14]. Siranesh getu [5] adopted ANN to recognize ancient handwritten Geez scripts by preparing a dataset that consists of 24 base characters of the Geez alphabet with 100 frequencies. Overall, the recognition accuracy of 93.75 percent was obtained using 3 hidden layers with 300 neurons.

Many studies have been performed on offline handwritten character recognition of different scripts using a Convolutional neural network [10]. In research [12] the best approach to get more than 90% accuracy in the field of HCR using CNN is described.

For the first time, a CNN approach is adopted for Amharic scripts, the paper titled "Handwritten Amharic Character Recognition Using a Convolutional Neural Network"[15] developed a convolutional neural network model to recognize handwritten Amharic characters. The authors used a Dataset from the work of Assabie u. Bigun[14]. They take twelve unique handwritings for every 265 characters and further augmented the data using different data augmentation techniques. They achieve an accuracy of 87.48% for training and 52.15% for validation at 300 epochs.

The paper [13] shows the use of CNN leads to significant improvements across different machine learning classification algorithms. The author develops a CNN model for handwritten Amharic characters by collecting 500 datasets for each character, having (500x265) 132,500 datasets in total. From the collected dataset 20% for validation and 80% for training and they have achieved an accuracy of 91.83% on the training dataset and 90.47% on the validation dataset using CNN. A work of Fitehalew Ashagrie and Boran[16] achieves a successful result, accuracy of 99.39% by using CNN for handwritten Geez script recognition and this shows the power of CNN in handwritten character recognition.

As the researchers observed, all papers except one[8] are focused on text character only, excluding digit and punctuation mark scripts. Eyob Gebretinsae[8] presents an offline handwritten and machine-printed Geez number recognition using feedforward backpropagation artificial neural network. In this paper 560 character images, 28 images for each digit are collected and used 82.2% of the data for training and 17.8% for testing. They achieved 98.03% accuracy for training and 65.3 % accuracy for testing classification performance.

3. METHODOLOGY

3.1. Dataset preparation

The Amharic Handwritten digit and punctuation mark character dataset used in our system is created by collecting various handwritings of different individuals that have different backgrounds. 100 individuals are given a white paper with boxes where each script's handwriting is written and a single piece of paper that has a format to show where to write each character. So that, every individual writes each digit and punctuation mark characters two times in a single white paper with the same format. Then all the papers are collected and scanned using a scanner. After that, each character on each paper is cropped out and saved as a single image file, and placed into a folder that represents its class. In our case, there are 29 different characters of which 20 of them are digit scripts and nine of them are punctuation mark scripts. So, we have 29 different folders named by each class. Finally, each folder contains 200 different images of a single script. In total the dataset has (200x29) 5800 images.

3.2. Image Preprocessing

Preprocessing is used to make input data appropriate for the recognition phase through filtering the noise, binarization, normalization, etc. for noise removal we used different techniques such as image dilation,

image erosion, and morphological opening and closing. The images are changed into binarized form and resized to 32 x 32 pixels

3.3. Architecture of the proposed model

For this work, a CNN is selected to construct a model because many research works [10] [7] [17] show the power of CNN on the handwritten character recognition of different languages and it has recently been used as an efficient feature extractor and classifier.

To get the best fit model of CNN-based architecture a lot of trial and error network configuration has been used. The best architecture is stated as follows. The convolutional layer acts as a feature extractor that extracts salient features of the input. The model has three convolutional layers having 64, 128, and 192 filters of size 3x3 from the first to last convolution layer respectively. The kernels (filters) are used to generate feature maps that are used to identify different features present in an image. At each convolution layer, the RELU activation function is used to add nonlinearity to the output of that layer. After each convolution layer, there are pooling layers. Those are used to precise the recognition by reducing the dimension of the image. There are many ways of pooling but for our system max-pooling with 2x2 pixel size is used because max-pooling keeps maximum information by taking the maximum values from each sub-region, and this leads to faster convergence and better generalization.[18]

After three repetitive convolution and pooling layers, the flatten layer is used to convert the final feature map, which is an output from the last max-pooling layer into one single 1D vector. Finally, there are two fully connected (Dense) layers. The first uses a relu activation function and a dense of 500. The last fully-connected layer use softmax function for classification and has a Dense of 29 that gives the distribution probability of each 29 class. For regularization dropout of 0.5 is used after the last pooling layer and after the first fully connected layer.

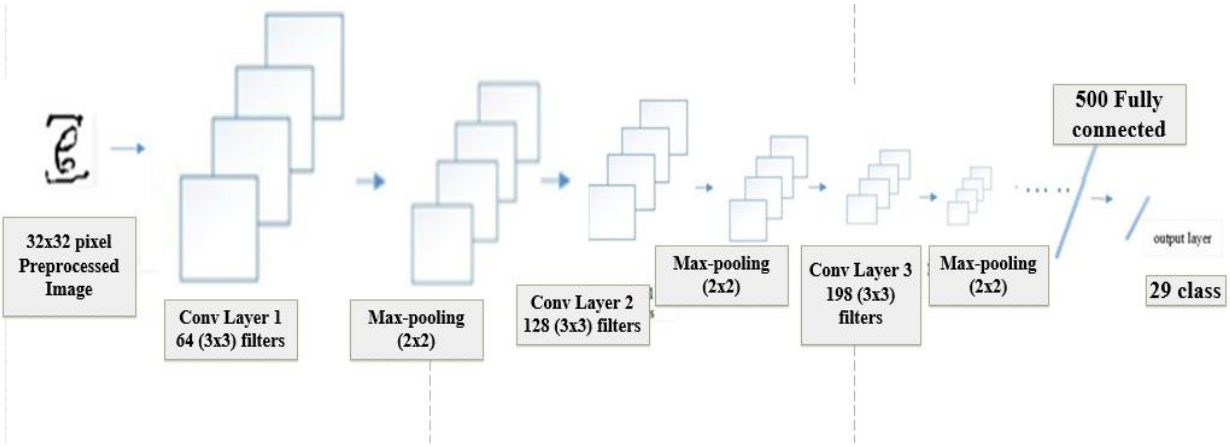


Figure 3: The architecture of the proposed system

4. RESULT AND DISCUSSION

To get the best fit model of CNN architecture a lot of trial and error network configuration has been made. CNN has many parameters and hyper-parameters that can affect the performance of the network. Such parameters are the number of convolution layers, number of filters and size of the filter, number of neurons on the dense layer, and so on. Some of the hyper-parameters are activation function, dropout rate,

optimizer, learning rate, batch size, number of epochs, padding and stride, and so on. To get the best model of CNN architecture those parameters and hyper-parameters should be optimized.

We configure four different models based on the number of the convolutional layer they have. We used constant values of hyper-parameters for all of the trials. Which is, Dropout 0.5. Optimizer adam (learning rate 0.001), filter size (3,3), number of filter=64, number of neuron on the dense=150, activation function = relu, batch size = 20 and epoch 100.

To prevent shrinking of the feature map as well as the loss of information present on the corner we used padding = 'same' and stride of (1, 1). And the result is shown in the Table below.

Models	Training loss	Training accuracy	Validation loss	Validation accuracy
Two Convolution Layer	0.1023	0.9631	3.2062	0.5872
Three Convolution Layer	0.1353	0.9500	2.3462	0.6197
Four Convolution Layer	0.0859	0.9717	3.1979	0.6107
Five Convolution Layer	0.0976	0.9696	4.0165	0.5717

As we can see from the table, a CNN model with three convolution layers has a better performance than the other. A CNN model with four convolution layers is also good but the validation loss is quite big. So, a three convolution layers network configuration is selected.

After that, the number of neurons and the size of neurons that should be used on each layer are taken into consideration.

The number of filters used in many studies is a multiple of 64 [3] [18] [2]. In some studies, using 64 filters for the first and increasing the number of filters by the number multiple of 64 for the next layers gives a better result [2] [18]. The paper entitled 'Handwritten Amharic Character Recognition System Using Convolutional Neural Network' used 64 filters for the first few and 32 filters for the rest and get a good result [3]. So, we perform a confusion matrix for all number of filters and size of filter pair scenarios and we find out that, having 64, 128, and 192 filters of size 3x3 from the first to last convolution layer respectively gives a better result. We also perform grid search optimization for other hyper-parameters and we find out that using 500 neurons gives a better result than others.

From a total of (29*200) 5800 data, by using train – test –split method, we take 20 % for testing, from the 80% of the training, 20 % of data are taken for validation. Finally, we used 3,712 for training, 1,160 for testing, and 928 for validation. All implementations are performed on google colab environment with jupyter notebook editor. By considering the above trials, the best parameters and hyper-parameters are taken and we got loss: 0.1325 - accuracy: 0.9557 for training, loss: 1.8229 - accuracy: 0.6490 for validation, and loss: 2.3077 - accuracy: 0.6256 for testing.

As we can see from the result (Figure 4) the validation accuracy is very small compared with the training accuracy and the validation loss becomes increased after epoch 20, from 1.250 it gets 2.3077 at epoch 100. This shows that there is overfitting. Overfitting means when a model performs well for training data but is unable to generalize for new data. This is mainly caused by having a minimum set of training data that CNN needs numerous data to perform well. The overfitting problem can be solved by regularizing the model,

early stopping before the loss becomes increased, or by increasing the training data. To increase the training data, we used one of the mostly used technique called data augmentation. Data augmentation is a strategy used to increase the amount of data by using techniques like cropping, padding, and flipping on real-time data. By combining the regularizing technique and the data generator, the overfitting problem is solved and a respectable result (Figure 5) was attained. Which is, loss: 0.3877 - accuracy: 0.8693 for training, loss: 1.1561 - accuracy: 0.7102 for validation and loss: 1.2455 - accuracy - 0.7004 for testing.

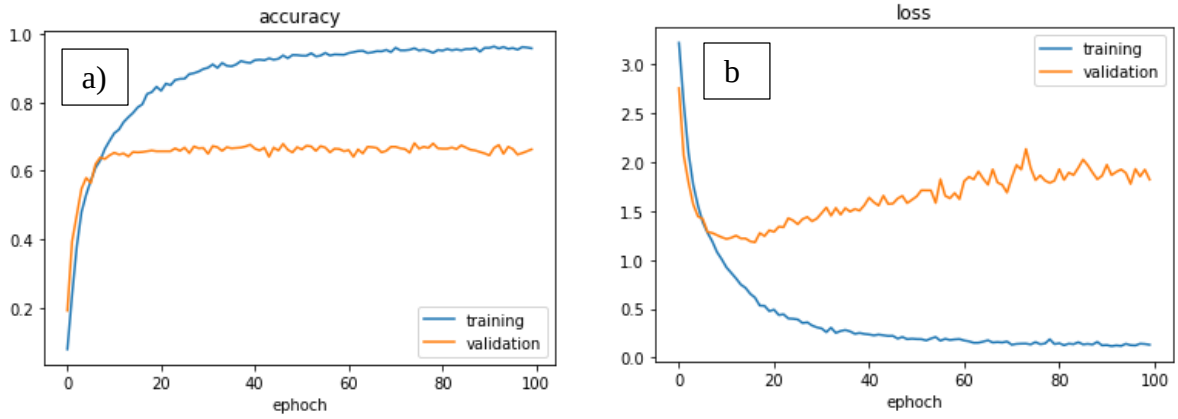


Figure 4: (a) Validation vs training accuracy and (b) validation vs training loss

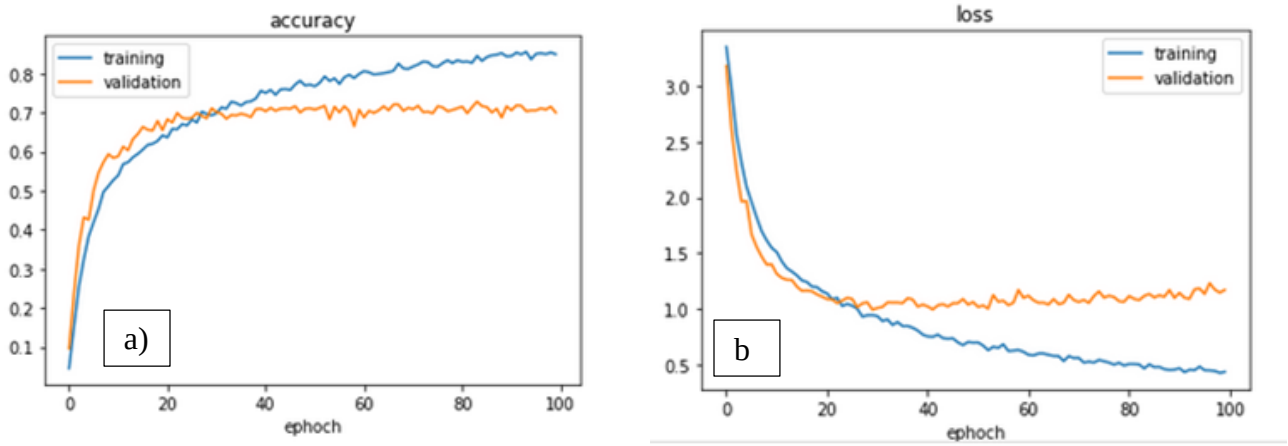


Figure 5: (a) Validation vs training accuracy (b) validation vs training loss

5. CONCLUSION AND RECOMMENDATION

In this study, Amharic handwritten digit and punctuation mark script recognition was addressed using a convolutional neural network. Because without the need for handcrafted feature extraction it was observed that one can achieve a reasonable recognition result using CNN. Even if digit and punctuation marks play a great role in the meaning of any textual documents, the previous research works mainly focused on the 265 characters, they excluded the Amharic numerals and punctuation marks from the studies. The work tried to fill this gap by developing a CNN model to recognize Amharic handwritten digit and punctuation mark script and we get a reasonable result. Due to the lack of research works on the area, there is a big

challenge to get a dataset for Amharic language scripts, especially for digit and punctuation mark scripts. In this research, we develop a dataset that can be used by other researchers in the future.

Even if the result is satisfactory, we recommend that different works should be done to improve the accuracy of the CNN model because many studies show that CNN is very powerful and can give even greater than 99% accuracy in this area. The deep neural network needs a lot of data for better performance. The more the size of the data, the higher the performance will be. So, by adding more data, a researcher can improve the accuracy of this model. In addition to that, it is very important to incorporate this work with the previous work of handwritten Amharic character recognition to develop an inclusive OCR system for the Amharic language.

REFERENCE

[1] M. Meshesha and C. V. Jawahar, "Optical Character Recognition of Amharic Documents," *Afr. J. Inf. Commun. Technol.*, vol. 3, no. 2, Aug. 2007, doi: 10.5130/ajict.v3i2.543.

[2] B. Belay, T. Habtegebrial, M. Meshesha, M. Liwicki, G. Belay, and D. Stricker, "Amharic OCR: An End-to-End Learning," *Appl. Sci.*, vol. 10, no. 3, p. 1117, Feb. 2020, doi: 10.3390/app10031117.

[3] F. Abdurahman, "Handwritten Amharic Character Recognition System Using Convolutional Neural Networks," 2019, doi: 10.12739/NWSA.2019.14.2.1A0433.

[4] "Optical character recognition," *Wikipedia*. Mar. 06, 2021. Accessed: Mar. 29, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Optical_character_recognition&oldid=1010662463

[5] S. Getu, "Ancient Ethiopic Manuscript Recognition Using Deep Learning Artificial Neural Network." Addis Ababa University, 2016.

[6] B. H. Belay, T. A. Habtegebrial, and D. Stricker, "Amharic character image recognition," in *2018 IEEE 18th International Conference on Communication Technology (ICCT)*, 2018, pp. 1179–1182.

[7] D. S. Maitra, U. Bhattacharya, and S. K. Parui, "CNN based common approach to handwritten character recognition of multiple scripts," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, Aug. 2015, pp. 1021–1025. doi: 10.1109/ICDAR.2015.7333916.

[8] E. G. Beyene, "Handwritten and Machine printed OCR for Geez Numbers Using Artificial Neural Network," *ArXiv191106845 Cs Eess*, Nov. 2019, Accessed: Mar. 31, 2021. [Online]. Available: http://arxiv.org/abs/1911.06845

[9] "Amharic alphabet, pronunciation and language." https://omniglot.com/writing/amharic.htm (accessed Mar. 30, 2021).

[10] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," *IEEE Access*, vol. 8, pp. 142642–142668, 2020, doi: 10.1109/ACCESS.2020.3012542.

[11] W. Alemu, "The application of ocr techniques to the amharic script," *MSc Thesis Addis Ababa Univ. Fac. Inform.*, 1997.

[12] A. Yaregal, "Optical character recognition of Amharic text: an integrated approach," *Sch. Inf. Stud. Afr. Addis Ababa Univ. Addis Ababa*, 2002.

[13] Y. Assabie and J. Bigun, "Offline handwritten Amharic word recognition," *Pattern Recognit. Lett.*, vol. 32, no. 8, pp. 1089–1099, 2011.

[14] Y. Assabie and J. Bigun, "A comprehensive Dataset for Ethiopic Handwriting Recognition," Halmstad University, 2009, pp. 41–43. Accessed: Mar. 30, 2021. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-25833

[15] M. S. Gondere, L. Schmidt-Thieme, A. S. Boltena, and H. S. Jomaa, "Handwritten Amharic Character Recognition Using a Convolutional Neural Network," *ArXiv190912943 Cs Stat*, Sep. 2019, Accessed: Mar. 27, 2021. [Online]. Available: http://arxiv.org/abs/1909.12943

[16] F. A. Demilew and B. Sekeroglu, "Ancient Geez script recognition using deep learning," *SN Appl. Sci.*, vol. 1, no. 11, p. 1315, Nov. 2019, doi: 10.1007/s42452-019-1340-4.

[17] A. Baldominos, Y. Saez, and P. Isasi, "A survey of handwritten character recognition with mnist and emnist," *Appl. Sci.*, vol. 9, no. 15, p. 3169, 2019.

[18] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, "Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)," *Sensors*, vol. 20, no. 12, Art. no. 12, Jan. 2020, doi: 10.3390/s20123344.

Sentence Level Automatic Speech Segmentation for Amharic

Rahel Mekonen Tamiru[1,*], Solomon Teferra Abate[2]

[1]*Bahir Dar University, Bahir Dar, Ethiopia*
[2]*Addis Ababa University, Addis Ababa, Ethiopia*
**Corresponding author, e-mail: rahelmekonen9@gmail.com*

ABSTRACT

Many speech processing systems require segmentation of Speech waveform into principal acoustic units. The extraction of information from a large archive requires extracting both audio file structure and its linguistic content. One of these processes is to add sentence boundaries to the automatic transcription of speech contents. The main objective of this segmentation process is to use the result for other area of speech processing. It is an essential preprocessing step in several speech research areas. Missing sentence segmentation makes meaning of some utterances ambiguous and cause significant problems to automatic downstream processes. In this work, we present an automatic sentence-level speech segmentation system for Amharic language. We have used Amharic read speech, and a spontaneous speech corpus for the development of automatic speech segmentation system. In this work, automatic speech segmentation system is completed by forced alignment. Monosyllable, tied-State tri-syllable, and mono phone acoustic models have been developed to build forced alignment. Rule-based and AdaBoost have been used to differentiate the accurate boundaries from candidate. We have used decision tree classifier and support vector classifier (SVC) as a base estimator. The evaluation of the experiments shows that encouraging automatic speech segmentation results are achieved using monosyllable acoustic model forced alignment. Our proposed AdaBoost classifier achieved the best results using a decision tree classifier as a base estimator with a segmentation accuracy of 91.93% and 85% result for read loud and spontaneous speech respectively. Based on the findings of our experiment, for segmenting and labeling of Amharic speech data at sentence level, monosyllable acoustic model is the better model to get accurate forced alignment with regard to its sentences segmentation accuracy and also pause feature is an important indicator of sentence boundaries.

Keywords: *Sentence Segmentation, Acoustic Model, Decision Tree Classifier, Support Vector Machine, AdaBoost, Forced Alignment*

1. INTRODUCTION

Current research in the field of speech technology aims to develop efficient speech systems that can be used for communication between peoples and devices for processing of information. Unfortunately, the ability of a computer to understand speech is still weak. Since human speech is continuously generated, the most difficult aspect of speech that challenges machines is its segmentation. Several speech processing systems require speech segmentation wave form into principal acoustic units (phonemes, syllables, sentence, and paragraph). In the field of speech technology, it is very primary phase. The primary purpose of this segmentation process is to use the outcome for other areas of speech research. In several speech research areas such as speech recognition, speech synthesis, language generation based system, and language identification and speaker identification system, speech segmentation is an important preprocessing phase (Ostendorf et al., 2008;). Therefore, to achieve this objective, well-organized, precise, and simple technique is required. While tending to the segmentation task, it must be considered that speech is not clearly organized as written text, especially spontaneous speech.

For both humans and machines, it is hard to identify sentence unit from continuous speech. Jones found that the legibility of speech transcripts is important for sentence breaks (Jones et al., 2003). In addition, missing segmentation of sentences makes certain utterances vague in importance. Similarly, kolar found that missing sentence boundaries cause automatic downstream processes to have major problems (Kolář, 2008). When evaluating the syntactic complexity of speech, sentence boundaries are significant, which can be a strong indicator of disability (Fraser et al., 2015). Manual segmentation of speech is costly, boring, prone to mistakes, and time-consuming if it has to be performed by humans(Emiru and Markos, 2016). Given these and other facts, it is becoming increasingly important to improve automatic speech segmentation.

There are similar studies performed in different languages on speech segmentation. However, there has been no previous work done for Amharic on sentence level speech segmentation. While there are similarities between different languages in the structure, and function of prosody, there are major differences suggested by cross-linguistic comparison of characteristics (Vaissière, 2012; Mekonen, 2019). We do not know which features or their combinations for Amharic speech will result in optimal segmentation of speech. We present our work in this paper to create an acoustic model and use this model to facilitate the segmentation process and to get segmented and labeled Amharic speech data.

We have used two classification algorithms to detect sentence boundary. The first is rule based and the second is statistical method, AdaBoost. In order to construct a strong classifier, a popular boosting approach known as AdaBoost combines weak-based classifiers. To generate an effective classifier, the concept of boosting is to combine several poor learning algorithms. There are several base classifiers are used by AdaBoost. It uses the classifier of the decision tree as the default classifier (Deng, 2007). We used decision tree classifier and support vector classifier (SVC) as a base estimator in our experiment.

In the following section, we explain some of the previous works that have developed automatic speech segmentation system. We'll provide a brief overview of the Amharic language considered in this study in section 3. In section 4, Amharic corpora used for our studies will be present. The results of automated speech segmentation experiments performed are summarized in section 5. Finally, there are closing remarks and potential future studies to strengthen the work done by this study.

2. RELATED WORKS

While automatic speech segmentation for other languages has been developed by different researchers, most of the studies were performed using only prosodic features directly from the speech to define sentence boundaries. There is a research report on spontaneous Malay language speech segmentation(Jamil et al., 2015). For the detection of sentence limits, acoustic and prosodic features were used. Other works have been carried out, such as an Automatic segmentation of speech into sentences-like units (Kolář, 2008), voice segmentation without voice recognition (Mulgi et al., 2013), prosody-based automatic segmentation of speech into sentences and topics (Shriberg, 2000). There is no sentence level speech segmentation for Ethiopian languages that have done well in other languages to the best of our knowledge. While there are similarities between various language such as pause, natural tendency for F0, in type and function of prosody, there are major differences suggested by cross-linguistic comparison of characteristics such as

different timing of essentially comparable phenomena, different relationships between F0 or different mutual effects of Fo, duration and intensity (Vaissière, 2012). The above research papers did not use the form of forced alignment for sentence segmentation.

The purpose of this research is therefore to select the best unit for acoustic models to design a speech segmenter with minimal error that can help in the creation of an efficient system of speech segmentation.

3. AMHARIC LANGUAGE

Amharic is the government of Ethiopia official working language, out of the 89 languages registered in the country. Next to Arabic, Amharic is the second largest Semitic language spoken in the world (CSA, 2007). Thus, with at least 27 million native speakers, it is one of the most commonly spoken Semitic languages in Ethiopia. On the other hand, very limited research on acoustic characteristics and spoken language technology, a lack of electronic tools for speech and language processing such as a transcribed speech data, monolingual corpora and pronunciation dictionaries (Abate and Menzel, 2007) has also been described as an under-resourced language. The Addis Ababa, Gojjam, Wollo, Gondar and Shewa dialects are five dialects of Amharic. It is taken from the place where they are spoken. As the standard dialect, Addis Ababa's speech has arisen and has wide currency in all Amharic-speaking communities. The Amharic language's basic word order is SOV. It is one of the languages which have its own method of writing. It is written using the term fidel. Amharic, a semi-syllabic system, has its own writing system. It has about 33 primary characters, each representing a consonant and seven vowels, resulting in 196, different pronunciations in 231 CV syllables. Across all Amharic dialects, Fidel is used (Girmay, 2008).

3.1. The Amharic consonants and Vowels

The Amharic language consists predominantly of 38 phonemes, 7 vowels and 31 consonants. An additional consonant /v/ is inherited and contains a total of 39 phonemes (Amare, 2018). In general, consonants are categorized as stops, fricatives, nasals, fluids and semi-vowels (Abate and Menzel, 2007). During speech formation, vowels have various categories depending on the location and height of the tongue and their shapes. Based on the location of the tongue in oral cavity, vowels are divided into tree front, central and back. These vowels are often graded into high, middle and low on the basis of height of the tongue. Vowels are categorized into two sub-classes that are rounded and unrounded based on their shapes during speech development(Abate and Menzel, 2007; Jokisch, Birhanu and Hoffmann, 2012). Amharic has a total of 7 vowels, five of the most common vowels, a, e, I o, and u, plus two additional central vowels, E and I.

4. CORPUS PREPARATION

Speech and text corpora are one of the most essential tools for any speech segmentation system and development. One of the most complicated and costly activities when dealing with under resourced languages is to collect structured and annotated corpora (Lewis and Yang, 2012). Amharic speeches and related text data were gathered from the Amharic Bible, broadcast news, broadcast conversation and Amharic fictions in order to obtain optimal speech corpus. The collected corpus contains over 5 hours of audio, along with its text corpus, preserved in 40 audio files. Problems such as spelling and grammar errors

are corrected in the document corpus, abbreviations are extended and numbers are transcribed textually. We also produced two corpora that lead to two different types of speech: a read speech corpus and a spontaneous speech corpus. By gathering existing broadcast news corpus and Amharic bibles, the first corpus was created, while the second corpus was created by mixing broadcast conversation and Amharic fictions (fikr skemekaber). 4000 Amharic speech sentences and the corresponding text corpus are gathered for training in this work. Manual segmentation should be done both forthe training and test data. We have, therefore, segmented the collected speech at sentence levelmanually. This segmentation is carried out using Audacity software. Each of the segmented sentence files are sampled at 44.10 kHz with 16-bit resolution, saved in the *.wav format.

5. SPEECH SEGMENTATION SYSTEM

Data preparation, HMM model building, forced alignment and sentence boundary detection are included in the general model of the automatic sentence speech segmentation framework.

5.1. Data preparation

The steps for getting data ready for speech segmentation system include data collection, manual segmentation, lexicon preparation, pronunciation dictionary preparation and feature extraction.

Manual segmentation

The next critical step in data processing, with the speech and corresponding text corpus, is manual segmentation. The development of a sentence level speech segmentation system for Amharic was one of the main objectives of this work. Since no speech corpora with sentence break annotation existed, such corpora had to be prepared as the very first part of our work. Text and speech corpora are divided into sets of training and test. Both for the training and testing results, manual segmentation should be performed. We have, therefore, manually segmented the collected speech at sentence level. For training, 2000 spontaneous speech sentences and 2000 read speech sentences are segmented, while for research, 400 speech sentences from both styles of speech are segmented. In order to be used in the HTK environments, we have transliterated the text transcription of both the training and testing sets into their corresponding ASCII representation.

Lexicon preparation

For our speech automated segmentation system, lexicons are prepared as pronunciation dictionaries. Lexicons are prepared with letter sequences, based on our system. For instance, let's consider Amharic word "ፈጠረ". There are three orthographies of the Amharic word "ፈጠረ" which are ፈ, ጠ and ረ. The corresponding transcription of ASCII becomes "faTara".

Pronunciation dictionary: By taking lexicons as data, we have prepared pronunciation dictionaries.

Feature extraction

Parameterizing the raw speech of the waveforms into sequences of feature vectors are the final stage of data preparation. The method of converting the speech waveform into a collection of feature vectors is feature extraction. To parameterize the speech signals into feature vectors with 39 MFCC coefficients, we use Mel Frequency Cepstral Coefficients.

5.2. HMM model building

Using Amharic speech and its text scripts, we have developed the acoustic model and compiled it into statistical representation of sounds that make up words. All the acoustic models were constructed using the HTK toolkit in the same manner (Young *et al.*, 2009). The basic units of speech used in our research are syllable and phone. We are using HMM in this research to model the acoustic component. Three modeling methods i.e. syllable / phone based acoustic model and tied state acoustic model, are used to perform the acoustic modeling system. We first developed an independent acoustic (monophone) model. To represent each phone, the acoustic model uses a 3-state left-to-right HMM without a skip. Then we initialized the model and created the monosyllable model with flat start techniques. In the next step, the tri-syllable model was derived by cloning, and then re-estimating the respective monosyllable models using tri-syllable transcription.

5.3. Forced Alignment

To line up the written words with the spoken words, we have used monosyllable, tied-State tri-syllable and monophone acoustic models. Various acoustic models can produce slightly different outcomes of forced alignment.

5.4. Speech segmenter

Automated segmented results of the test data set are given by the speech segmenter. A simple classifications method based on the rules is used in our initial sentence segmentation experiment. The segmenter conducts the actual segmentation into sentence level files of the continuous speech and generates a corresponding transcription file. In our second experiment features from forced alignment is extracted. Then AdaBoost, is used to distinguish between false and true boundaries. In our experiment we have used decision tree classifier and support vector classifier (SVC) as a base estimator.

5.5. Experimental Results

In our initial experiment, the Speech segmenter takes two main inputs: forced alignment and an audio file. Then, from forced alignment pause features are extracted. Pause duration is calculated and is used to evaluate the sentence boundary as a threshold. In the first experiment, we used rule-based for sentence segmentation. If the characteristics of a boundary candidate are assessed as valid, the boundary candidate indicates a sentence boundary. In the meantime, if the function of a boundary candidate is measured to a Wrong, the boundary candidate is not a sentence boundary. To detect sentence boundaries, we tried different threshold values (10000, 500, 800, 1000 millisecond). In this experiment we decide minimum pause for sentence break is 1000 millisecond (1 sec). In the second experiment pause features from forced alignment is extracted. A statistical method, AdaBoost, is then applied to all candidates for sentence boundaries. With two corpora Amharic read-aloud speech and spontaneous speech, we test our methods. All experiments were evaluated using human-generated reference transcripts.

Regarding Sentence Segmentation Accuracy (SSA), the experimental output is presented. The comparison of the automated sentence segmentation experiments carried out with respect to the monosyllable, tied-State syllable and monophone as an acoustic model toward read-aloud and spontaneous speech based on the rule-based approach is shown in Table 1 below.

Table 1: Segmentation accuracy of read-aloud and spontaneous speech based on rule-based method

	Read-aloud speech	Spontaneous speech
	SSA	SSA
Monosyllable acoustic model	69.1%	53%
Tied-State syllable acoustic model	61%	47%
Monophone acoustic model	56%	45%

The comparison of the automated sentence segmentation experiments performed with respect to the monosyllable, tied-State syllable and monophone as an acoustic model against the statistical approach, adaBoost, based on read-aloud and spontaneous speech is shown in table two below.

Table 2: Segmentation accuracy of read-aloud and spontaneous speech based on statistical method, adaBoost.

		read-aloud speech	Spontaneous speech
		SSA	SSA
monosyllable acoustic model	Decision tree classifier	91.93%	85%
	SVM classifier	84.3%	79%
Tied-State acoustic model	Decision tree classifier	88.93%	82.7%
	SVM classifier	80.08%	70%
monophone acoustic model	Decision tree classifier	82.2%	80.6%
	SVM classifier	80.2%	76%

As it could be seen from table 1 and 2, we have achieved (69.1% and 53% accuracy) for read aloud and spontaneous speech respectively using rule based method and (91.93% and 85% accuracy) result using decision tree classifier for read aloud and spontaneous speech respectively on monosyllable acoustic model.

The percentage of accuracy monophone acoustic model is low in both experiments. These indicate that various acoustic models produce slightly different outcomes of forced alignment. The better acoustic model gives the more accurate the forced alignments. Therefore, the researcher believes that the best acoustic model for Amharic speech to achieve correct forced alignment is a monosyllable acoustic model. The overall system efficiency is affected by forced alignment. The assessment of the experiments indicates that the adaboost classifier achieves greater accuracy than rule based method. As shown in the experiment, the adaboost classifier consistently showed good results, especially in the classifier of decision tree. The resulting processing time per decision tree classifier is faster than the classifier of the support vector. The majority of our experiments showed the best results. The researcher therefore believes that decision tree classifier is the best method of classification for the segmentation of Amharic speech than support vector classifier (SVC).

6. CONCLUSION AND FURTHER WORK

We presented automatic sentence level speech segmentation that we performed for the Amharic language in this work. In this paper, the automated speech segmentation systems introduced are the first for the

Amharic language. In general, it is accomplished by defining the boundaries in a continuous speech signal between sentences. These findings are promising and will open a broad door for further studies. We are also working in this area by using neural network-based approach to achieve greater accuracy for these languages. To study strong speech systems, it is also important to continue further study in speech preprocessing. A study on automatic speech segmentation on prosodic characteristics (perceptual and linguistic level) and research on automatic speech segmentation on other distinct units (syllable, phoneme, and word level) are also required for future work.

REFERENCES

1. Abate ST and Menzel W (2007) Syllable-based speech recognition for Amharic, vol 33. https://doi.org/10.3115/1654576.1654583
2. Amare G (2018) Yamariñña Säwasäw. J Ethiop Stud 28(2):55–60. http://www.jstor.org/stable/41966049
3. CSA (2007) Central Statistics Agency CSAGoogle Scholar
4. Deng H (2007) A brief introduction to adaboost, pp 1–35Google Scholar
5. Emiru ED and Markos D (2016) Automatic speech segmentation for amharic phonemes using hidden Markov model toolkit (HTK), 4(4):1–7Google Scholar
6. Fraser KC et al (2015) Sentence segmentation of aphasic speech HLT-NAACL 2015–human language technology conference of the North American chapter of the association of computational linguistics proceedings of the main conference, pp 862–871. https://doi.org/10.3115/v1/N15-1087
7. Girmay G (2008) Prosodic modeling for AmharicGoogle Scholar
8. Jamil N et al (2015) Prosody-based sentence boundary detection of spontaneous speech. In: Proceedings - international conference on intelligent systems modelling and simulation ISMS, 2015-Septe(July 2017):311–317. https://doi.org/10.1109/ISMS.2014.59
9. Jokisch O, Birhanu Y and Hoffmann R (2012) Syllable-based prosodic analysis of Amharic read speech, pp 258–262Google Scholar
10. Jones D et al (2003) Measuring the readability of automatic speech-to-text massachusetts institute of technology information systems technology group 2. System 1585–1588Google Scholar
11. Kolář J (2008) Automatic segmentation of speech into sentence-liked unitsGoogle Scholar
12. Lewis WD and Yang P (2012) Building MT for a severely under-resourced language: White Hmong. In: AMTA 2012—proceedings of the 10th conference of the association for machine translation in the AmericasGoogle Scholar
13. Mekonen R (2019) Prosody based authomatic speech segmentation for AmharicGoogle Scholar
14. Mulgi M, Mantri V, Gayatri M (2013) Voice segmentation without voice recognition, 2(1), 2–6Google Scholar
15. Ostendorf M et al. (2008) Speech segmentation and its impact on spoken language technology. IEEE Signal Process Magazine 1–20. https://doi.org/10.1117/12.877599
16. Vaissière J (2012) Language-independent prosodic features, 53–65Google Scholar
17. Young S et al (2009) The HTK bookGoogle Scholar

Amharic-English Machine Translation

Andargachew Mekonnen Gezmu[1,*], Andreas Nürnberger[1], Tesfaye Bayu Bati[2]

[1]*Faculty of Computer Science, Otto von Guericke Universität Magdeburg, Germany*
[2]*Faculty of Informatics, Hawassa University, Ethiopia*
**Corresponding author, e-mail: andargachew.gezmu@astu.ovgu.de*

ABSTRACT

This paper describes the use of deep learning for machine translation of Amharic-English with complex morphology in low-resource conditions. Amharic has a rich morphology and uses the Ethiopic script. To tackle the complex morphology and to make an open vocabulary translation, we used subwords. Furthermore, based on the best practices of prior research in this line of work, we conducted machine translation in low-data conditions. In the automatic evaluation of word-based and subword-based neural machine translation models trained on a benchmark dataset, all subword-based models outperform word-based ones. All neural machine translation models also outperform statistical machine translation models.

1. INTRODUCTION

Amharic is a Semitic language that serves as the official language of Ethiopia. Though it plays several roles in the government, it is considered a low-resource language because of its lack of basic tools and resources for natural language processing (Gezmu et al., 2018; Tracey and Strassel, 2020).

Amharic uses a syllabic writing system, Ethiopic. Each Amharic letter systematically conflates a consonant and vowel (e.g., በ /bə/ and ቡ /bu/). Sometimes consonants and vowels can be written as bare consonants (e.g., ቭ /b/) or bare vowels (e.g., አ /a/ in አገር /agər/). Some phonemes that have one or more homophonic script representations and peculiar labiovelars, sometimes compromise the consistency of the writing system. In Amharic orthography, there is no case difference; it is written from left to right. In present-day Amharic writings, words are delimited by plain space.

Like other Semitic languages, Amharic words are highly inflectional and have a root-pattern morphology.

In this research, we used a public bench mark dataset[3] to train and evaluate neural machine translation (NMT) and phrase-based statistical machine translation (PBSMT) models. To go around the highly inflectional morphology and to make an open vocabulary translation, we used subwords. To segment words into subwords, we used Byte Pair Encoding (BPE) (Sennrich et al., 2016). Furthermore, based on the best practices of prior research in this line of work, we conducted machine translation in low-data conditions. Thus, we used the transformer-based neural machine translation (Vaswani et al., 2017) and phrase-based statistical machine translation models (Koehn et al., 2003).

2. RELATED WORK

So far, Teshome et al. (2015) performed a phoneme-based SMT by converting Amharic grapheme to phoneme. However, they used a small corpus of approximately 18 thousand sentence pairs for training the model. Still, their dataset is not available for the research community.

3 Available at: http://dx.doi.org/10.24352/ub.ovgu-2018-145

3. NMT SYSTEM ARCHITECTURE

To train NMT models, we used the encoder-decoder architecture implemented with Transformers. The system uses the Adam optimizer, a dropout rate of 0.1, label smoothing of value 0.1., training batch size of 1024, eight attention heads, six Transformer blocks, filter size of 2048, and hidden size of 512.

Training steps were 250000 steps. For decoding, we used beam search with beam size of 4 and length penalty of 0.6.

4. EXPERIMENTS AND EVALUATION

We used the public bench mark dataset to train and evaluate NMT and PBSMT models. The training, development, and test sets have 140 thousand, 2864, and 2500 sentence pairs, respectively.

There are many improvements over the baseline NMT and PBSMT during recent years. Nevertheless, to make objective evaluation, we relied on baseline systems for both approaches.

4.1. Data Preprocessing

To share named-entities between the languages, the Amharic datasets were transliterated with a transliteration scheme, Amharic transliteration for machine translation[4] . They were tokenized with an in-house tokenizer. The English data sets were tokenized with Moses' script (Koehn et al., 2007).

4.2. Evaluation

Translation outputs of the test sets were detokenized and evaluated with the BLEU metric (Papineni et al., 2002). To address the limitations of BLEU, we also used BEER (Stanojević and Sima'an, 2014) and CharacTER (Wang et al., 2016) metrics.

5. RESULTS

Table 1 shows example translations of Amharic sentences into English using NMT and PBSMT models.

Table 1: Example translation outputs.

Source	በእርግጥ ለዘላለም መኖር እንችላለን?
Transliteration	bəɨrgṭ ləzəlaləm mənor ɨnclalən?
Reference	can we really live forever?
NMT-Word-Based	can we really live forever?
NMT-Subword-Based	can we really live forever?
PBSMT	really, we can live forever?
Source	በዚያው ጊዜ አካባቢ ወላጆቼ ወደ ቤት እንድመለስ ጠየቁኝ።
Transliteration	bəziyaw gize akababi wəlajoce wədə bet ɨndmələs ṭəyəquñ.
Reference	about that time, my parents asked me to come back home.
NMT-Word-Based	about that time, my parents asked me to return home.
NMT-Subword-Based	at that time, my parents asked me to return home.
PBSMT	, 1929. about that time, my parents and asked me to the house.
Source	አስጨናቂ ሁኔታዎች የስሜት ቀውስ ሊያስከትሉብን ይችላሉ።

4 The implementation is available at: https://github.com/andmek/AT4MT

Transliteration	asċənaqi hunetawoc yəsmet qəws liyaskətlubn yclalu.
Reference	distressing circumstances can have a terrible impact on us.
NMT-Word-Based	painful situations can cause anxiety.
NMT-Subword-Based	distressing events can affect us.
PBSMT	when distressing situations liyaskətlubn emotional pain.
Source	የወይራ ዘይት በከፍተኛ መጠን ስለሚመረት በብዛት ጥቅም ላይ ይውላል።
Transliteration	yəwəyra zəyt bəkəftəña məṭən sləmimərət bəbzat ṭqm lay ywlal.
Reference	olive oil is used copiously, as it is produced there on a large scale.
NMT-Word-Based	olive oil is used in high demand for supplies.
NMT-Subword-Based	olive oil is largely used because it is abundant.
PBSMT	do not harm the olive oil, in the great sləmimərət.
Source	ከስድስት አመታት በኋላ የመላው አለም ኢኮኖሚ ተንኮታኮተ።
Transliteration	kəsdst amətat bəhwala yəmələw aləm ikonomi tənkotakotə.
Reference	six years later, the whole world economy collapsed.
NMT-Word-Based	six years later the entire world is destroyed.
NMT-Subword-Based	six years later, the entire world crisis.
PBSMT	six years later, the economy of the entire world, have been shattered.

In the examples, both models translate short sentences with similar accuracy and fluency. But for long sentences, NMT-Subword-Based models produce more accurate and fluent translations.

Objective evaluations are shown on table 2. It shows performance results of the machine translation models from Amharic-to-English and English-to-Amharic translation with BLEU, BEER, and CharacTER metrics. Unlike BLEU and BEER, the smaller the CharacTER score, the better.

Table 2 Performance results of NMT and PBSMT models.

Translation Direction	MT Model	BLEU	BEER	CharacTER
English-to-Amharic	NMT-Word-Based	23.0	0.510	0.510
	NMT-Subword-Based	26.6	0.560	0.515
	PBSMT	20.2	0.502	0.646
Amharic-to-English	NMT-Word-Based	29.1	0.537	0.592
	NMT-Subword-Based	33.3	0.578	0.522
	PBSMT	25.8	0.508	0.633

In both translation directions, NMT models with subword units score the highest values of all.

Among the NMT models, the subword models outperform the word-based ones by three up to four BLEU. Moreover, the subword-based neural machine translation models outperform the pharse-based machine translation models by approximately six up to seven BLEU.

6. CONCLUSIONS AND FUTURE WORK

We conducted machine translation of Amharic-English with complex morphology in low-resource conditions. Amharic has a rich morphology and uses the Ethiopic script. To tackle the complex morphology and to make an open vocabulary translation, we used subwords. To segment words into subwords, we used BPE. Based on the best practices of prior research, we conducted machine translation in low-data conditions. To this end, we employed PBSMT and transformer-based NMT to train machine translation models.

Both PBSMT and NMT models translate short sentences with comparable accuracy and fluency. However, NMT models produce more accurate and fluent translations of long sentences.

In the automatic evaluation of word-based and subword-based neural machine translation models trained on a benchmark dataset, all subword-based models outperform word-based ones. All neural machine translation models also out perform statistical machine translation models.

We urge on using auxiliary data like monolingual corpora and other word segmentation techniques for further research. We also recommend application of the same approach for machine translation of other Ethiopian local languages.

REFERENCES

Andargachew Mekonnen Gezmu, Andreas Nürnberger, and Binyam Ephrem Seyoum. 2018. Portable spelling corrector for a less-resourced language: Amharic. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018), Miyazaki, Japan. European Languages Resources Association (ELRA).

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, pages 127–133.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Mulu Gebreegziabher Teshome, Laurent Besacier, Girma Taye, and Dereje Teferi. 2015. Phoneme-based English-Amharic statistical machine translation. In AFRICON 2015, pages 1–5. IEEE.

Jennifer Tracey and Stephanie Strassel. 2020. Basic language resources for 31 languages (plus English): The LORELEI representative and incident language packs. In Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-

Resourced Languages (CCURL), pages 277–284, Marseille, France. European Language Resources association.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. CharacTer: Translation edit rate on character level. In Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers, pages 505–510, Berlin, Germany. Association for Computational Linguistics.

Development of dependency parser for Amharic sentences

Mizanu Zelalem Degu*, Worku Birhanie Gebeyehu

Faculty of Computing and Informatics, Jimma Institute of Technology, Jimma University, Jimma, Ethiopia
**Corresponding author, e-mail: mizanu.zelalem@ju.edu.et*

ABSTRACT

Dependency parsing provides information regarding word relationships and has many applications in natural language processing. Several methods of dependency parsing have been proposed in the literature for English and European languages. No sufficient dependency parsing system is available for Amharic, which is a Semitic and a national language of Ethiopia. Due to its morphological structure and low-resource availability, customizing available dependency parser systems is not efficient for Amharic language. In this paper, a novel dependency parser system is proposed for the Amharic language based on a long-short-term memory (LSTM) classifier in two steps, unlabeled dependency parsing, and relation label assignment. First, an arc-eager transition-action classifier was designed and trained on transition configurations generated from Amharic treebank to predict. Then, the output of the classifier is used by the arc-eager transition algorithm to produce an unlabeled dependency tree. Second, a relation-label classifier was designed and trained on pairs of parts of speech tags of the head and the dependent words from the treebank to assign an appropriate label for the dependency relation. Experiments were conducted on 1574 annotated sentences collected from universal-dependency Amharic treebank (1074) and a treebank that was prepared during this study (500). Both classifiers were tested on 30% of the dataset, and 92% and 81% accuracies were found for the transition-action classifier and relation-label classifier, respectively. The proposed system was also evaluated using an unlabeled and labeled attachment score on 30% of the dataset, and 91.54% unlabeled and 86% labeled attachment scores were found. Our experimental results demonstrate that the proposed system can be used for parsing Amharic sentences and as a preprocessing tool during the development of natural language processing tools.

Keywords: *Dependency parsing; Amharic; Under-resourced; LSTM; Arc-eager transition*

1. INTRODUCTION

Amharic is a Semitic language spoken in Ethiopia. It is the second most widely spoken Semitic language in the world next to Arabic. The Amharic language has its alphabet called *Fidel*. Fidel is a syllabary writing system where the consonants and vowels co-exist within each graphic symbol. The language has 33 basic characters, each of which has seven forms depending on which vowel is to be pronounced in the syllable (Gobena, 2011).

Amharic exhibits a root-pattern morphological phenomenon (Tachbelie & Menzel, 2009) in which a root word is combined with a particular prefix or suffix to create a single grammatical form or another stem. Because of this, different authors use a different part of speech (POS) tag sets to incorporate the created word forms. Some authors such as (Demeke & Getachew, 2016) use additional tag sets for each of the derived words and others such as (Seyoum et al., 2018) separates the clitic from the host word and assign a POS tag to the host word and the clitic. Despite a large number of speakers, Amharic is one of the under-resourced languages that need the development of many linguistic tools in general and dependency parser in particular.

Dependency parsing is one of the techniques for analyzing the syntactic structure of a sentence. It represents the head-dependent relationship between words in a sentence classified by functional categories or relationship labels (Rangra, 2015). Dependency parsing provides a clear predicate-argument structure, which is useful in many NLP applications (Nivre, 2010).

There are two ways of performing dependency parsing, rule-based and data-driven, (Nivre, 2005). The first method uses a set of pre-defined grammar rules to parse a sentence and it is restrictive and sensitive to the syntactical structure of a language (Jurafsky & Martin, 2014). The data-driven method, also known as the statistical method, learns the pattern of parsing from an annotated dataset and uses the experience to parse new sentences (Nivre et al., 2005).

The availability of syntactically annotated datasets enhanced the growth of data-driven parsing (Kallmeyer & Maier, 2013). There are two ways of data-driven parsing, transition-based and graph-based. In transition-based dependency parsing, the dependency tree is constructed as a result of the application of a sequence of transition actions on a sentence. The graph-based approach parametrizes a model over smaller substructures to search the space of a valid dependency graph and select the most likely one (Kubler et al., 2009).

Transition-based dependency parsing is pioneered by transition systems. A transition system is an abstract machine that consists of a set of configurations (or states) and transition actions. A transition action converts a given configuration to another configuration. We can see transition systems like a finite-state automaton, which has an initial state, terminal state, and sequence of transition actions. This sequence of transitions from starting state to the final or terminal state produces a dependency structure for a given input sentence (Kubler et al., 2009). There are two popular transition systems, arc-standard and arc-eager transition systems (Kuhlmann et al., 2011). Both have three arguments, a list of processed tokens (stack), a list of unprocessed tokens (buffer), and a sequence of transition actions made so far. Arc–standard transition system has three transitions, which are Shift, Right-arc, and Left-arc. *Shift* removes the word from the front of the buffer and pushes on the stack, *right-arc* asserts a head-dependent relation between the second top word on the stack and the word at the top of the stack and pops the stack, *left-arc* asserts a head-dependent relation between the word at the top of the stack and the second top word on the stack and removes the second top word from the stack. Arc-eager has four transition states, which are *left-arc*, *right-arc*, *shift*, and *reduce*. *Left-arc* asserts a head-dependent relation between the word at the front of the buffer and the word at the top of the stack and pops the stack. *Right-arc* asserts a head-dependent relation between the word at the top of the stack and the word at the front of the buffer and pushes the word from the front of the buffer onto the stack. *Shift* pushes the word from the front of the buffer onto the stack (Kuhlmann et al., 2011).

The application of artificial neural networks in the development of a dependency parsing system was initiated by Chen and Manning (Chen & Manning, 2014). They applied distributed word representation for encoding words, their POS tag, and arc-label in the configuration and used an artificial neural network classifier to predict the next transition action of a given configuration. Nowadays, the availability of deep learning architectures like the LSTM network and frameworks such as Tensorflow, Theano, and Keras leads to more sophisticated results for dependency parsing. Dyer et.al (2015) presented a dependency parser using

stacked LSTM to represent a complete history of the configuration, the state of the stack, state of buffer, and state of transition sequences made so far. (Kiperwasser & Goldberg, 2016) used bi-directional LSTM based feature extractions for developing a transition-based dependency parser.

Many works of literature have proposed the development of language-processing tools, but they were mainly concentrated on English, European, and East-Asian languages. The work by (Nivre et al., 2005) presented a development of a universal dependency parser called Malt parser to parse sentences from different languages without making language-specific modifications to the system. Languages incorporated during the study were few namely English, Dutch, Turkish, and Italian. However, parsing in morphologically rich languages is different. In morphologically rich languages dependency relation exists not only between orthographic words (space-delimited tokens) but also within the word itself (Tsarfaty et al., 2013). For example, in Amharic, an orthographic word combines some syntactic words into one compact string. These words can be function words such as prepositions, conjunctions, and articles (Seyoum et al., 2018).

The study in (Goldberg & Elhadad, 2009) used maximum spanning tree (MST) and Malt parser dependency parsers on Hebrew, one of the morphologically rich languages, treebank. As the authors found from their study, both systems were inefficient to represent the morphological richness of the language. The study in (Marton et al., 2013) presented a way to improve the performance of Malt parser for parsing Arabic sentences by adding lexical, inflectional, and POS tags. However, the system was unable to show significant improvement relative to the unmodified version of the parser.

The only attempt for the development of an Amharic dependency parser was proposed by (Gasser, 2010). They presented a development of dependency grammar using the extensible dependency grammar (XDG) rules. However, the work lacks considering the dependency relation that founds between root word and its morphological affix (clitic), and the performance of the developed grammar was not evaluated due to lack of annotated dataset by the time.

To this end, this study aims to develop a dependency parser system for Amharic language that can produce an unlabeled and labeled dependency tree for a given Amharic sentence.

2. METHODS

In this study, three classifiers were trained and tested on an Amharic treebank that has 1574 annotated Amharic sentences. The first is the transition-action classifier which was trained on a sequence of transition configurations generated from the treebank. The second is the relation-label classifier which was trained on part of speech (POS) tags of head-dependent pairs of the treebank. The last one is the POS tag classifier which was trained on twelve different features of words extracted from the treebank. As far as our knowledge, no POS tagger system is developed based on the tag set from Amharic treebank. Therefore, a POS tagger was developed to make the proposed dependency parser complete. A web-based application was also developed to make the proposed system publicly available. The overall workflow of the proposed system is presented in Figure 1.

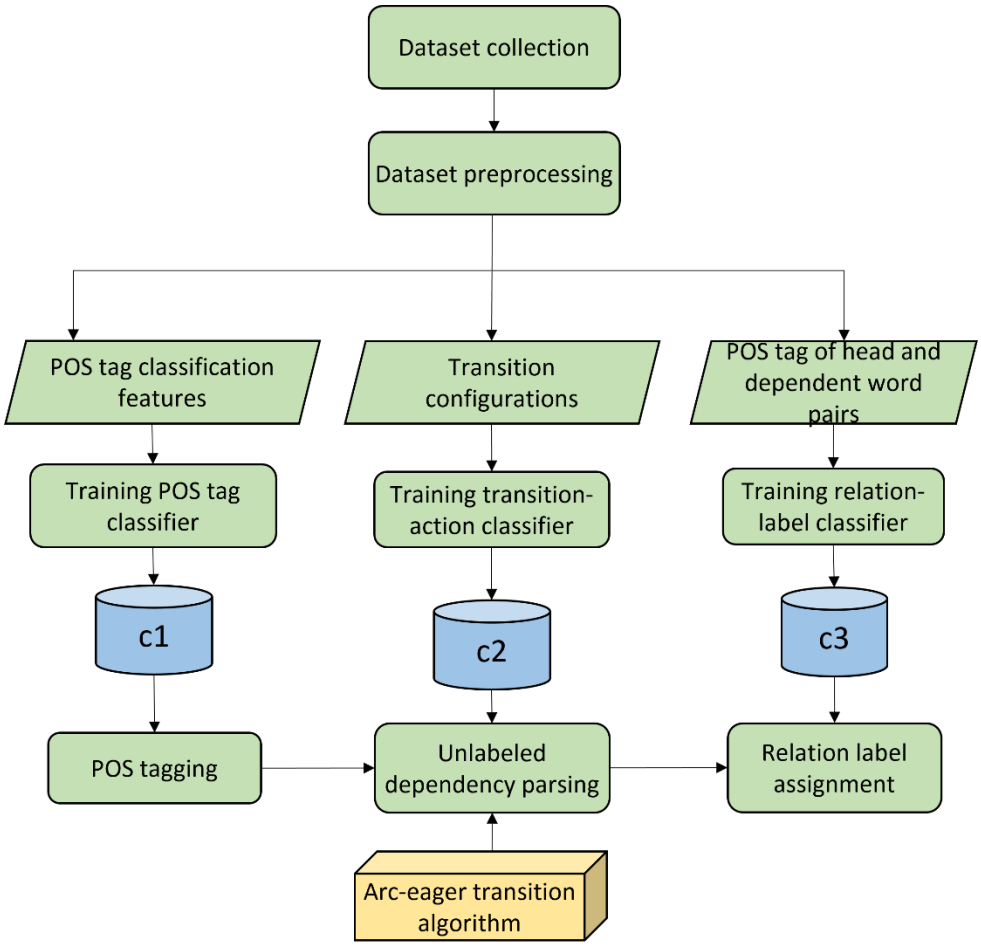


Figure 1: Overall workflow of the study

2.1. Dataset collection

In this study, initial datasets were collected from the Amharic treebank annotated in (Seyoum et al., 2018). The treebank contains 1074 sentences. Annotation for another 500 sentences was performed during this study with the help of linguist experts from Jimma University. The annotation method followed for the new treebank such as extracting the clitic, POS tagging, and relation label assignment was similar to the development of the previous treebank and several revisions were performed to keep it consistent with the previous treebank. Sentences for the new treebank were collected from fiction and history books for the sake of relative structural correctness. A sample of the Amharic treebank is shown in table 1. The table shows a dependency relation for the sentence እርሱ ን ማ አውቅ ኧው የል ኧ ም?/ *irisu ni ma āwik'i ewi yeli e mi?* (Don't I know him?)

2.2. Data preprocessing

In this step, the Amharic treebank was processed to train and test the three classifiers.

2.2.1. Feature extraction for POS tag classifier

For training and testing the POS tag classifier, features of a word such as the number of terms(words) in the sentence, the word itself, the position of the word (*is first*, *is last*), the previous-word, next-word, prefix, and suffix were extracted for each word from the treebank.

Table 1: Sample annotated sentence from Amharic treebank.

Index	Word	Stem word	Universal POS tag	Amharic POS tag	Morphological features	Head index	Dependency label
1	እርሱ	እርሱ	PRON	PRON	_	4	obj
2	ን	ን	PART	ACC	_	1	case
3	ማ	ማ	INTJ	INTJ	_	1	discourse
4	አውቅ	አውቅ	VERB	VERB	_	6	ccomp
5	እው	እው	PRON	SUBJC	Number=Sing|Person=3	4	nsubj
6	የል	የል	VERB	VERB	_	0	root
7	እ	እ	PRON	SUBJC	Gender=Masc|Number=Sing|Person=3	6	nsubj
8	ም	ም	PART	NCM	_	6	discourse
9	?	?	PUNCT	PUNCT	_	6	punct

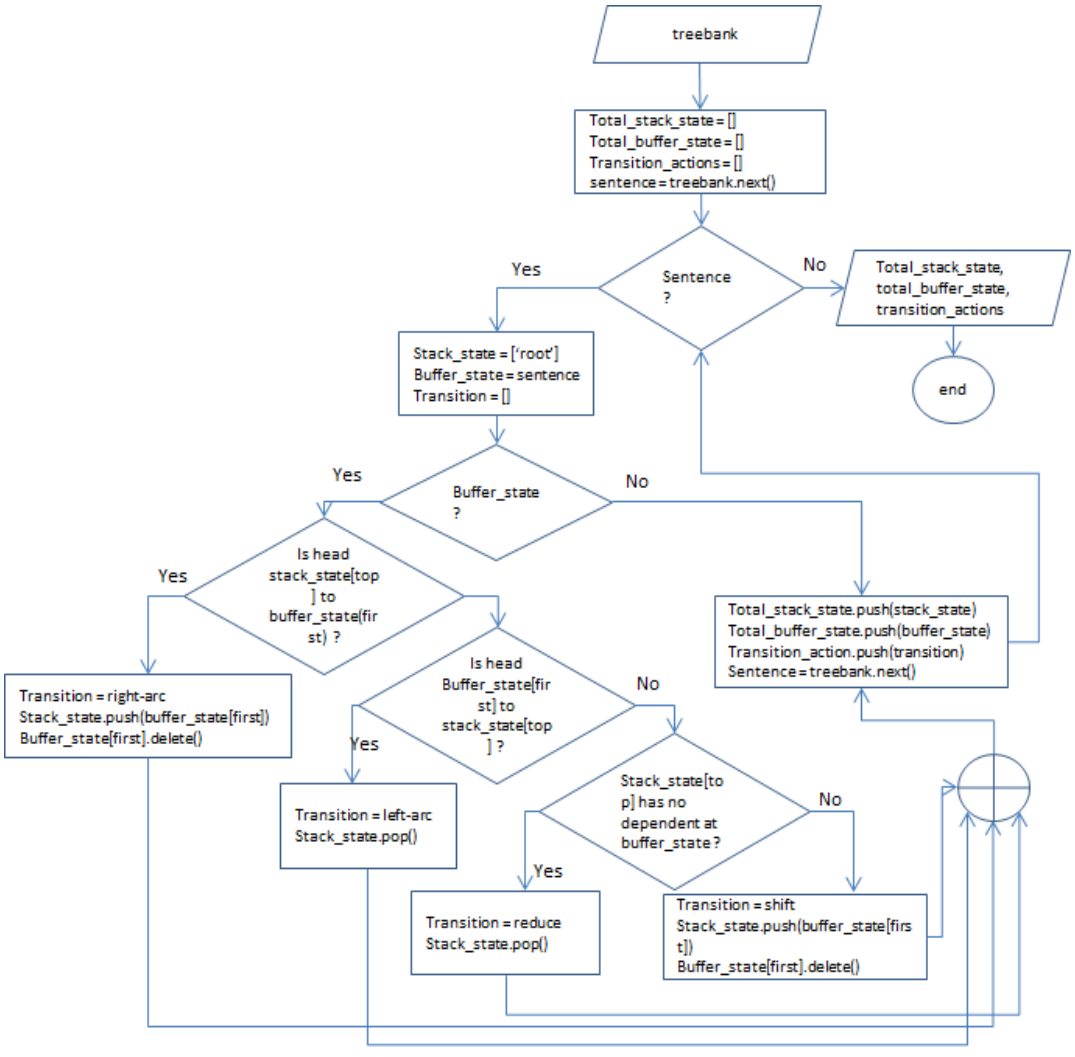


Figure 2: Flowchart of reversed arc-eager transition algorithm for extracting transition configuration.

2.2.2. Generating transition configurations

Transition configuration is used to train and test the transition-action classifier. It is a collection of *stack state*, *buffer state*, and *transition-action*. *Stack state* is a list of words and their POS tag moved from the buffer due to *left-arc* or *shift* transition action. *Buffer state* is a list of words and their POS tag that are waiting for a dependency attachment to be performed. *Transition-action* is an *arc-eager* transition action that is going to be performed for a given stack and buffer state. To generate transition configurations from the treebank, a reversed version of the *arc-eager* transition algorithm was developed. Figure 2 illustrates a flowchart of the designed algorithm to generate transition configurations.

2.2.3. Extracting POS tag of head-dependent pairs

Pairs of POS tags of head and dependent words were extracted from the treebank for training and testing the relation-label classifier. The extraction was performed by finding the headword indicated using a *head-index* in front of the dependent word and grouping POS tags of the dependent and the headword along with their relation label.

2.3. Training and testing classifiers

In this paper, three deep learning classifiers were trained. These are the POS tag classifier, transition-action classifier, and relation-label classifier. Various hyper-parameter tunings such as batch size, number of epochs, activation function, and optimizer function were performed to get an efficient classifier model. All the classifiers were designed and trained using the TensorFlow-based Keras deep learning framework.

2.3.1. Training and testing transition-action classifier

For constructing an unlabeled dependency tree, predicting transition actions is the first step. For this purpose, a transition-action classifier was trained and tested on a collection of transition configurations. The architecture of the classifier is illustrated in Figure 3. The model was designed to accept four inputs, *stack state of words*, *stack state of POS tags*, *buffer state of words*, and *buffer state of POS tags*. Then each input is vectorized using an embedding layer. Embedding layers for *stack state of POS tags* and *buffer state of POS* tags have *26* output dimensions representing 26 unique POS tags found in the treebank including *zero-padding* (*PAD), out of vocabulary (OOV),* and *ROOT*. Embedding layers for *stack state of words* and *buffer state of words* have *1478* output dimensions representing *1478* unique words of the treebank including *PAD* and *OOV*. The output of embedding layers for *stack state* and *buffer state* were merged using the dot product layer. LSTM layers were used to extract sequence information from the output of the dot product layers and then the information was merged using a concatenation layer. LSTM layer with 256 output dimensions was used for final feature extraction and finally, a time-distributed dense layer with four output units representing *shift*, *reduce*, *left-arc*, and *right-arc* transition actions were used for classification. The model was trained on *adam* optimizer with a learning rate of 0.01 for 30 epochs.

2.3.2. Training and testing relation-label classifier

In transition-based dependency parsing, particularly in arc-eager, the number of transition-action labels required for training a classifier is 2n+2, where n is the number of relation labels of the language. This is because for the left-arc and the right arc transition actions there is n number of alternatives. However, there

is no dataset for the Amharic language that can give enough examples for such several classes to train a classifier. In addition, as observed from the experiments, the relation label between a dependent and headword can be determined by the POS tag of the words. For instance, the relation label between ነብር/nebir(Tiger) and ኡ/u (determinant indicator post-fix) in the sentence ነብሩ ተገደለ ።/ nebiru tegedele (The tiger has been killed.) is det, because the POS tag of the word ኡ/u is DET and the POS tag of ነብር/nebr is SUBJC.

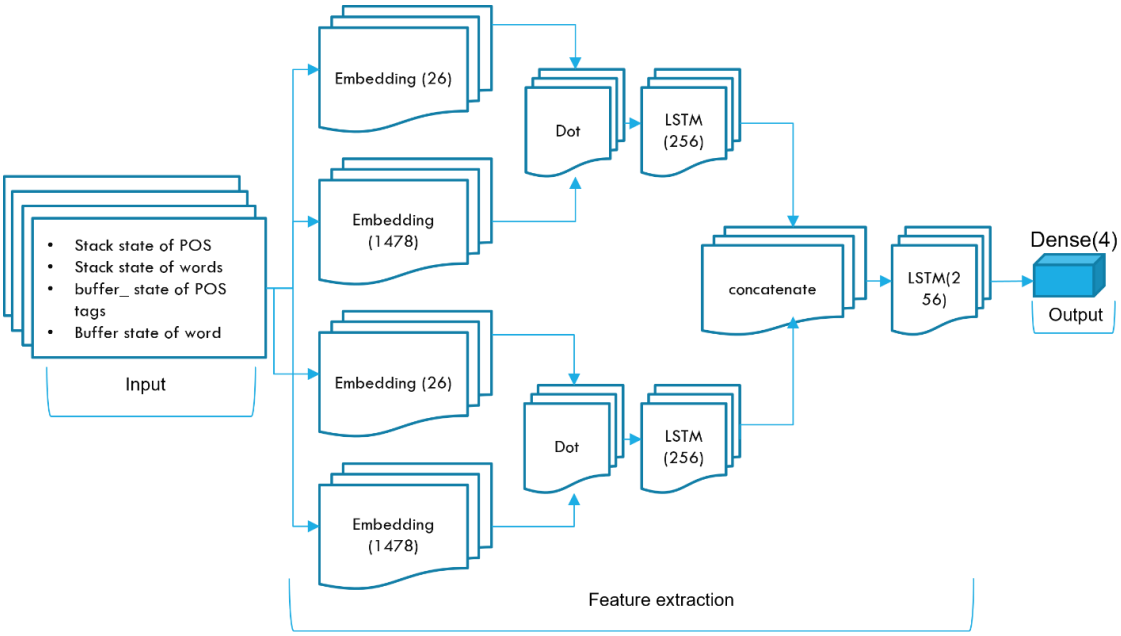


Figure 3: Architecture of transition-action classifier

Therefore, unlabeled dependency parsing and relation label assignment were performed using two classifiers. Unlabeled dependency parsing was performed using transition-action classifier, 2.3.1. which requires four classes, and the relation-label assignment was performed using a relation-label classifier that requires n classes. As illustrated in figure 4, the classifier is composed of an input layer to accept a list of POS tags of the head and the dependent word, an embedding layer for vectorizing the inputs, and an LSTM layer for extracting sequence information, and a time-distributed dense layer with *36* output units, for 36 relation labels. The model was trained using *adam* optimizer with a 0.01 learning rate for 100 epochs.

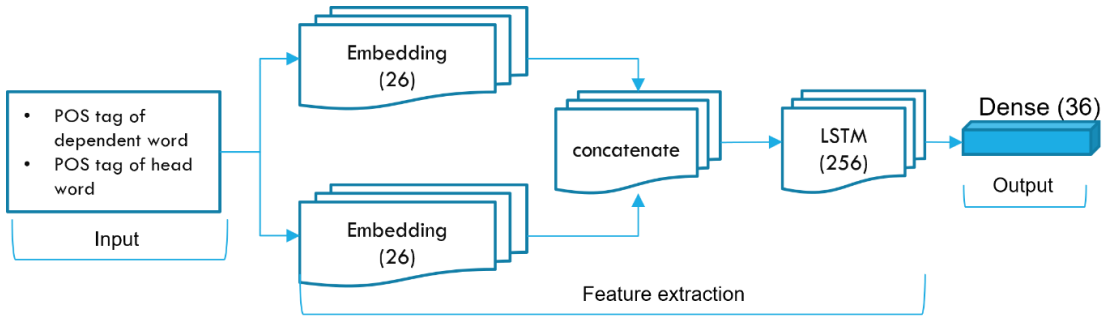


Figure 4: Dependency relation type classifier

The role of the LSTM in the relation-label classifier was to extract historical information from the input to predict an appropriate relation label. For instance, Table 2 shows the relation label of a head-dependent pair for a similar sentence except *(a)* a sentence in which its subject is explicitly stated, አስቴር/*aster*, and *(b)* a sentence with no explicit subject. To start from the second case, the clitic ኣች/*aach is* used to indicate a subject, a third-person female. Therefore, the relation label of the word ኣች/*aach* with its head becomes *nsubj*. In the first case, the clitic ኣች/*aach* holds the same information as the first sentence, but the subject is already mentioned at the beginning of the sentence. At this time, the relation label of ኣች/*aach* with its head will be *expl*.

Table 2: Sample Head-dependent pair dataset for the dependency-relation label classifier. (a) When the subject is explicitly stated, (b) When the subject is not explicitly stated

(a)

Dependent word	Dependent word's POS tag	Headword's POS tag	Headword	Dependency relation label
አስቴር/*aastier*(Name of a person)	PROPN	VERB	መጥ	nsubj
መጥ/*met'* (come)	VERB	root	root	root
ኣች/*aach* (subject indicator postfix)	SUBJC	VERB	መጥ	expl

(b)

Dependent word	Dependent word's POS tag	Headword's POS tag	Headword	Dependency relation label
መጥ/met'	VERB	root	root	root
ኣች/aach (subject indicator postfix)	SUBJC	VERB	መጥ	nsubj

2.3.3. Training and testing POS tag classifier

The designed POS tag classifier has three layers, input layer, hidden layer, and output layer. The input layer has 12 units to accept features of the words and the output layer has 23 output units for the 23 POS tag set found in the Amharic treebank, presented in Table 3.

This classifier was trained using *adam* optimizer for 10 epochs with an *early-stopping* mechanism. Figure 5 illustrates the architecture of the POS tag classifier.

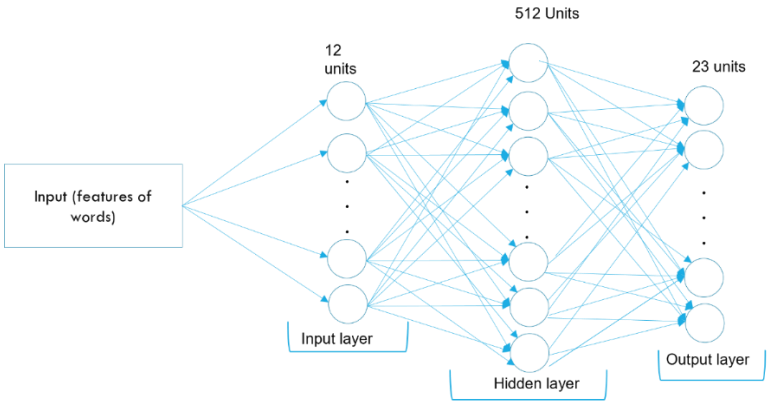


Figure 5: POS tag classifier

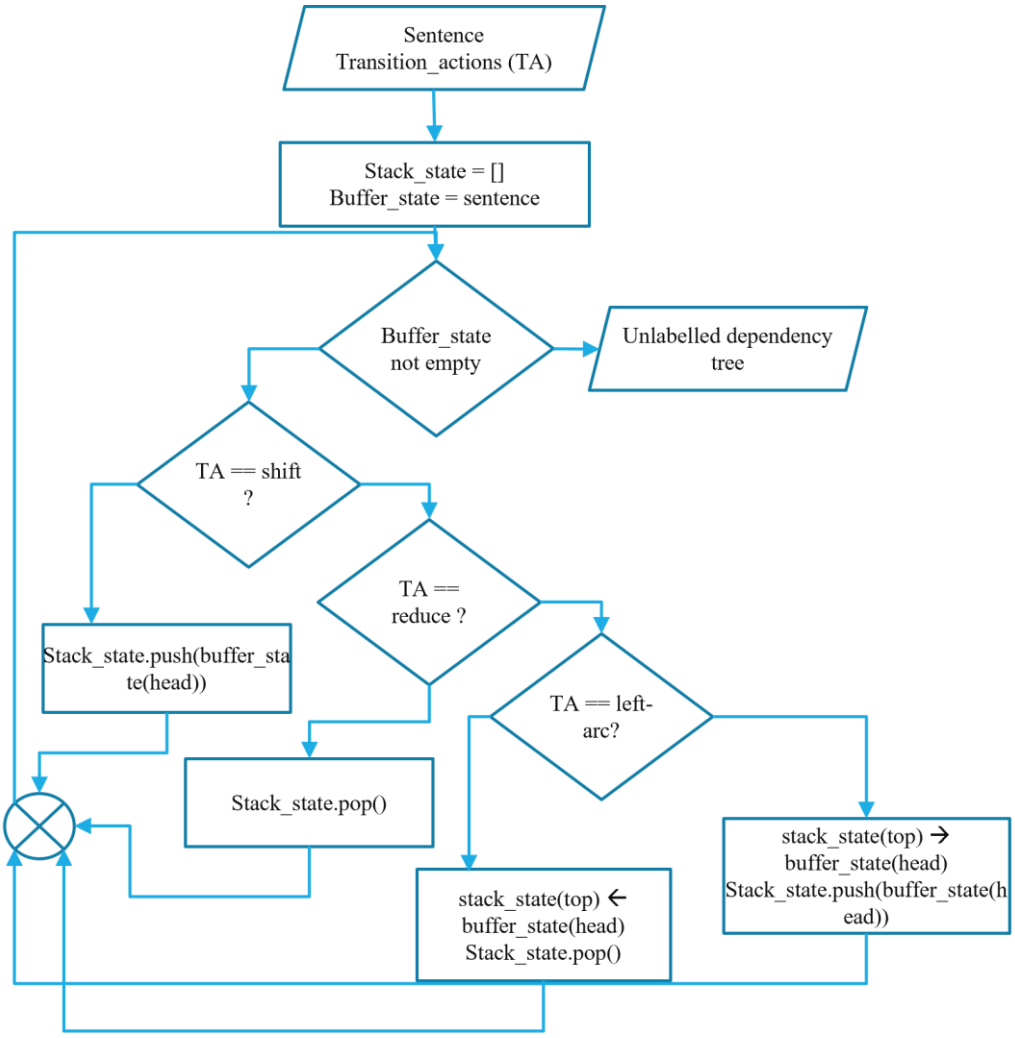


Figure 6: Flowchart of Arc-eager transition algorithm

2.4. Constructing dependency tree

2.4.1. Constructing unlabeled dependency tree

An unlabeled dependency tree is a dependency tree in which the dependency relation between the head and the dependent word has no label. In this study, the construction of an unlabeled dependency tree was performed by following the rules of the arc eager-transition algorithm. As illustrated in the flowchart in Figure 6, the algorithm accepts a sentence and transition actions that are predicted by the transition-action classifier as input and it iteratively constructs an unlabeled dependency tree. The algorithm has four rules namely, *shift*, *reduce*, *right-arc* and *left-arc*. *Shift* pushes the first word from the buffer to the stack. *Reduce* pops the stack. *Right-arc* adds a *dependency-arc* between the word at the top of the stack and the first word of the buffer and pushes the first word from the buffer into the stack. *left-arc* adds a *dependency-arc* between the first word of the buffer and the word on the top of the stack and pops the stack.

2.4.2. Relation label assignment

This step will be performed right after the construction of the unlabeled dependency tree. It is used to assign a relation label for the unlabeled dependency tree by using the relation-label classifier.

3. RESULTS

3.1. Data preprocessing results

3.1.1. Result of extraction of transition configurations.

In this preprocessing step, 26,242 transition configurations were extracted from the Amharic treebank. Table 3 illustrates a sample of the raw Amharic treebank and the extracted transition configurations.

Table 1: Amharic POS tag set from Amharic universal dependency Treebank

1	እርሱ	እርሱ	PRON	PRON	_	4	obj
2	ን	ን	PART	ACC	_	1	case
3	ማ	ማ	INTJ	INTJ	_	1	discourse
4	አውቅ	አውቅ	VERB	VERB	_	6	ccomp
5	ኧው	ኧው	PRON	SUBJC	Number=Sing|Person=3	4	nsubj
6	የል	የል	VERB	VERB	_	0	root
7	ኧ	ኧ	PRON	SUBJC	Gender=Masc|Number=Sing|Person=3	6	nsubj
8	ም	ም	PART	NCM	_	6	discourse
9	?	?	PUNCT	PUNCT	_	6	punct

Table 4a: Sample of the raw Amharic treebank

Part of speech	**Abbreviation**	**Part of speech**	**Abbreviation**
adjective	ADJ	particle	PART
adposition	ADP	position marker	POSM
adverb	ADV	pronoun	PRON
auxiliary	AUX	proper noun	PROPN
coordinating conjunction	CCONJ	punctuation	PUNCT
determiner	DET	relationship marker	RLP
interjection	INTJ	subordinating conjunction	SCONJ
indirect relationship marker	IRLP	subject	SUBJC
noun class marker	NCM	verb	VERB
negation	NEG	object	OBJC
noun	NOUN	particle	PART
numeral	NUM		ACC

Table 4b: Sample of the extracted transition configurations

S.No.	Stack_state of words	Stack_state of POS tags	Buffer_state words	Buffer_state of POS tags	Transition action
1	[root]	[root]	[እርሱ,ን,ማ,አውቅ, ኧው, የል,ኧም,?]	[PRON,ACC,INTJ,VERB, SUBJC,VERB,SUBJC,NCM, PUNCT]	Shift
2	[root, እርሱ]	[root, PRON]	[ን,ማ,አውቅ,ኧው, የል,ኧ,ም,?]	[ACC,INTJ,VERB, SUBJC,VERB,SUBJC,NCM, PUNCT]	Right-arc

3	[root, እርሱ, ን]	[root, PRON,ACC]	[ማ,አዉቅ,ኧው ,የል,ኧ,ም,?]	[INTJ,VERB, SUBJC,VERB,SUBJC,NCM, PUNCT]	Reduce
4	[root, እርሱ]	[root, PRON]	[ማ,አዉቅ,ኧው, የል,ኧ,ም,?]	[INTJ,VERB, SUBJC,VERB,SUBJC,NCM, PUNCT]	Right-arc
5	[root, እርሱ, ማ]	[root, PRON, INTJ]	[አዉቅ,ኧው,የል, ኧ,ም,?]	[VERB, SUBJC,VERB,SUBJC,NCM, PUNCT]	Reduce
6	[root, እርሱ]	[root, PRON]	[አዉቅ,ኧው,የል, ኧ,ም,?]	[VERB, SUBJC,VERB,SUBJC,NCM, PUNCT]	Left-arc
7	[root]	[root]	[አዉቅ,ኧው,የል, ኧ,ም,?]	[VERB, SUBJC,VERB,SUBJC,NCM, PUNCT]	Shift
8	[root, አዉቅ]	[root, VERB]	[ኧው,የል,ኧ,ም,?]	[SUBJC,VERB,SUBJC,NCM, PUNCT]	Right-arc
9	[root, አዉቅ, ኧው]	[root, VERB, SUBJC]	[የል,ኧ,ም,?]	[VERB,SUBJC,NCM, PUNCT]	Reduce
10	[root, አዉቅ]	[root, VERB]	[የል,ኧ,ም,?]	[VERB,SUBJC,NCM, PUNCT]	Left-arc
11	[root]	[root]	[የል,ኧ,ም,?]	[VERB,SUBJC,NCM, PUNCT]	Right-arc
12	[root, የል]	[root, VERB]	[ኧ,ም,?]	[SUBJC,NCM, PUNCT]	Right-arc
13	[root, የል, ኧ]	[root, VERB, SUBJC]	[ም,?]	[NCM, PUNCT]	Reduce
14	[root, የል]	[root, VERB]	[ም,?]	[NCM, PUNCT]	Right-arc
15	[root, የል, ም]	[root, VERB, NCM]	[?]	[PUNCT]	reduce
16	[root, የል]	[root, VERB]	[?]	[PUNCT]	Right-arc

3.1.2. Result of extraction of dependent-head POS tag pairs

For training and testing the relation label classifier, *15,534* dependent-head pairs have been extracted from the treebank. Table 5 shows the extracted dependent-head pairs for a sample treebank illustrated shown in Table 2 (a).

Table 5: Sample of the extracted POS tag pairs of the head and dependent word

Index	POS tag of Dependent word	POS tag of Headword	relation label
1	PRON	VERB	obj
2	ACC	PRON	case
3	INTJ	PRON	discourse
4	VERB	VERB	ccomp
5	SUBJC	VERB	nsubj
6	VERB	root	root
7	SUBJC	VERB	nsubj

8	NCM	VERB	discourse
9	PUNCT	VERB	punct

3.1.3. Result of extraction of word features for POS tagger

For *15,534* words (terms) that are found in the Amharic treebank, twelve features were extracted for training and testing the POS tag classifier. Table 6 illustrates a list of extracted features for each word in a sample sentence ልጅ ኡ በር ላይ ቆም ኧ ።/*lj u ber lay k'om ee*. (The boy stands at the door).

3.2. Result of the classifiers

Three classifiers, transition action classifier, relation label classifier, and POS tag classifier, have been trained and tested on a preprocessed dataset. For training and testing all of the classifiers, the dataset was randomly divided into 70% and 30% respectively.

Table 6: Sample of the extracted features for the POS tag classifier

No terms	term	Is first	Is last	Prefix 1	prefix 2	Prefix 3	Suffix 1	Suffix 2	Suffix 3	Prev. word	Next. word	POS tag
7	ልጅ	TRUE	FALSE	ል	ልጅ	ልጅ	ጅ	ልጅ	ልጅ		ኡ	NOUN
7	ኡ	FALSE	FALSE	ኡ	ኡ	ኡ	ኡ	ኡ	ኡ	ልጅ	በር	DET
7	በር	FALSE	FALSE	በ	በር	በር	ር	በር	በር	ኡ	ላይ	NOUN
7	ላይ	FALSE	FALSE	ላ	ላይ	ላይ	ይ	ላይ	ላይ	በር	ቆም	ADP
7	ቆም	FALSE	FALSE	ቆ	ቆም	ቆም	ም	ቆም	ቆም	ላይ	ኧ	VERB
7	ኧ	FALSE	FALSE	ኧ	ኧ	ኧ	ኧ	ኧ	ኧ	ቆም	።	SUBJC
7	።	FALSE	TRUE	።	።	።	።	።	።	ኧ		PUNCT

3.2.1. Result of the transition-action classifier

The transition action classifier was evaluated on 30% of the generated transition action configurations and 92% accuracy was found. Figure 7 shows training and testing curves for the transition action classifier.

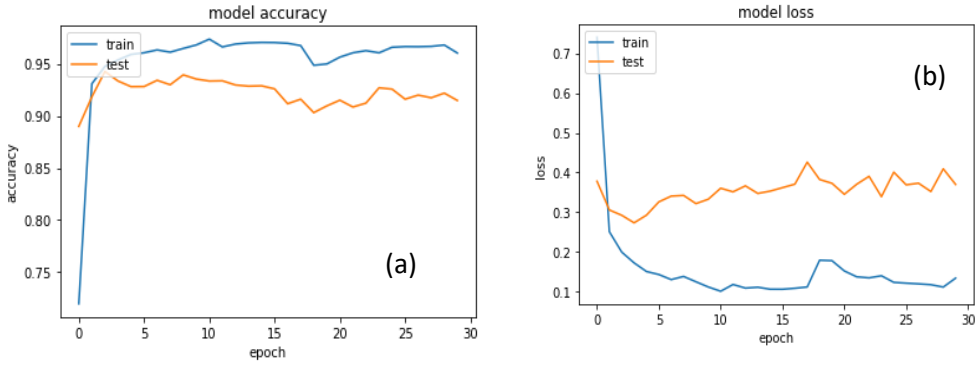


Figure 7: Training and testing curves of transition-action classifier (a) Accuracy of the classifier (b) Loss of the classifier

3.2.2. Result of the relation-label classifier

Figure 8 illustrates training and testing curves for the relation-label classifier. The classifier was evaluated on 30% of the dataset and 81% accuracy was found.

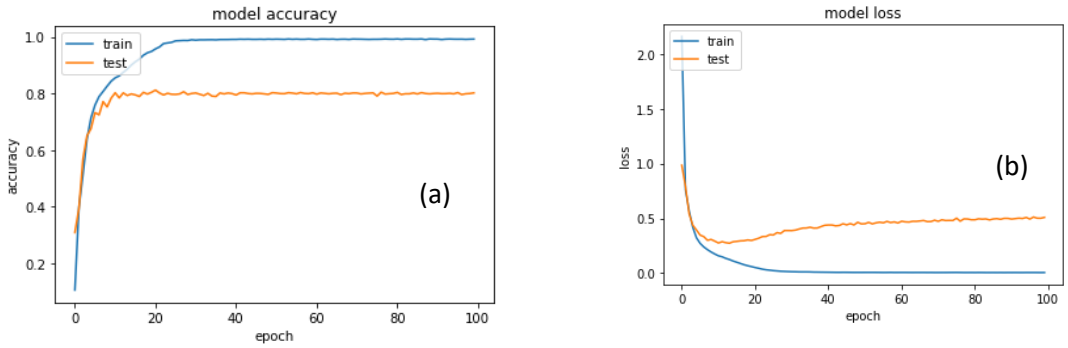


Figure 8: Training and testing curves for the relation-label classifier. (a) Accuracy curve of the classifier (b) Loss curve of the classifier

3.2.3. Result of the POS tag classifier

The classifier was tested on 30% of the extracted features from the Amharic treebank and achieved 88.1% accuracy. The training curves of the classifier are illustrated in Figure 9.

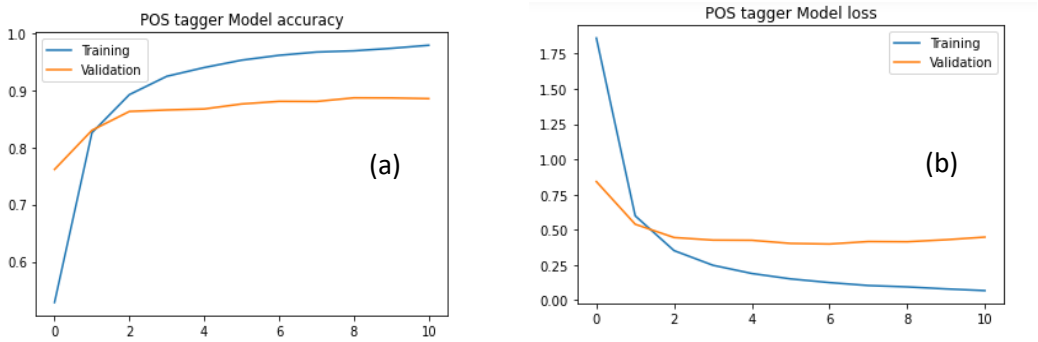


Figure 9: Training and testing curves for the POS tag classifier. (a) Accuracy curve of the model (b) Loss curve of the model.

3.2.4. Result of the construction of the unlabeled dependency tree

An unlabeled dependency tree was constructed by using an arc-eager transition algorithm. The algorithm uses the predicted transition actions by the transition-action classifier as a direction. The performance of the unlabeled dependency parser was evaluated on 30% of the dataset and a 91.54% *unlabeled attachment score* was found. Figure 10(a) shows a chart of unlabeled attachment scores.

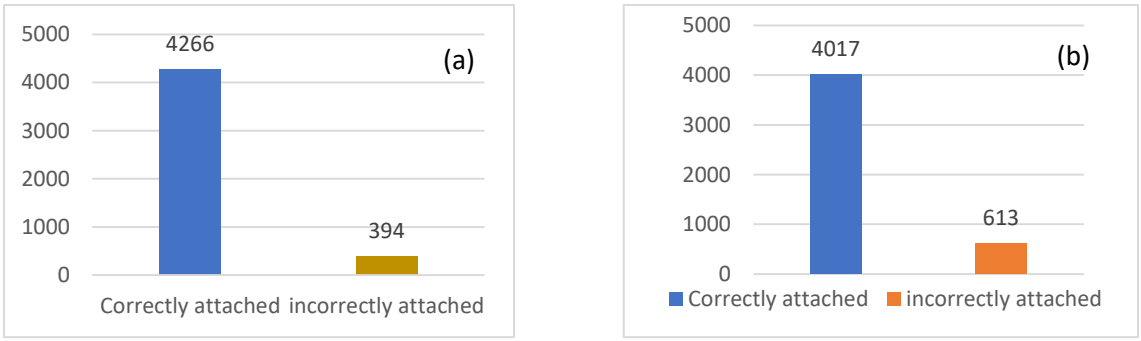


Figure 10: Attachment score of the system. (a) Unlabeled attachment score, (b) labeled attachment score

3.2.5. Result of relation label assignment

The constructed unlabeled dependency tree was then labeled using the relation label classifier to produce labeled dependency tree. The system was evaluated on 30% of the treebank and 86% *labeled attachment score* was found. Figure 10(b) illustrates the labeled attachment score of the system.

3.3. Web-based application development for the proposed system

A web-based application was developed to make the proposed system accessible. The system allows a user to enter an Amharic sentence and it produces a labeled and unlabeled dependency tree along with its graphical representation. Sample screenshots of the system are presented in Figure 11. The application has been hosted on *Streamlit* and can be accessed from the following link,

https://share.streamlit.io/mizgithub/amharic_dependency_parserapp/index.py.

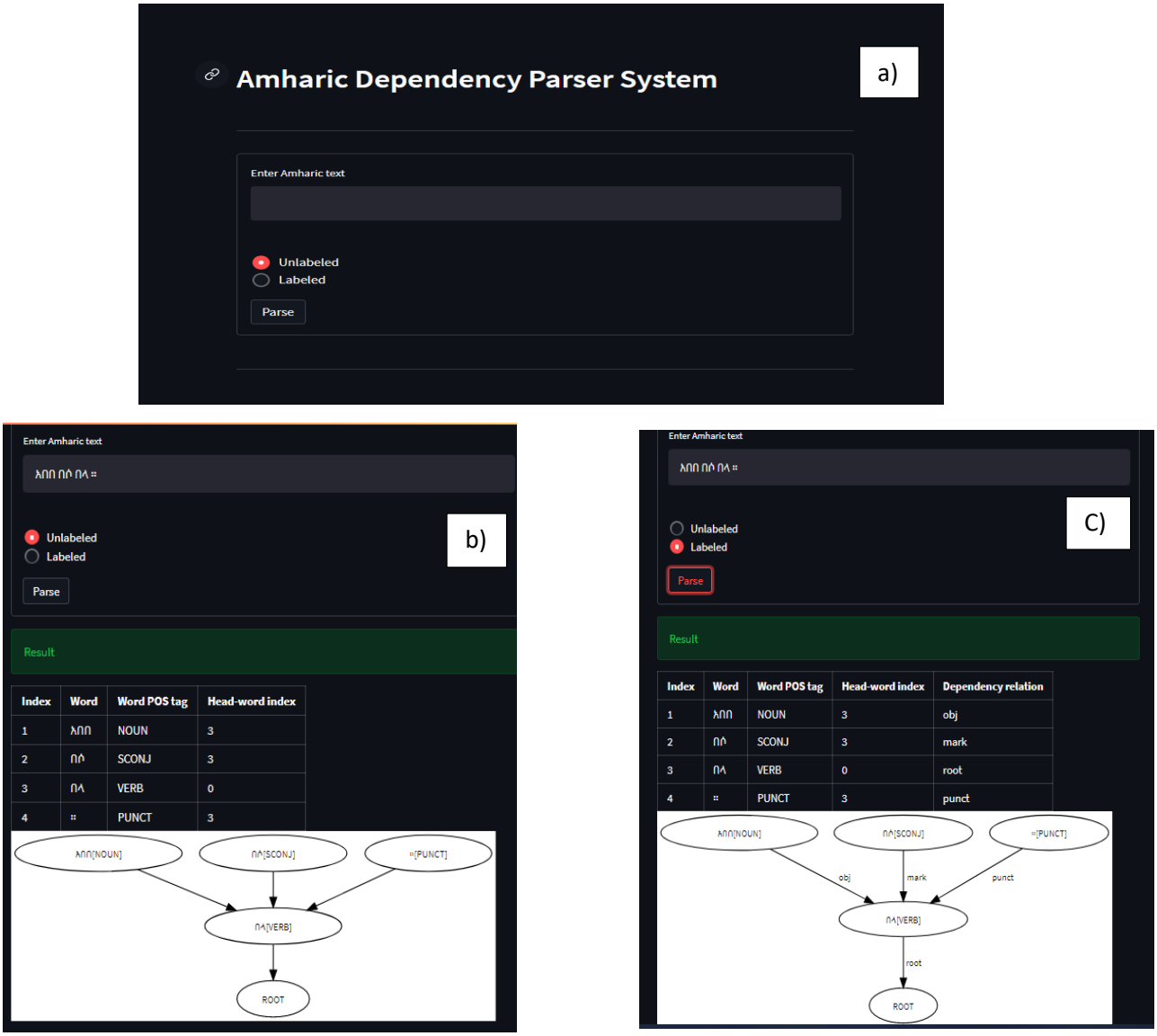


Figure 11: Amharic dependency parser user interface. (a) Home page (b) Output of unlabeled dependency paring (c) Output of labeled dependency parsing.

4. DISCUSSION

Dependency parsing is the process of analyzing the grammatical structure of a sentence based on the dependencies between the words in a sentence. It plays an important role in many natural language processing applications such as machine translation, language modeling, information extraction, relation extraction, etc. Several methods of dependency parsing have been proposed in the literature for English and European languages. No sufficient dependency parsing system is available for Amharic.

To address this problem, a dependency parser was developed for the Amharic language. The system was made up of three modules. The first module is the POS tagger that was used to give POS tags for words of a given sentence. The second is the unlabeled dependency parser that was used to construct an unlabeled dependency tree. The last is the relation-label assignment that was used to give a relation label for each of the relations in the unlabeled dependency tree.

The POS tagger module contains a classifier that uses different attributes of a word such as *position*, *previous-term*, *next-term*, *prefix*, *suffix*, etc. to predict its POS tag.

In the second module, a transition-action classifier is used to predict a sequence of transition actions for a given sentence. After that, the transition actions are used by the arc-eager transition algorithm to construct the unlabeled dependency tree.

In the last module, a relation label assignment for the unlabeled dependency tree was performed using a relation-label classifier. The classifier determines the relation label from the POS tag of the dependent and head pairs.

The developed system can be used for parsing Amharic sentences and as a preprocessing tool for developing other NLP applications. This work can also be used as a baseline for further improvement of a dependency parsing system for the Amharic language.

5. CONCLUSION

This paper presented the development of a dependency parser system for the Amharic language in a low-resource setting. Three modules for POS tagger, unlabeled dependency parsing, and relation label assignment were developed. In each of the modules, deep learning classifiers were trained and tested on the preprocessed Amharic treebank. The modules were then integrated to produce a complete labeled dependency tree for a given Amharic sentence. According to the results, the proposed system can be used for parsing Amharic sentences, for constructing treebank, as a preprocessing tool for the development of other NLP applications. The system can be also used as a baseline for the development and improvement of dependency parsers for Amharic language.

REFERENCES

Chen, D., & Manning, C. (2014). *A Fast and Accurate Dependency Parser using Neural Networks* Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar.

Demeke, G., & Getachew, M. (2016). *Manual annotation of Amharic news items with part of speech tags and their challenges* In ELRC working papers.,

Gasser, M. (2010). *A dependency grammar for Amharic* Indiana University].

Gobena, M. K. (2011). *Implementing an open source amharic resource grammar in GF*

Goldberg, Y., & Elhadad, M. (2009, oct). Hebrew Dependency Parsing: Initial Results.*Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)* Paris, France.

Jurafsky, D., & Martin, J. H. (2014). Speech and language processing.

Kallmeyer, L., & Maier, W. (2013). Data-Driven Parsing using Probabilistic Linear Context-Free Rewriting Systems [journal article]. *Computational Linguistics*, *39*(1), 87-119. https://doi.org/10.1162/COLI_a_00136

Kiperwasser, E., & Goldberg, Y. (2016). Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations [journal article]. *Transactions of the Association for Computational Linguistics*, *4*, 313-327. https://doi.org/10.1162/tacl_a_00101

Kubler, S., McDonald, R., Nivre, J., & Hirst, G. (2009). *Dependency Parsing*. Morgan and Claypool Publishers.

Kuhlmann, M., Gómez-Rodríguez, C., & Satta, G. (2011, jun). Dynamic Programming Algorithms for Transition-Based Dependency Parsers.*Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* Portland, Oregon, USA.

Marton, Y., Habash, N., & Rambow, O. (2013). Dependency Parsing of Modern Standard Arabic with Lexical and Inflectional Features [journal article]. *Computational Linguistics*, *39*(1), 161-194. https://doi.org/10.1162/COLI_a_00138

Nivre, J. (2005). *Two strategies for text parsing*.

Nivre, J. (2010). Dependency Parsing. *Lang. Linguistics Compass*, *4*, 138-152.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., & Marsi, E. (2005). MaltParser: A Language-Independent System for Data-Driven Dependency Parsing. *Natural Language Engineering*, *13*, 95 - 135.

Rangra, R. (2015). BASIC PARSING TECHNIQUES IN NATURAL LANGUAGE PROCESSING.

Seyoum, B. E., Miyao, Y., & Mekonnen, B. Y. (2018, may). Universal Dependencies for Amharic.*Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* Miyazaki, Japan.

Tachbelie, M. Y., & Menzel, W. (2009). Amharic Part-of-Speech Tagger for Factored Language Modeling. RANLP,

Tsarfaty, R., Seddah, D., Kübler, S., & Nivre, J. (2013). Parsing Morphologically Rich Languages: Introduction to the Special Issue. *Computational Linguistics*, *39*(1), 15-22. https://doi.org/10.1162/COLI_a_00133

Offline Handwritten Text Recognition of Learning Techniques

Mesfin Geresu[1,*], Million Meshesha[2], Elsabet Wedajo[1]

[1]*Information Science Department, Jimma Institute of Technology, Jimma University, Jimma, Ethiopia*
[2]*Information Science Department, Addis Ababa University, Addis Ababa, Ethiopia*
**Corresponding author, e-mail: fishmesh2000@yahoo.com*

ABSTRACT

Handwriting recognition of historical documents is still largely unsolved problem in the field of pattern recognition. This study investigates how the state-of-the-art deep learning techniques perform handwriting recognition in the context of historical Ge'ez manuscripts. Though Ge'ez was the language of literature in Ethiopia until the middle of the 19th century, it is underrepresented in the research areas of document image analysis and recognition. Thus handwriting recognition system is proposed and its architecture is comprised of pre-processing (binarization and skew estimation), page layout analysis, recognition model, and post-processing tasks. For each task, experimental setup is designed. In the task of binarization, four binarization methods (Otsu's global method, Otsu's local method, Sauvola's method, and Gato's adaptive method) were investigated using FM, ps-FM, PSNR, and DRD evaluation metrics. Sauvola's method outperforms all the other methods on all the metrics.In the document image skew estimation task, Hough transform based method was investigated using evaluation criterion, AED, TOP80, and CE; obtained values equal to 0.3115, 0.058, and 76.00, respectively. In the page layout analysis task, the performance of Leptonica which is open source C library was investigated and achieved results with high success rate on region and text-line level over a wide variety of page layouts of actual historical Ge'ez manuscripts. For building the recognition model, LSTM based Tesseract OCR engine is used. Due to a difficulty to prepare large training data with ground truth from actual historical documents, fine tuning approach (transfer learning) was proposed and applied. A total of 257 text-line images collected from 15 different pages were prepared and able to build a recognition model with character error rate of 2.63%. Overall, the performed experiments with the prototyping approach have produced encouraging results so that a complete OCR system development for historical Ge'ez manuscripts is applicable. As a future work, however, investigation needs to consider incorporating post-processing into the recognition process.

Keywords: *Handwriting recognition; Ge'ez manuscripts; OCR; text-line image; deep learning; pattern recognition*

1. INTRODUCTION

Handwriting is a concatenation of graphical symbols drawn using a hand to represent linguistic constructs for communication and knowledge storage. These graphical marks or writing symbols have deep orthographic relation to the phonology of a spoken language. However, to a machine or computer, handwriting is nothing but a pattern [1]. The patterns can be observed as internal relationships within the pixels of a document image. Therefore, recognition of this pattern is performed in order to read a manuscript by a computer. The process of automatic pattern recognition of characters from an optically scanned document image is known as Optical Character Recognition (OCR) [2]. OCR works by involving the extraction of features and discrimination or classification of these features based on patterns. These patterns are highly based on the nature of the input data.

Based on the writing generation strategy and data processing, the handwritten input data for handwriting recognition can be broadly categorized into two modes, i.e., offline and online [1][2]. The offline handwritten input data for handwriting recognition is a static data and generated from scanned images while the online handwritten input data is dynamic and its generation is based on the movement of pen tips having certain velocity, projection angle, position and locus point [2]. Offline handwriting recognition to historical documents, in particular, is a complex task mainly due to low document quality and various complex page layouts. It can be defined as a task to recognize handwritten text in order to generate a transcript of a given document [3]. Manual transcribing is a laborious job, and requires expertise and a fair amount of time with keen attention. Therefore, there is a growing interest in pattern recognition for automatic handwriting recognition in order to ease this transcript generation task. However, handwriting patterns are complex due to the challenges of their multifold variations. Since the early time, hence, automatic handwriting recognition has become an important research topic in the areas of image and pattern recognition [1]. It has also been a major research problem for several decades and has gained attention in recent times due to the potential value that can be unlocked from extracting the information stored in historical documents.

Various research works of handwriting recognition uses pattern recognition approaches which are known as template matching, statistical, structural and syntactical for feature extraction and classification tasks [3]. More recently, however, deep learning techniques and methods derived from statistical learning theory have been receiving increasing attention in pattern representation. Unlike simple artificial neural networks, deep learning is not only used for the mapping from representation to output but also to learn the representation itself. This approach is known as representation learning [4]. Learned representations often result in much better performance than can be obtained with hand-designed representations. Thus, the powerful automatic feature extraction ability of deep learning reduces the need for a separate handcrafted feature extraction process. The recently released multilingual Tesseract[5] OCR engine by Google, for instance, is also purely implemented using deep learning techniques. Some other successful application areas of deep learning include image classification, object detection, video processing, natural language processing (NLP), and speech recognition [4].

Though Ge'ez was the language of literature in Ethiopia until the middle of the 19th century[5], it is underrepresented in the research areas of document image analysis and recognition.Previous related research works reveals that very few studies i.e., by Yaregal & Bigun [6], Siranesh [7], Shiferaw [8], and Fitehalew[9] attempted to apply OCR to historical Ge'ez manuscripts. However, all of them focused towards character level recognition, i.e. none of them formulated the handwriting recognition problem as a sequence of pattern classification problem on text-line level. One of the advantages of text-line level recognition is that it does not require text-line to word, and word to character segmentations, which is one of the most common reasons for high word or character error rate.In terms of the pre-processing step, none of them applied objective evaluation method which accounts for the performance of the binarization

[5]https://tesseract-ocr.github.io/

and skew estimation tasks. In addition, page layout analysis was not considered. Because it is often not sufficient to simply segment the scanned pages into text and non-text areas to proceed with OCR. Hence a detailed page layout analysis, i.e. page segmentation and region classification is required.

Therefore, in this study, the proposed handwriting recognition system is made based on real-world large scale digitization scenarios. Thus the preferred workflow for designing the system is the one that allows modular and sequential processing of information. Its architecture is comprised of tasks, namely: pre-processing (binarization and skew estimation), page layout analysis (page segmentation and region classification), recognition model, and post-processing.For each task, experimental setup is designed.

2. RELATED WORKS

Extensive study of previous related research works reveals that very few studies i.e., by Yaregal & Bigun [6], Siranesh [7], Shiferaw [8], and Fitehalew [9] attempted to apply OCR to historical Ge'ez manuscripts. In light of the pattern recognition techniques, the studies applied hybrid (structural/syntactical approach), deep Multilayer perceptron (MLP), statistical approach, and deep convolutional neural networks (CNN), respectively.

The first study i.e., by Yaregal & Bigun [6], applied structural and syntactical pattern recognition approach to OCR of handwriting recognition scale to both modern and historical documents. In this work, directional field tensor was proposed as a tool for character segmentation and extracting primitive structural features and their spatial relationships. A special tree structure was used to represent the spatial relationship of the primitive structures and traversed to generate a set of unique sequence of primitive strokes for each character. Then the generated sequence of strokes was matched against a stored knowledge base of primitive strokes for each character. The study attempted to design a generic recognition system that invariably works for different handwriting styles and sizes. However, document image analysis tasks were not considered in the recognition process.

The second study i.e., by Siranesh [7], applied deep learning techniques to OCR on historical Ge'ez manuscripts. The document page was decomposed stage-by-stage into characters using projection profiles and normalized to 30x30 pixels size. Multilayer perceptron (MLP) network with three hidden layers was employed for feature extraction process and Softmax output layer for the classification task. The network was trained over Restricted Boltzmann Machine (RBM) in a greedy layer-wise unsupervised training manner. Finally the whole network was fine tuned in supervise manner using the Softmax function criteria. But, no regularization techniques were applied to overcome the overfitting problem. The scope of the study was limited to 24 basic syllables (consonants) of the Ge'ez script only.

The third study i.e., by Shiferaw [8], applied statistical pattern recognition approach to OCR on historical Ge'ez manuscripts. The document page was decomposed stage-by-stage into characters using projection profiles and character's size was normalized using nearest-neighbor interpolation technique. Following thinning/skeletonization process, extent, connected component analysis and projection profile were applied to extract handcrafted statistical features (produced a total of six features for each character image). Finally,

one-against-all multiclass classification of SVM with RBF kernel function were employed for the classification task. The scope of the study was limited to 202 syllables of the Ge'ez script.

The fourth study i.e., by Fitehalew [9], applied deep learning techniques of convolutional neural network topology to OCR on historical Ge'ez manuscripts. The characters were directly segmented using contour analysis method. Following it, extracted characters were then normalized to 28x28 pixels size. Deep convolutional neural network was employed for feature extraction process and Softmax output layer (28 nodes) for the classification task. The network was trained with Adam optimizer. Early stopping and dropout regularization techniques were applied to overcome the overfitting problem. The scope of the study was limited to 28 essential syllables of the Ge'ez script only.

For a comprehensive list of OCR research works for Amharic and Ge'ez scripts, refer to [10].

3. METHODS AND APPROACHES

Experimental research design with prototyping approach was employed to build the proposed handwriting recognition system since it is very helpful to improve the system through experiment.The proposed system is made based on real-world large scale digitization scenarios. Its architecture is comprised of tasks, namely: pre-processing (binarization and skew estimation), page layout analysis (page segmentation and region classification), recognition model (trained model), and post-processing.

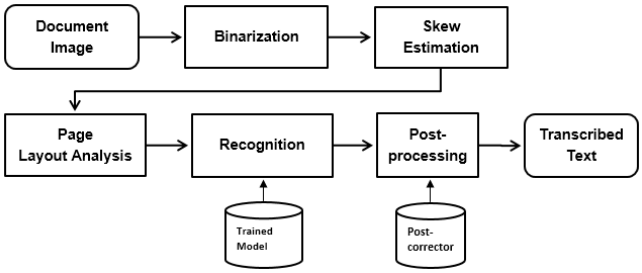


Figure 1: Architecture of the proposed handwriting recognition system

The proposed system accepts scanned document image as an input. Binarization, skew estimation and page layout analysis tasks are performed sequentially prior to the text recognition. All the tasks are equally important and contribute to the final recognition process. Thus we need to perform an objective evaluation of the suitability of each task for the practical handwriting recognition problem.The final stage is post-processing. For each task, experimental setup is designed.

The first experimental setup is designed with a goal mainly to select the best algorithm for the binarization task. Based on their good records in the document image analysis literature, four binarization methods are proposed in this experimental setup. Three of them are the standard approaches known as Otsu's global method, Otsu's local method and Sauvola's method. The remaining one is Gato's adaptive method [11][12][13]. The best method among from them needs to be chosen by experimenting and examining the results from a testing dataset. The testing dataset for the binarization of degraded documents with Ground truth are collected from the DIBCO[6] contest held in 2019. It consists of nine (9) images which

[6] https://users.iit.demokritos.gr/~bgat/DIBCO2009/benchmark/

have representative degradations of a similar problem at hand.The widely used performance evaluation metrics known as F-Measure (FM), pseudo F-measure (ps-FM), Peak Signal-to-Noise Ratio (PSNR), and Distance Reciprocal Distortion (DRD) are used for the evaluation purpose[13][14][15]. These evaluation metrics were also adopted by DIBCO in its latest international Document Image Binarization Contest held in 2019[17].

The second experimental setup is designed to investigate the skew estimation method. The detection and correction of document skew is one of the most important tasks in the document image analysis step of OCR system [18]. Many document page segmentation algorithms are designed to process document images with zero skew. Thus we need to apply skew correction process prior to document page segmentation. Because it affects the handwriting recognition process indirectly. Therefore, skew estimation method known as Hough transform method [19] is selected to process skew detection and correction. Hough transform is a widely known technique in computer vision and image analysis.The quality of the Hough transform method is investigated by experimenting and examining the results over a dataset. The dataset[7] with Ground truth is employed from the DISEC'13 competition which was an international Document Image Skew Estimation Contest (DISEC) organized in the context of ICDAR conference in 2013 [18]. In order to measure the performance of the Hough transform method, three criterions are used: (a) the Average Error Deviation (AED), (b) the average error deviation of the Top 80% (TOP80), and (c) the percentage of Correct Estimation (CE) with the threshold of 0.1^0. The threshold of 0.1^0 was chosen due to the fact that a skew angle greater than this threshold may be visible to a human observer. These performance evaluation criterions were also adopted by DISEC'13 competition organized in the context of ICDAR conference in 2013[18].

The third experimental setup is designed to evaluate the page layout analysis task using a realistic document images and an objective performance analysis system known as Aletheia [20]. A number of distortions frequently visible in digitized historical Ge'ez manuscripts are the major hurdles when building a complete OCR system for mass digitization. Because it is often not sufficient to simply segment the scanned pages into text and non-text areas. The objective of page layout analysis is primarily to carry out page segmentation and region classification, i.e. to group image pixels according to constituent regions or objects[21][22][23]. A detailed page layout analysis that considers: accurate semantic distinction of region types (image, text, paragraph and caption),a reading order that includes all text regions on a page, and accurate detection of text lines, words and glyphs are required. Therefore, this experimental setup aims to investigate the performance of Leptonica[8] which is open source C library for efficient document image analysis. Its performance is investigated using a testing set consists of five document pages of historical Ge'ez manuscripts with various page layouts. Each document page got processed using Leptonica and the results are stored using the PageXML standards and compared against the ground truth created manually using Aletheia tool. As input: the ground truth XML file, the segmentation result XML file and

[7] https://users.iit.demokritos.gr/~alexpap/DISEC13/resources.html
[8] https://github.com/DanBloomberg/leptonica

the black-and-white document image are required. For the evaluation, the ground truth regions are compared to the segmentation result regions. Differences are logged as evaluation errors (merge, split, miss, partial miss, misclassification, false detection and overall error).

The fourth experimental setup is designed for building a recognition model. Mainly due to the introduction of multilingual open source OCR engines (e.g. Tesseract OCR engine by Google), it is now possible to train models in order to recognize even historical documents with excellent accuracies. Hence Tesseract is selected as OCR engine for building the recognition model primarily due to its support to Ethiopic script as well as its open source ethos and popularity with large scale digitization. The latest Tesseract uses Long Short-Term Memory (LSTM) network based models. The challenges at hand can be formulated as optimization problem, which hypothesized to maximize Tesseract's recognition model accuracy over a set of training samples from actual historical Ge'ez manuscripts. Training Tesseract OCR engine involves three main steps: training data preparation, running the training process, and performance evaluation of the newly trained recognition model.

The experiments were performed using HP PRO 3500 Series MT with hardware specification: Intel® Core™ i3-3240 CPU @ 3.40 GHz, 4GB Ram, and x64-based processor. The operating system was Windows 10 Pro. MATLAB version R2020b was used to implement the image processing libraries. MATLAB Runtime version 9.0 (R2015b) was used to run DIBCO evaluation tool.Aletheia Pro version 1.2.4 is employed for the training data generation. It is known for its robustness and script independent in the process of document image analysis. Ubuntu 20.4 was used for running the training process with terminal commands.

4. EXPERIMENTAL RESULTS AND DISCUSSION

The first experimentation was conducted to select the best algorithm for the binarization task. The testing dataset consists of nine (9) degraded images and their associated ground truth were collected from the DIBCO competition held in 2019. The selection of the images in the testing dataset was made so that should contain representative degradation similar with the problem at hand. Evaluation results for each test image (T1, T2 … and T9) with respect to the metrics used for the binarization methods are presented in Table 2. At Table 2, for each encountered measure, the detailed performance of each algorithm is given. The final ranking as shown in Table 1, 'Total Score' was calculated by firstly, sorting the accumulated ranking value for all measures for each test image. The summation of all accumulated ranking values for all test images denote the final 'Total Score' which is shown in Table 1.

Let $R^i(j,m)$ be the rank of the method i concerning the j^{th} image when using the m^{th} measure. Then, for each binarization method i, the Total Score S_i is given by the following Equation:

$$S_i = \sum_{j=1}^{K} \sum_{m=1}^{L} R^i(j,m)$$

where, K is the number of images used in the evaluation (i.e. K =9) and L is the number of the evaluation metrics (i.e. L =4).

Table 1: overall evaluation results and the final ranking of the binarization methods

Method	FM	ps-FM	PSNR	DRD	Total Score	Overall Rank
Otsu_Global	7.64	7.60	8.77	27.79	144	4th
Otsu_Local	71.97	76.02	14.94	9.41	83	3rd
Sauvola	**74.14**	**80.30**	**15.87**	**6.63**	**61**	**1st**
Adaptive	72.84	78.64	15.01	6.69	72	2nd

Note: The best results are shown in bold.

The detailed performance for each binarization methods is also given in Table 2 below. It shows Evaluation results for each test image (T1, T2 … and T9) with respect to the metrics used.

Table 2: Evaluation results of binarization for each test image with respect to the metrics

Metrics	Method	T1	T2	T3	T4	T5	T6	T7	T8	T9
FM	Otsu_Global	14.9006	12.8416	8.1776	9.3087	8.3993	4.8511	1.4718	2.5214	6.2393
	Otsu_Local	44.4327	67.7965	**48.9389**	63.2103	**85.3138**	**92.8829**	**80.8532**	74.2212	90.0449
	Sauvola	53.63	**73.3612**	41.2534	**76.1372**	77.1447	91.3149	80.0514	**79.9096**	**94.4817**
	Adaptive	**56.006**	69.1116	47.2924	76.1213	81.9821	83.3187	79.9856	75.3027	86.4697
ps-FM	Otsu_Global	15.0728	12.6474	8.1534	9.2153	9.8798	5.7331	0	1.6175	6.0947
	Otsu_Local	44.4981	68.5106	49.7442	63.4585	**85.5918**	93.8663	87.2958	**99.9766**	91.1966
	Sauvola	53.8065	**76.7713**	50.5516	78.4874	77.2003	**94.7663**	**92.3783**	99.7318	**99.043**
	Adaptive	**56.2669**	71.631	**52.8831**	**78.8612**	82.3783	85.8218	91.2377	99.7611	88.9277
PSNR	Otsu_Global	4.5985	6.5215	7.3313	6.2235	9.4035	10.323	14.9592	11.3498	8.2283
	Otsu_Local	6.9445	11.3258	11.2705	10.4821	**17.4052**	**21.5395**	19.7167	17.6033	18.1682
	Sauvola	8.5715	**12.8572**	**12.41**	13.4457	15.0225	20.8373	**20.0166**	**18.4707**	**21.2233**
	Adaptive	**9.0067**	11.8798	12.3107	**13.4924**	16.362	17.5653	19.9165	17.7405	16.8086
DRD	Otsu_Global	45.1834	30.6362	50.3437	30.1252	20.7442	20.6821	7.2204	21.9757	23.1546
	Otsu_Local	27.1274	10.2824	20.3962	12.2289	**3.3472**	**1.7144**	3.2743	4.0699	2.2351
	Sauvola	18.2521	**7.2583**	**14.0178**	5.9303	5.9314	1.7531	**2.3823**	**3.4831**	**0.67032**
	Adaptive	**15.8394**	8.6869	15.0362	**5.3471**	3.9665	3.071	2.3915	3.9449	1.9633

After a careful analysis of the binarization evaluation results presented in Table1, it can be easily observed that the best performance is achieved by Sauvola's method. Sauvola's method outperforms all the other methods on all the metrics that were used. Similarly, the second-ranked Gato's Adaptive method is also the second-best using all the metrics that were used. On the other hand, Otsu's global method achieves the worst performance on all the metrics that were used. However, in the previous research works of OCR system for historical Ge'ez manuscripts such as by Siranesh [7], Shiferaw [8] and Fitehalew [9] attempted to incorporate binarization technique based on mainly Otsu's method. But, none of them had applied objective evaluation method.

The experimental result demonstrates or has proven that global thresholding method is NOT sufficient on low quality and degraded historical document images. On other words, local or adaptive thresholding methods have demonstrated good records on low quality and degraded historical document images. Even though Sauvola's method outperforms all the other methods on all the metrics that were used, it requires further investigation to determine its optimal parameter values (i.e. window size and weight). Size of the

testing set has also played a significant role to assess robustness of the method. For instance, the third-ranked Otsu's local method outperforms all other methods on all metrics in a single testing case, T5. However, the performance is not consistent to the rest of all testing cases.

Figure 2: Samples of binarization results by Sauvola's method

On the other experimentation to investigate the quality of the proposed Hough transform method forskew estimation, well-known experimental dataset from ICDAR 2013 Document Image Skew Estimation Contest (DISEC'13) is employed. The experimental dataset consists of 20 unique images with a total of 200 images. For each unique image 10 rotated samples are generated. These rotation angles are randomly selected from the limited range (-15^0, $+15^0$). The performance evaluation criterion AED, TOP80, and CE were used and obtained values equal to 0.3115, 0.058, and 76.00, respectively.

In Table 3 below, comparison is made with other skew estimation methods performed over the same dataset. Huang *et al.*[24] reported the performance of Projection profile and the Standard Hough Transform (SHT) methods on the same dataset. The SHT method was applied on extracted contours of objects using Canny filter. As shown in Table 3, the proposed Hough transform method outperforms all the rest methods on TOP80 criterion with high margin and achieves relatively nearly the same performance with the Projection profile method on the AED criterion.

Table 3: Evaluation results and comparison of skew estimation methods

Method	AED (^{0})	TOP80 (^{0})	CE (%)
Projection profile	**0.290**	0.195	-
SHT	6.120	4.374	-
Hough transform	0.3115	**0.058**	76.00

Note: The best results are shown in bold.

As it shown in Table 3, the proposed Hough transform method has high precision, i.e. 76% correct estimation. The method was succeeded to perform under the well accepted threshold of 0.1^0 in the TOP80 criterion. This demonstrates that the method behaves accurately in its desired operation status as well as it shows the method is robust and treats most of the cases in the same way. However, the method fails to perform under the threshold of 0.1^0 in an AED criterion. But this can be improved with some optimization techniques which require further investigation.

Though previous research works of OCR system for historical Ge'ez manuscripts such as by Siranesh [7] and Fitehalew[9] attempted to incorporate skew estimation technique based on Projection profile and Hough transform respectively, none of them had applied objective evaluation method.

On the other experimentation, the effectiveness of Leptonica which is an open source C library for image analysis was investigated. Leptonica library can be used in conjunction with Tesseract for OCR. In this experimentation, Leptonica's page segmentation and region classification performance over a testing set is investigated using Aletheia tool. The testing set consists of five (5) document images of realistic historical Ge'ez manuscripts with a wide variety of complex layouts and physical formats. The following figure shows the page layout description of each document image in the testing set.

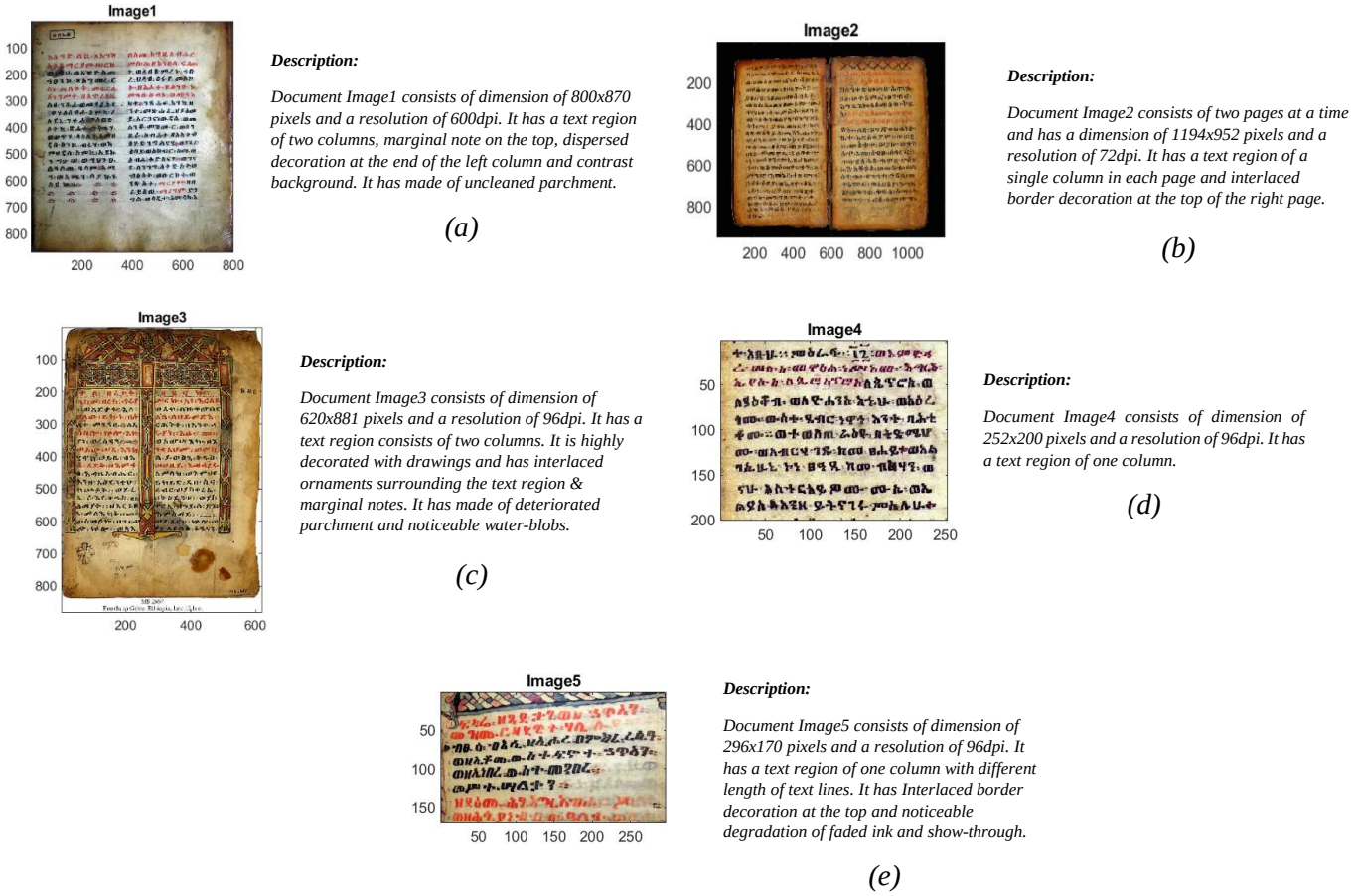
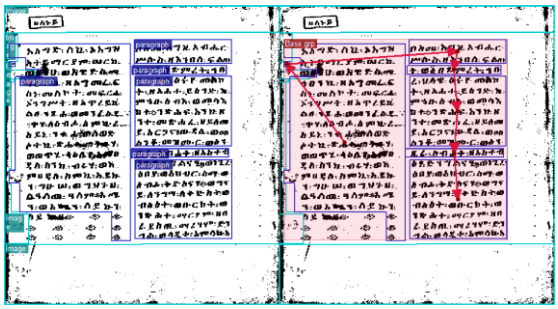


Figure 3: Page layout description of each document image in the testing set

Before performing the page layout analysis, each document image in the testing set were pre-processed. Savoula's method of binarization and the Hough transform based method of skew detection and correction were performed prior to the page layout analysis task. Both methods are selected based on their effectiveness in the previous experiments, described above. The detected skew angle of each document image in the testing set were -0.2, -0.65, 2.32, 12.65 and 6.7 degrees correspondingly.

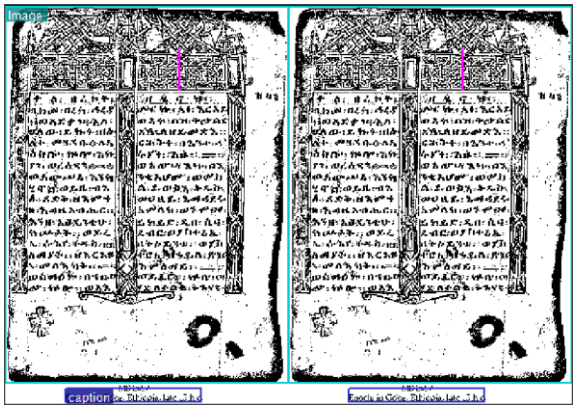
Figure 4 below shows the semantic distinction of region types (image, text, paragraph and caption) and reading order that includes all text regions on the page. When it comes to the region classification, Leptonica performed accurately in Image1, Image2 and Image4 relatively. The worst region classification is manifested in Image3. The whole page is classified as an image region and a short length separator, though it has a text region consists of two columns. The possible reason the text region is not accurately classified is that it is highly decorated with drawings and has interlaced ornaments surrounding the text

region. However, the caption is accurately identified. Similarly, most of the text region in Image5 has been misclassified as caption instead of paragraph. One of the possible reason this happens due to the surrounding of the portion of the text region with interlaced decoration on the top.

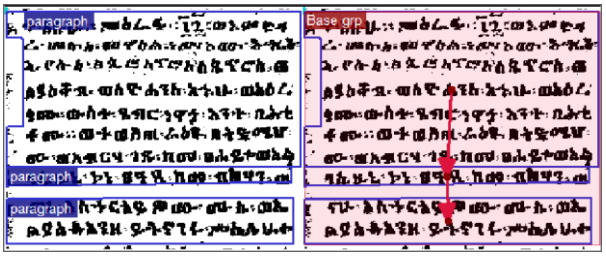


(a) In Image1, Leptonica accurately classified the text region. All the rest part including the marginal note on the top are classified as image zones. The arrows on the right side shows the reading order.

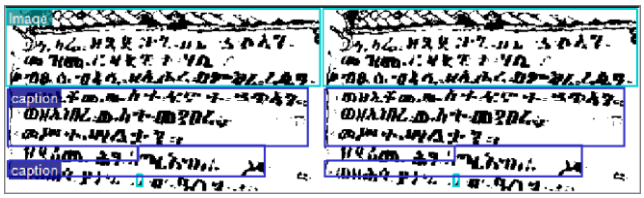
(b) In Image2, Leptonica accurately classified the text region. All the rest part is classified as image zone. The arrows on the right side shows the reading order.



(c) The whole Image3 is misclassified as an image region and a short length separator, though it has a text region consists of two columns.



(d) In Image4, Leptonica accurately classified the text region. The arrows on the right side shows the reading order.



(e) In Image5, most of the text region has been misclassified as caption instead of paragraph.

Figure 4: Region classification and reading order of each document image in the testing set

The other important issue that must be noted in page layout analysis task is text-line detection in the text region. When it comes to complex layout and degraded historical documents, text-line detection is still challenging task. In this experimentation, however, Leptonica has performed text-line detection in acceptable manner. Figure5 shows how Leptonica has performed text-line detection in the tested document image.

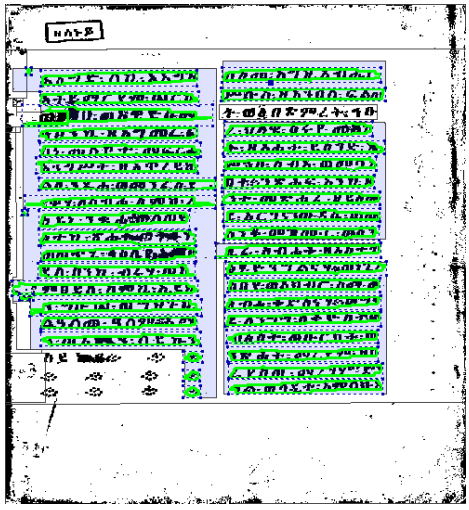


Figure 5: Sample of text-line detection in Image1

In addition to the text-line detection, further investigation of word and glyph detection were also performed. After a careful analysis of the preliminary investigation of the word and glyph detection, however, it has been observed that Leptonica has failed to perform at acceptable success rate in all the tested document images.

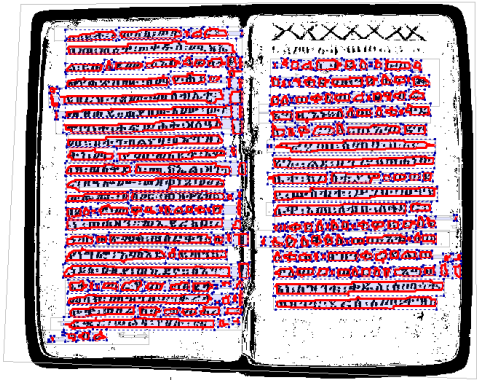


Figure 6: Sample of word detection in Image2

In addition to the above qualitative description of the experimental results, objective evaluation method was also used in the page layout analysis. In order to use the objective evaluation method, each document page got processed using Leptonica and the results were stored using the PageXML standards and compared against the ground truth created manually using Aletheia tool. The ground truth XML file, the segmentation result XML file and the black-and-white document image were used as input for the evaluation. The ground truth regions were compared to the segmentation result regions. Differences were logged as evaluation errors (merge, split, miss, partial miss, misclassification, and false detection in terms of success rate). The default settings including weights were considered. The evaluate levels considered were regions, text-lines and groups. 'Plain.evx' option was used as evaluation profile. Finally, success rates of the following evaluation metrics are recorded.

Table 4: Page layout evaluation results of success rate for each document images

Evaluation Metrics	Image1		Image2		Image3		Image4		Image5	
	Region level	Text line level	Region level	Text line level	Region level	Text line level	Region level	Text line level	Region level	Text line level
Merge	73.89%	97.44%	100.00 %	26.33%	29.32%	24.37%	90.19%	99.22%	52.42%	35.19%
Split	49.97%	98.20%	40.60%	45.08%	99.72%	100.00 %	52.86%	99.89%	39.34%	85.72%
Miss	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	99.17%
Partial Miss	99.97%	96.89%	100.00 %	100.00 %	99.79%	99.28%	98.74%	99.92%	100.00 %	99.89%
Misclassification	96.82%		62.34%		63.97%		96.67%		41.31%	
False Detection	100.00 %	33.94%	100.00 %	99.90%	100.00 %	98.60%	100.00 %	99.94%	100.00 %	100.00 %
Overall success rate:										
Arithmetic mean	63.06%	66.27%	79.57%	51.82%	75.70%	59.06%	**88.19%**	**99.79%**	68.73%	66.64%
Harmonic mean	55.51%	50.63%	69.06%	39.29%	58.11%	37.43%	**82.45%**	**99.79%**	57.77%	52.62%

A careful analysis of the results of objective evaluation in Table 4 also ratifies the in-depth observational analysis described previously. For instance, in the case of Image4, Leptonica has performed region classification accurately at region and text-line level as shown in Figure 4 (d) which is also manifested with the highest success rate as shown in Table 4.

The final experimentation was conducted for building a recognition model.One of the challenges to apply deep learning technique is its requirement of huge amount of training data. In order to cope up with this challenge, however, there are multiple options. One of the multiple options for training is known as fine tuning. Instead of training from the scratch, fine tuning approach (transfer learning) enables starting with an existing base model and then train on a specific additional data. Due to a difficulty to prepare large training data with ground truth from actual historical documents, fine tuning approach is getting popular nowadays. To address the problem, in this study, a similar approach is proposed and applied in the context of historical Ge'ez manuscripts.

The LSTM based Tesseract training process requires ground truth on text-line level. Hence a training data is prepared from actual historical Ge'ez manuscripts. The State of Emergency and curfew imposed due to the Corona Virus pandemic had made it unable to collectthe manuscripts. Hence, the sample data source of the digitized historical Ge'ez manuscripts were obtained from Shiferaw [8] who had collected

them for a purpose of another study few years ago. The manuscripts were collected from parishes and monasteries found in the North Gonder, Ethiopia. The collected manuscripts were digitized using two methods; CamSacnner software that is installed in the Samsung Note5 mobile device with 16MP camera and Iphone 4s with 8MP camera. For the purpose of the training data, a total of 257 text-line images collected from 15 different pages are prepared using Aletheia tool. The corresponding ground truth is UTF-8 encoded text-line with text files. The prepared training data and the corresponding ground truth is available here at http://dx.doi.org/10.13140/RG.2.2.29875.96802.

Figure 7: Sample of scanned pages

The Tesseract training process is made via a collection of command line tools and Linux shell scripts after installing all the required pre-requisite libraries. The procedure is accomplished using OCR-D on Ubuntu 20.04. The procedure is composed of the following steps:

1. Download base model: 'amh.traineddata' data file of tessdata-best type from https://github.com/tesseract-ocr/tessdata_best/blob/master/amh.traineddata.
2. Download OCR-D open source library from https://github.com/kevinbicycle/ocrd-train
3. Install the C++ compiler: $ sudo apt install g++
4. Install the latest Tesseract: $ sudo apt-get install tesseract-ocr
5. Provide training images and their ground truth text labels to OCR-D. Provide images with .tiff format and their labels in .gt.txt extension. Images should contain only one line of text image, and their ground truth labels also should contain only one line of text. We need to place all these files in data/ground-truth folder inside OCR-D.
6. Install Ubuntu make: $ sudo apt install make
7. Open terminal in the OCR-D folder
8. Execute the following command to start the training: $ sudo make training

After the training is finished, the output on the terminal looks like the following.

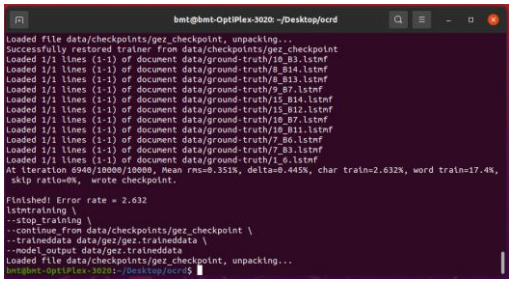


Figure 8: The output on the terminal that shows the lstmtraining

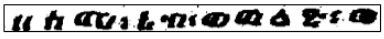
The following default parameter values were used during the training process:

MAX_ITERATIONS	Max iterations. Default: 10000
LEARNING_RATE	Learning rate. Default: 0.0001 with START_MODEL, otherwise 0.002
PSM	Page segmentation mode. Default: 6
RANDOM_SEED	Random seed for shuffling of the training data. Default: 0
RATIO_TRAIN	Ratio of train / eval training data. Default: 0.90
TARGET_ERROR_RATE	Stop training if the character error rate (CER in percent) gets below this value. Default: 0.01

The output of the training is a 'gez.traineddata' file. By convention, Tesseract models use (lowercase) three-letter codes defined in ISO 639 with additional information separated by underscore. Hence the newly trained recognition model for the Ge'ez language named as 'gez' as per the convention.

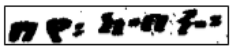
The experimental result shows that the character error rate of the newly trained recognition model, i.e. the 'gez.traineddata' is 2.63%.To assess the impact of the training, OCR results with and without training werecompared. Here, without training means using the base model, i.e. 'amh.traineddata'. In order to use both recognition models, we need to copy them to tessdata directory. On other words, 'amh.traineddata' and 'gez.traineddata' files need to be copied to the data folder of the Tesseract instance that will be used to perform OCR. The following text line images from the evaluation list are randomly selected and examined to compare the OCR results of both models.

(a)



Ground truth ⟶ በስመ፡አብ፡ወወልድ፡ወ
Recognition using the base model ⟶ ያጥፒየወር፤ደ ወ፡ወ፡ናመመራኅልውጅው፡፡ና፡መሞሯ
Recognition using the new model ⟶ ዘስለመ፡አብ፡ወወሰይሰድ፡ወ

(b)



Ground truth ⟶ በየ፡ክብሩ፡
Recognition using the base model ⟶ ጋዋፒ፡ጀቁወም።
Recognition using the new model ⟶ በየ፡አብድያ፡

(c)

Ground truth ⟶ ር፡ኢወድቁ።
Recognition using the base model ⟶ ር።።ጴወ ጅ.።
Recognition using the new model ⟶ ር፡ኢወ ።ናቁ።

The comparison is also made on degraded low quality document image as follows.

(d)

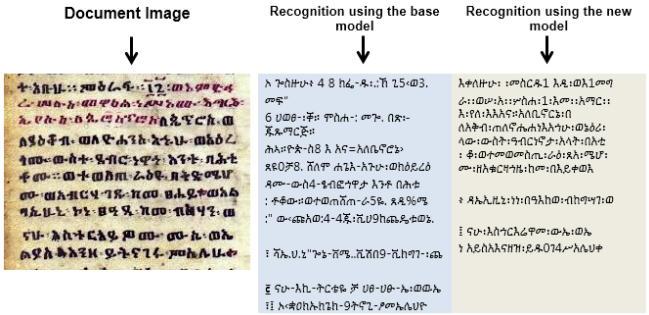


Figure 9: Comparison of OCR performance by the base and new models

Overall, the method demonstrates that the character error rate of Tesseract's base model 'amh.traineddata' can be decreased by continuing training using very few training data in the context of

historical Ge'ez manuscripts. From a qualitative point of view, the change in the error rate is substantial. This implies that a larger training sample can make a better accuracy.

Previous related research works i.e., by Yaregal & Bigun [6], Siranesh [7], Shiferaw [8], and Fitehalew [9]attempted to apply OCR to historical Ge'ez manuscripts with a focus towards character level recognition, not text-line level recognition. Therefore, we were not able to compare results with their work.

5. CONCLUSION

The performed experiments have produced encouraging results, which ensure the applicability of the proposed investigation. The study has empirically showed the performance of a set of algorithms and techniques in the document image analysis and recognition on building the handwriting recognition system for historical Ge'ez manuscripts. The performed experiments with the prototyping approach have produced encouraging results so that integrated Tesseract OCR engine and Leptonica library can be an ideal solution for a complete OCR system development for historical Ge'ez manuscripts. On other words, Tesseract which is fully implemented in the C++ programming language can be integrated with the Leptonica library which is implemented in the C language.

The major weakness of the studyis optimization. Primarily in the task of binarization, Sauvola's method was not optimized. Similarly, the Hough method of skew detection and correction was not optimized. The trained model also was not optimized with large dataset. Therefore, further optimization technique with large training sample is required. Moreover, in all the tasks, Ground truth is the basis for objective performance evaluation methods. Accurate Ground truth, however, is crucial for the evaluation.

6. FUTURE WORKS

Still there is a long way to go in investigating handwriting recognition problem for historical documents. Extension works that need further consideration in the future to advance the current works in the handwriting recognition problem for historical Ge'ez manuscripts are explained as follows.

Handwriting is a form of language representation. Usually, the characters to be recognized are not random sequences, but they are meaningful words. Similarly, sequences of words normally convey a meaning, and form sentences that are syntactically, grammatically, and semantically coherent. On other words, the final transcriptions are required to form sequences of dictionary words. As a future work, investigation needs to consider incorporating Natural Language Processing (NLP) or statistical language model into the recognition process.

Though the state-of-the-art OCR engines have achieved reasonably high accuracy for printed documents, they may not perform well on degraded historical documents. Therefore, it is a wise decision to investigate on another alternative technique for the handwriting recognition problem of historical documents. Thus the other possible extension work is keyword spotting. The aim of the keyword spotting is to identify a particular set of handwritten words within the historical document image. The discriminative power of neural networks is interesting for keyword spotting because they are able to concentrate on identifying and distinguishing raw pixels of the keywords, while ignoring the rest.

REFERENCES

[1] C. Adak, "A Study on Automated Handwriting Understanding," PhD thesis, University of Technology Sydney. Retrieved from https://opus.lib.uts.edu.au/bitstream/10453/134139/2/02whole.pdf , 2019.

[2] R. Plamondon and S. N. Srihari, "On-line and Off-line Handwriting Recognition: A Comprehensive Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vols. 22, No.1,2000.

[3] A. Fischer, "Handwriting Recognition in Historical Documents," PhD thesis, Universität Bern, 2012.

[4] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016.

[5] A. Worku and S. Fuchs, "Handwritten Amharic Bank Check Recognition Using Hidden Markov Random Field," in *Computer Vision and Pattern Recognition Workshop (CVPRW'03)*, 2003.

[6] A. Yaregal and J. Bigun, "Writer-independent Offline Recognition of Handwritten Ethiopic Characters," in *ICFHR/2008*, 2008.

[7] G. Siranesh, "Ancient Ethiopic Manuscript Recognition Using Deep Learning Artificial Neural Networks,"Master's thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2016.

[8] T. Shiferaw, "Optical Character Recognition For Ge'ez Scripts," Master's thesis, University of Gonder, Gonder, Ethiopia, 2017.

[9] A. Fitehalew, "Ancient Geez Script Recognition Using Deep Convolutional Neural Network," Master's thesis, Near East University, 2019.

[10] G. Mesfin, "Offline Handwritten Text Recognition of Historical Ge'ez Manuscripts Using Deep Learning Techniques," Master's thesis, DOI: http://dx.doi.org/10.13140/RG.2.2.33193.72800, Jimma Institute of Technology, Jimma University, Jimma, Ethiopia, 2021.

[11] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," in *IEEE Transactions on Systems, Man, and Cybernetics*, 1979.

[12] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, no. 2, pp. 225-236, 2000.

[13] B. Gatos, I. Pratikakis and S. Peranto, "An Adaptive Binarization Technique for Low Quality Historical Documents," in *DAS 2004: International Workshop on Document Analysis Systems VI*, 2004.

[14] B. Gatos, K. Ntirogiannis and I. Pratikakis, "ICDAR2009 Document Image Binarization Contest (DIBCO 2009)," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2009.

[15] K. Ntirogiannis, B. Gatos and I. Pratikakis, "Performance evaluation methodology for historical document image binarization," *Trans Image Process*, vol. 22, no. 2, p. 595–609, 2013.

[16] H. Lu, A. Kot and Y. Shi, "Distance-reciprocal distortion measure for binary document images," *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 228 - 231, 2004.

[17] I. Pratikakis, K. Zagoris, X. Karagiannis, L. Tsochatzid and T. Mondal, "ICDAR Marthot-Santaniello - Competition on Document Image Binarization (DIBCO 2019)," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2019.

[18] A. Papandreou, B. Gatos, G. Louloudis and N. Stamatopoulos, "ICDAR 2013 Document Image Skew Estimation Contest (DISEC 2013)," in *2013 12th International Conference on Document Analysis and Recognition*, 2013.

[19] R. Duda and P. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, 1972.

[20] PRImA Research Lab, "Aletheia User Guide," 2019.

[21] L. O'Gorman and R. Kasturi, Document Image Analysis, IEEE Computer Society Executive Briefings, 1997.

[22] K. Khurshid, "Analysis and Retrieval of Historical Document Images," PhD thesis, Université Paris Descartes, 2009.

[23] M. Shafii, "Optical Character Recognition of Printed Persian/Arabic Documents," PhD thesis, University of Windsor, 2014.

[24] Huang et al., "An Efficient Document Skew Detection Method Using Probability Model and Q Test," *Electronics*, vol. 9, no. 55, 2019.

Deep Neural Network for Non-linear Initial Value Problems

Tamirat Temesgen Dufera

Adama Science and Technology University, Adama, Ethiopia
E-mail: tamirat.temesgen@astu.edu.et

ABSTRACT

This paper is aimed at applying deep artificial neural networks for solving system of initial value problems. We developed an algorithm and implemented using python code. We conducted different experiments for selecting better neural architecture. For the learning of the neural network, we utilized the adaptive moment minimization method. Finally, we compare the method with one of the traditional numerical methods-Runge-Kutta order four. We have shown that, the artificial neural network could provide better accuracy for smaller numbers of grid points.

Keywords: *Deep Neural Network; Algorithm; Systems of ordinary differential equation*

1. INTRODUCTION

Deep neural network (DNN) has obtained great attention for solving engineering problems. System of initial value problems (IVPs) that can model various physical phenomena could utilize the advantages of using the method. Though there are well established traditional numerical methods for solving systems of IVPs, they have their own advantages and disadvantages in-terms of accuracy, stability, convergence, computation time, etc. One of the well known method is the fourth order Runge-Kutta method (RK4). It is among the finite difference methods well suited for non-stiff problems.

Artificial neural network (ANN) is an alternative method known to the sci- entific community since 1940s. The beginning of ANN is often attributed to the research article by McCulloch & Pitts (1943). It was less popular due to the capacity of computational machines. The recent development and progresses in the area is attributed to the exponential improvement in the computing capacity of machines both in storing data and processing speed (Basheer & Hajmeer, 2000). "An artificial neural network is an information processing system that has certain performance characteristics in common with biological neural net works" (Yadav et al., 2015). The network imitates the work of biological human brain (Basheer & Hajmeer, 2000). The structure of the architecture constitutes layers: input, hidden and output. Each layer have neurons or units. The name deep neural network (DNN) is used when the structure has more than one hidden layers (Schmidhuber, 2015; Goodfellow et al., 2016).

2. RELATED WORKS

2.1. Deep Neural Networks

The DNN method has contributed a lot to the progress in artificial intelligence specifically in computer vision, image processing, pattern recognition and Cybersecurity (Dong et al., 2021; Dixit & Silakari, 2021; Minaee et al., 2021). The performance is due to features are learned rather than handcrafted, the deep layers are able to capture more variances (Bruna & Dec, 2018).

Some of the challenging issues related to DNN are, stability, robustness, provability and adversarial perturbation which are discussed in Zheng et al. (2016); Haber & Ruthotto (2017); Malladi & Sharapov

(2018); Zheng & Hong (2018) and Szegedy et al. (2013). The optimization problems arising form learning the DNN also need special consideration which are presented in Nouiehed & Razaviyayn (2018) and Yun et al. (2018).

Moreover, the search and selection of an optimal neural architecture is difficult task (Elsken et al., 2019). Some widely implemented deep learning architectures-autoencoder, convolutional network, deep belief network and restricted Boltzmann machine were presented in Liu et al. (2017). A broder survey of advance in convolutional neural network can be found in Gu et al. (2018). More related works and recent advances in application of deep neural networks such as in Cybersecurity, image segmentation, background subtraction and self- supervised image recognition, are presented in Yi et al. (2016); Dong et al. (2021); Dixit & Silakari (2021); Minaee et al. (2021); Bouwmans et al. (2019); Ohri & Kumar (2021) .

2.2.Works related to solving differential equations

Nowadays, researchers are applying ANN for solving differential equations. Some of the advantages of using ANN over the traditional numerical methods are: the solutions obtained by ANN are differentiable, and closed analytic form, the method could handle complex differential equations and helps to overcome the repetition of iteration (Chakraverty & Mall, 2017).

Lee & Kang (1990) implemented neural algorithm for solving differential equations. They have used the method for minimization purpose where development of highly parallel algorithms for solving the difference equations required. Meade & Fernandez (1994) implemented a feedforward neural networks to approximate the solution of linear ordinary differential equation (ODE). They have used the hard limit activation function to construct direct and non-iterative feedforward neural network. The author implemented the method on three layers, input layer, a hidden layer and output layer. Simple first and second order ordinary differential equation were considered for testing the method. Lagaris et al. (1998) used ANN for solving ordinary and partial differential equations. For solving initial and boundary value problems, they used trail solution satisfying the given conditions. Then, network were trained to satisfy the differential equations. The results were compared with well established numerical method–finite element. The authors obtained accurate and differentiable solution in a closed analytic form.

Partial differential equation with initial and boundary condition were solved using neural network (Aarts & Van Der Veer, 2001). The architecture of the network were, multiple input units, single output unit and single hidden layer feedforward with a linear output layer with no bias. Evolutionary algorithms were implemented for the cost minimization. The authors tested the method on problems from physics and geological process.

For solving ODE using ANN, unsupervised kernel mean square algorithm were used in Sadoghi Yazdi et al. (2011). Trial solution similar to the authors in Lagaris et al. (1998), were implemented to obtain accurate results. Nasci- mento et al. (2020) presented the direct implementation of integration of ODE through recurrent neural networks.

Berg & Nystrom (2018) implemented deep feedforward ANN to approximate solution of partial differential equations in complex geometries. They solved problems that could not be addressed or difficult by the traditional method. They did comparison between shallow versus deep networks. More recent

development and applications of ANN in partial differential equations can be found in Berg & Nystrom (2019); Wang et al. (2019); Raissi et al. (2019); Rackauckas et al. (2020).

The application of ANN were also extended to the computation of integral equations. Asady et al. (2014), introduced an efficient application of ANN for approximating solution of linear two-dimensional Fredholm integral equation of the second kind. They have found remarkable accuracy and proposed extension to the case of more general integral equations.

For the implementation of ANN, clear and reproducible algorithm with implementation needs great attention. An efficient neural network architecture has to be investigated corresponding to system of IVPs. We need to look at the effects of numbers of hidden layers in the network model as well as the numbers of neurons in the layers on accuracy, speed and performance of the model in general. We need to investigate and propose the best selection of activation function. Addressing the issue of minimization method for the cost function is also crucial.

In this paper, we present a vectorized algorithm for solving system of IVPs using DNN. We implement the algorithm in python and perform experimental simulations to look at the effects of different neural architecture on the performance of the model. Moreover, we observe the advantage of using the ANN over the traditional methods. Specifically, we consider the fourth order Runge-Kutta finite difference method.

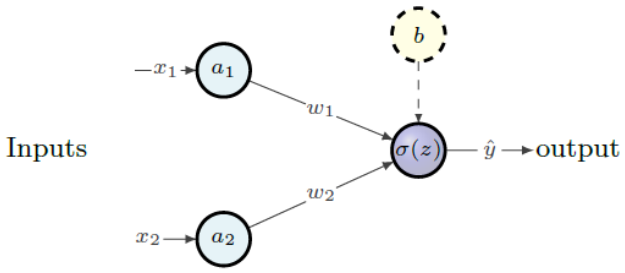
The paper is structured as follows: first we briefly present the neural net- work and remind our reader the general formulation of system of initial value problems. Next we setup the general form of DNN and its application for IVPs. Moreover, we perform different experiment using python code. At the end we implement the algorithm and compare the result with the analytical solution and with numerical solution obtained using Runge-Kutta method.

3. DEEP NEURAL NETWORK

"A neural network is a parallel information-processing system that has cer- tain characteristics in common with certain brain functions. It is composed of neurons and synaptic weights and performs complex computations through a learning process" (Freeman & Skapura, 1991; Goodfellow et al., 2016). Neural networks are a series of algorithms that mimic the operations of a human brain to recognize relations between vast amounts of data.

3.1. A Simple Network

The following diagram shows a simple neural network, with two inputs and one output.



As it is well known, in machine learning algorithms in general, x_1, x_2 represents features for a given sample $\boldsymbol{x}$. The $\hat{y}$ variable represent the network output approximating the target value y.

Given a set of training data $\{(x_1^{(1)}, x_2^{(1)}), (x_1^{(2)}, x_2^{(2)}), \ldots..(x_1^{(m)}, x_2^{(m)})\}$ and corresponding target values $\{y^{(1)}, y^{(2)}, \ldots.., y^{(m)}\}$, the algorithm search for the optimal weights (w_1, w_2) and bias (b) such that the network out put is as close as the target value. For instance, as it is indicated in the diagram, for the sample (x_1, x_2) the process goes as follows:

$$\begin{aligned} a_1 &= x_1, \quad a_2 = x_2, \\ z &= w_1 a_1 + w_2 a_2 + b, \\ \hat{y} &= \sigma(z). \end{aligned} \tag{1}$$

One can rewrite (1) in matrix form as;

$$z = W \cdot X + b,$$

where, $W = [w_1, w_2]$, and $X = [x_1, x_2]^T$. The function $\sigma(z)$ is called an *ac- tivation function or transfer function*. The purpose of activation functions is to introduce non-linearity into the network. Some of the activation functions are; sigmoid, tan hyperbolic and Rectifiable Linear Unit, see Figure 1. These function are;

$$\begin{aligned} \text{Sigmoind: } \sigma(z) &= \frac{1}{1+e^{-z}}, \\ \text{Tanh: } \sigma(z) &= \frac{e^z - e^{-z}}{e^z + e^{-z}}, \\ \text{ReLU: } \sigma(z) &= \max(z, 0). \end{aligned}$$

If the activation function is identity, the network model will become the well es-tablished ordinary linear regression model. Notethatifwehave $X = [x_1, x_2, \ldots\ldots, x_m]$ inputs, then

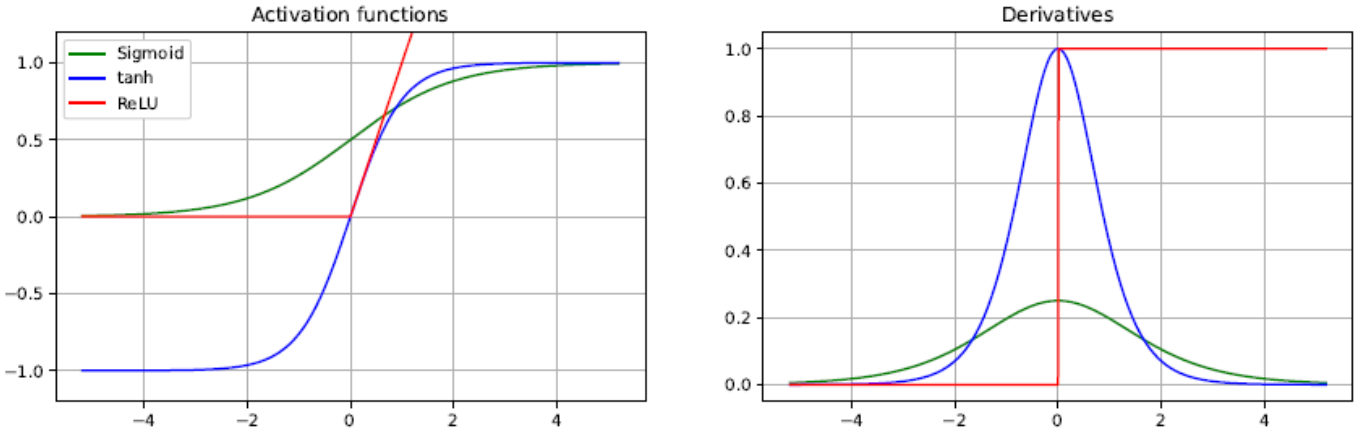


Figure 1: Common activation functions and their derivatives

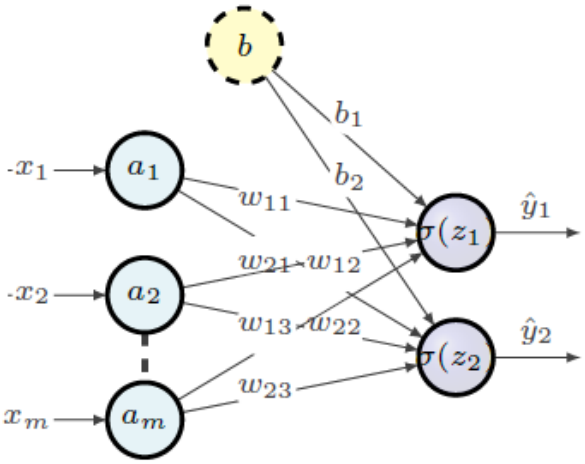
$$z = \sum_{j=1}^{m} x_j w_j + b = W \cdot X + b$$
$$\hat{y} = \sigma(z)$$

To find the optimal parameters, i.e., the weights and the biases, one can apply the objective function given by the L^2 norm.

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} |\hat{y}^{(i)} - y^{(i)}|^2 = \|\hat{y} - y\|^2 \qquad (2)$$

3.2. Multi Output Network

The same discussion can be made with slight modification; the following network is for training data with m features and two outputs $\hat{y}_1$ and $\hat{y}_2$. As can be seen form the following diagram, the networks are densely connected with corresponding learning weights and biases.



Similar to the case of one output network, for each training sample $\boldsymbol{x} = [x_1, x_2, \ldots, x_m]$, the network outputs $\hat{y}_1$ and $\hat{y}_2$ are obtained as follow;

$$z_1 = \sum_{j=1}^{m} x_j w_{j,1} + b_1,$$

$$z_2 = \sum_{j=1}^{m} x_j w_{j,2} + b_2,$$

$$\hat{y}_1 = \sigma(z_1), \quad \hat{y}_2 = \sigma(z_2).$$

For the multiple network output say $[\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_i, \ldots, \hat{y}_n]$, we have;

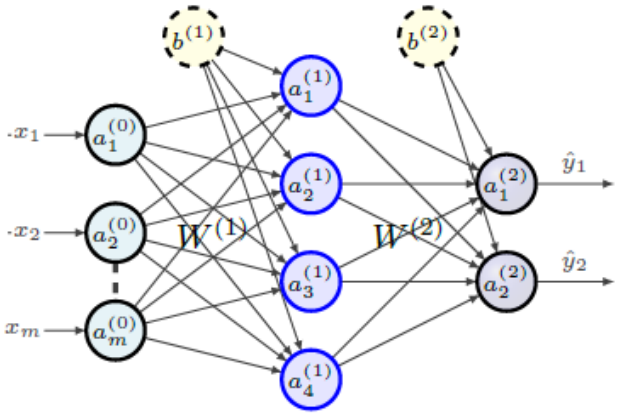
$$z_i = \sum_{j=1}^{m} x_j w_{j,i} + b_i,$$

$$\hat{y}_i = \sigma(z_i).$$

3.3. Single Layer Neural Network

For a data with non-linear relation, adding one more layer will significantly improve the performance of the model (Schmidhuber, 2015). The following schematic diagram illustrates an architecture of a single layer (one hidden layer) neural network.

For this particular model, the network output $\hat{y}$ is given by the following feed forward propagation. Given a sample or a training data X, from input to hidden layer;



$$
\begin{aligned}
X &= \mathbf{a}^{(0)}, \\
\mathbf{z}^{(1)} &= W^{(1)}\mathbf{a}^{(0)} + \mathbf{b}^{(1)}, \\
\mathbf{a}^{(1)} &= \sigma^{(1)}(\mathbf{z}^{(1)}).
\end{aligned}
$$

From hidden layer to output layer;

$$
\begin{aligned}
\mathbf{z}^{(2)} &= W^{(2)}\mathbf{a}^{(1)} + \mathbf{b}^{(2)}, \\
\mathbf{a}^{(2)} &= \sigma^{(2)}(\mathbf{z}^{(2)}), \\
\hat{\mathbf{y}} &= \mathbf{a}^{(2)}.
\end{aligned}
$$

Rewriting the above we get a sequence of composition of functions;

$$\hat{\mathbf{y}} = \sigma^{(2)}(W^{(2)}\sigma^{(1)}(W^{(1)}\mathbf{a}^{(0)} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)}). \qquad (3)$$

Based on the problem at hand, one can add more layer called hidden layers for better performance. A neural network with more than one hidden layer is called deep neural network.

4. DEEP NEURAL NETWORK FOR SYSTEM INITIAL VALUE PROBLEMS

4.1. Initial Value Problems

The general form of *n* system of initial value problems is given by,

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(t, \mathbf{u}) \quad t \in [0, t_f], \qquad (4)$$

$$\mathbf{u}(0) = \mathbf{u}_0, \qquad (5)$$

where $\mathbf{u} = [u_1, u_2, \ldots, u_n]^T$ is the unknown having dimension of $n \times 1$, and

$$\mathbf{f}(t, \mathbf{u}) = \begin{bmatrix} f_1(t, u_1, u_2, \ldots, u_n) \\ f_2(t, u_1, u_2, \ldots, u_n) \\ \vdots \\ f_n(t, u_1, u_2, \ldots, u_n) \end{bmatrix}$$

is given vector valued function having dimension of n -1. The uniqueness and existence of the solutions to the initial value problem is well established theory (Coddington & Levinson, 1955).

4.2. Deep neural network model set up

We consider a dense network of L layers indicated in Figure 2.

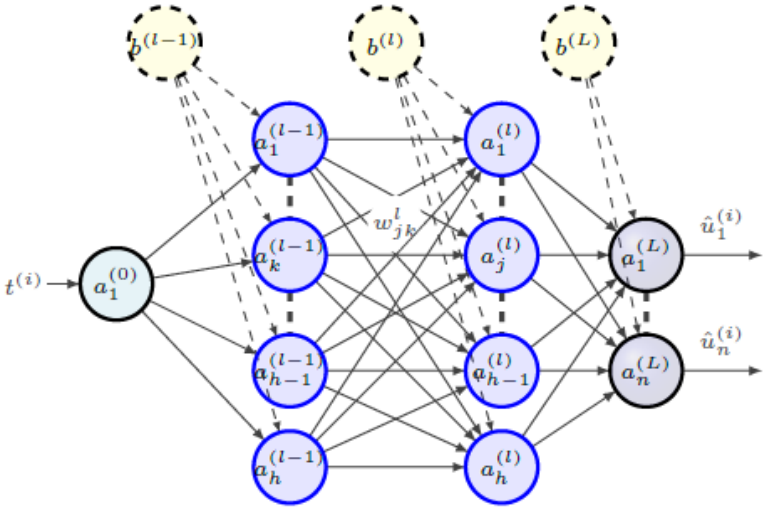


Figure 2: The schematic diagram of deep ANN

The network contains one neuron in the input layer corresponding to the in dependent variable for the system of IVP. The output layer contains n neurons corresponding to the unknown variables. For training the model, we take m sample points from the domain $[0, t_f]$, and form a matrix $X = [t^{(1)}, \ldots, t^{(m)}] \in \mathrm{R}^{1\times m}$. Here $t^{(i)} \in [0, t_f]$ R is the i^{th} sample point or training example. More- over, we denote by $N_k(t^{(i)}, P_k)$ the output of the k^{th} unknown corresponding to the i^{th} sample point, where P_k stands for the corresponding parameters, the weights and the bias.

Following the references, see eg., Lagaris et al. (1998) and Malek & Shekari Beidokhti (2006), for each $t \in [0, t_f]$ we set a trial solution given by,

$$\hat{u}_j(t, P_j) = u_{0j} + tN_j(t, P_j), \quad j = 1, \ldots, n. \tag{6}$$

The trial solution in equation (6), satisfies the initial conditions (5). We train the network in such a way that the total cost function given by,

$$J = \sum_{i=1}^{m}\sum_{j=1}^{n}\left(\frac{d\hat{u}_j}{dt} - f_j\right)^2, \tag{7}$$

Converges to zero. Here, $f_j = f_j(t^{(i)}, \hat{u}_j(t^{(i)}), P_j)$. Note that, the learning process or the training is unsupervised as there are no targeted solutions.

4.3. The Vectorized algorithm

Here we describe algorithm for deep neural network method for solving sys- tem of initial value problems (4)-(5):

Step 1. *Input data*: Take m discrete points from the domain $[0, t_{end}]$ and form a vector $X = [t^{(1)}, t^{(2)}, \ldots,$

$t^{(m)}$] of size $1 \times m$.

Step 2. *Chose the deep neural network structure*: Here we determine the number of layers L, input layer having one units, L 2 hidden layer having h^l units, for each $1 \leq l \leq L-1$ and the output layer having n units which is equal to the number of unknown in the system.

Step 3. *Initialize the parameters*, P_j, $j = 1, \ldots, n$ and $2 \leq l \leq L-1$:

- W^1 has $h^1 \times 1$ dimension,
- W^l has $h^l \times h^{l-1}$ dimension,
- W^L has $n \times h^{L-1}$ dimension,
- $\mathbf{b}^1$ has $h^1 \times 1$ dimension,
- $\mathbf{b}^l$ has $h^l \times 1$ dimension, and
- $\mathbf{b}^L$ has $n \times 1$ dimension.

Step 4.Forward propagation:

- For the input layer start by assigning, $A^0 = X$.
- For the hidden layers, $1 \leq l \leq L-1$,

$$Z^l = W^l A^{l-1} + \mathbf{b}^l,$$
$$A^l = \sigma^l(Z^l),$$

where, σ^l is the activation function corresponding to the l^{th} hidden layer.

- For the output layer,

$$Z^L = W^L A^{L-1} + \mathbf{b}^L,$$
$$A^L = \sigma^L(Z^L),$$
$$\mathbf{N}(X, P_j) = A^L.$$

- Assign the trial solution using the equation (6): To arrive at the trial solution of an unknown function, we need to initialize a corresponding sets of parameters.

Step 5.Compute the cost and its gradient, using equation (7): Calculate gradients with respect to X and with respect to the learning parameters. Here we implement the automatic differentiation (Baydin et al. 2018; Bradbury et al., 2018).

Step 6.Update the parameter using the method of gradient decent or any other best optimization method.

4.4. Minimization of the Loss function

One of the widely used minimization algorithm in machine learning is the gradient decent method. We randomly initialize the parameters and update according to the following rule; for j = 1, 2,

$$P_j^{k+1} = P_j^k - \eta \nabla J(P_j^k),$$

where η is the learning rate and k corresponds to iteration. The gradient decent is well suited for convex function. For non convex function, we are not grantee for global minimum point. Note that, in addition to the simple gradient decent method, currently there are more advanced optimization tools and still is an

active research topics Calin (2020).

The *moment method* is the modification of gradient decent method designed to avoid getting stuck in a local minimum. The updating rule is as follows;

$$P_j^{k+1} = P_j^k + V_j^{k+1},$$
$$V_j^{k+1} = \mu V_j^k - \eta \nabla J(P^k),$$

where $\eta > 0$ is the learning rate and μ is a coefficient between 0 and 1 called the momentum. Here k indicates iteration and V_j is a new parameter (velocity) initialized from zero corresponding to each unknown.

The *Nesterov accelerated Gradient (NAG)* is obtained by modifying the mo- mentum method and the update rule is given as follow,

$$P_j^{k+1} = P_j^k + V_j^{k+1},$$
$$V^{k+1} = \mu V_j^k - \eta \nabla J(P^k + \mu V_j^k).$$

The main difference from the moment method is that, the argument of the gradient is computed at the correlated value $P^k + \mu V_j^k$ instead of computing it at the current position P^k.

The *AdaGrad*, Adaptive Gradient: the update rule have the form;

$$V_j^k = V_j^{k-1} + (\nabla J(P_j^k))^2$$
$$P_j^{k+1} = P_j^k - \frac{\eta}{\sqrt{V_j^k + \epsilon}} \nabla J(P_j^k),$$

where ϵ is small number to avoid division by zero. The method changes the learning rate for the parameters in proportional to the update history. It decays the learning rate.

The *Root Mean Square Propagation*, or RMSProp is family of the gradient decent method having adaptive learning rate, again following Calin (2020), our update rule is as follows;

$$V_j^k = \beta V_j^{k-1} + (1 - \beta)(\nabla J(P_j^k))^2$$
$$P_j^{k+1} = P_j^k - \frac{\eta}{\sqrt{V_j^k + \epsilon}} \nabla J(P_j^k),$$

where $\beta \in (0, 1)$ is the forgetting factor.

Adam, Adaptive Moment, is also an adaptive learning method which com- bines AdaGrad and RMSProp methods. In our case the updating rule have the form;

$$M_j^k = \beta_1 M_j^{k-1} + (1 - \beta_1) \nabla J(P_j^k),$$
$$V_j^k = \beta_2 V_j^{k-1} + (1 - \beta_2)(\nabla J(P_j^k))^2,$$
$$\hat{M}_j^k = \frac{M_k}{1 - \beta_1^k}, \qquad \hat{V}_j^k = \frac{V_k}{1 - \beta_2^k},$$
$$P_j^{k+1} = P_j^k - \frac{\eta}{\sqrt{\hat{V}_j^k + \epsilon}} \hat{M}_j^k,$$

where $\beta_1, \beta_2 \in [0, 1)$, are decay rates for the moment estimates, and we initialize the parameters V_j and M_j to be zero.

5. IMPLEMENTATION AND COMPARISON

In this section we implement the algorithm for solving a known non-linear system of initial value problems. First, we perform simulation for selecting appropriate number of layers and neurons in the layer. Then, we compare with the analytical solution and with a numerical solution obtained using the traditional methods. For this purpose, we consider the following problem found in Lagaris et al. (1998),

$$\frac{dy_1}{dt} = \cos(t) + y_1^2 + y_2 - (1 + t^2 + \sin^2(t)), \qquad (8)$$
$$\frac{dy_2}{dt} = 2t - (1 + t^2)\sin(t) + y_1 y_2,$$

with $t \in [0, 1]$ and $y_1(0) = 0$ and $y_2(0) = 1$. The analytic solutions are $y_1 = \sin(t)$ and $y_2 = 1 + t^2$.

5.1. Experiment on the network

We conducted an experiment on number of neurons in a layer. We looked at the effect of number of neurons on the error function. We took different sizes of neurons in the hidden layer, h = 4, 17, 60, 150, 200, and we plotted the cost function verses the number of iterations for the comparison of convergence. In the simulation, we displayed the cost at the end of iterations corresponding to each neuron size and the time it take for the calculation. All other parameters are the same.

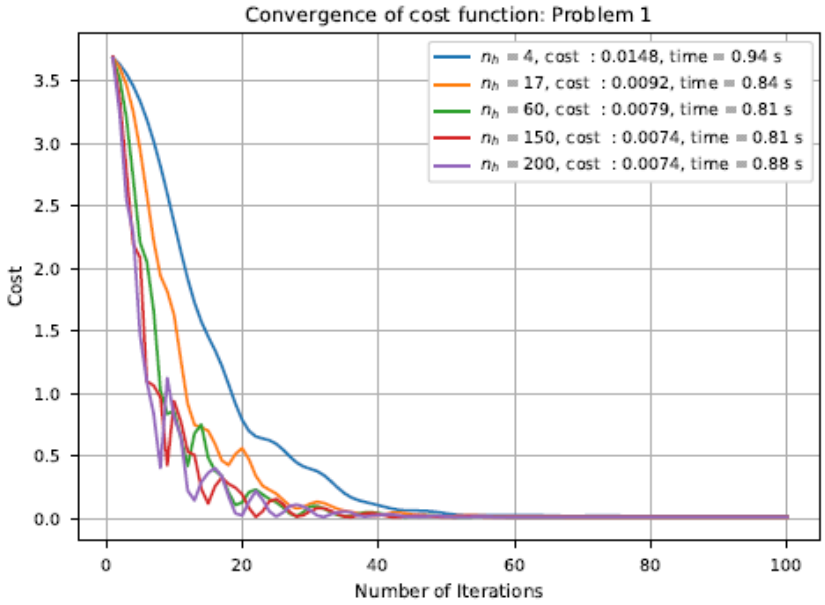


Figure 3: Convergence of loss functions for system (8).

From the simulation shown Figure 3, we observe that, one can obtain the required accuracy even for a single neuron in the hidden layer. However, it needs large number of iterations for smaller number of neurons leading to problem of computational time. Increasing the numbers of neuron has advantage on the performance of the model. However, an arbitrary increase is unnecessary. In this case h = 60 has similar accuracy with h = 200 with less computational time.

The next experiment is on the number of hidden layers. Raissi et al. (2019), have shown that for Burgers' equation, more hidden layer results in better performance as far as error is concerned. Also, Berg & Nystrom

(2018), observed the improvement of accuracy of solving diffusion equation. In our case, fixing all parameters and activation functions the same as the previous experiments, we performed a simulation to compare one hidden layer and two hidden layers varying the numbers of neurons.

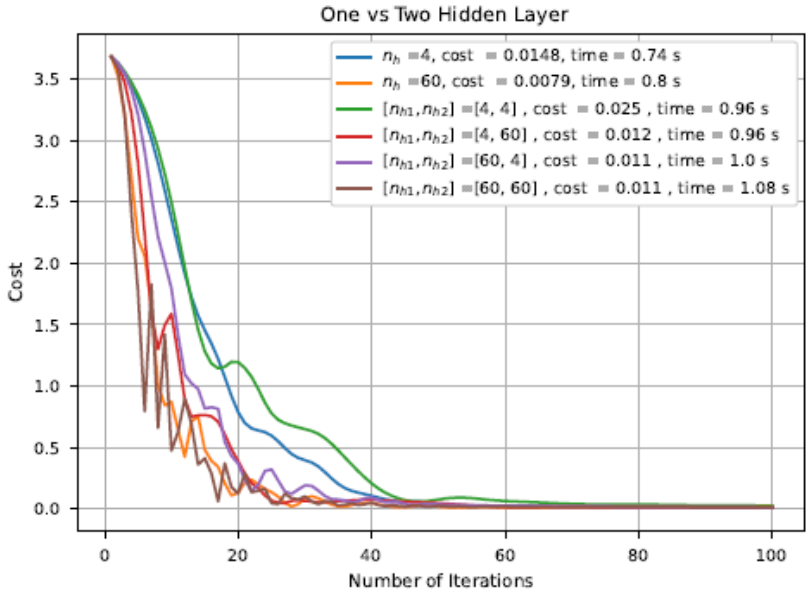


Figure 4: Convergence of error functions for problem (8), two hidden layers

In Figure 4, the result shows that, for the system of differential equation (8), adding more hidden layer do not lead to better performance.

5.2. Numerical solutions

Now we use the ANN method for solving the system of differential equa- tions. In line with the above simulations, we selected a single hidden layer with 60 neurons (tuning in plus minus may not have significant effect). For this experiment, m = 11, uniform grid points were sampled from the given interval. The solutions using ANN and the corresponding analytical solutions are indicated in Figure 5. The numerical quantities are indicated in Table 1. Table 2 indicates the error due to the neural network method.

Table 1: ANN and analytical solutions

t	y_1 ANN	y_1 Analytic	y_2 ANN	y_2 Analyitc
0.0	0.000000	0.000000	1.000000	1.00
0.1	0.099759	0.099833	1.009897	1.01
0.2	0.198447	0.198669	1.039812	1.04
0.3	0.295184	0.295520	1.089752	1.09
0.4	0.389015	0.389418	1.159711	1.16
0.5	0.478967	0.479426	1.249679	1.25
0.6	0.564089	0.564642	1.359636	1.36
0.7	0.643493	0.644218	1.489553	1.49
0.8	0.716370	0.717356	1.639394	1.64
0.9	0.782005	0.783327	1.809116	1.81
1.0	0.839784	0.841471	1.998666	2.00

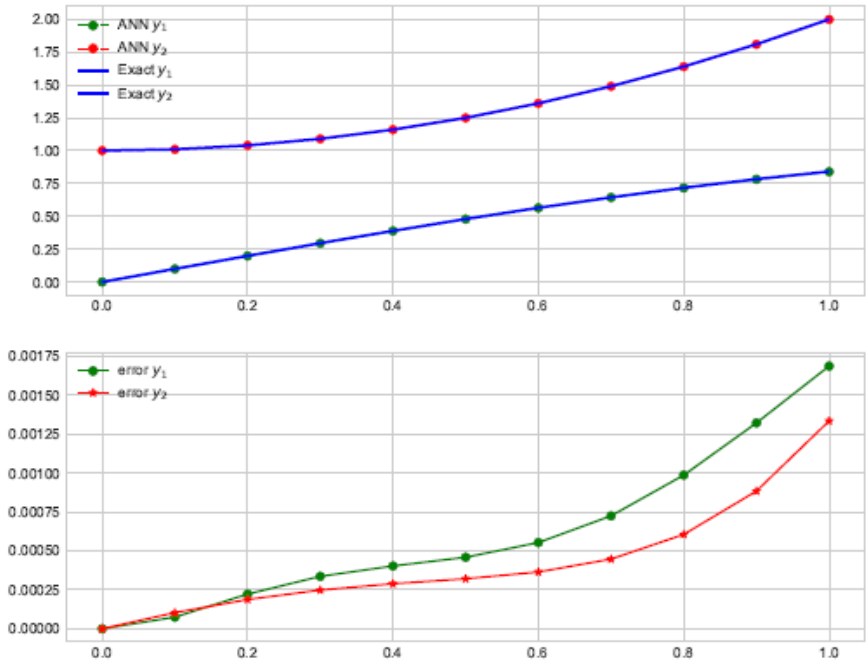


Figure 5: Comparing the ANN solution of (8), with the exact solution and error plot

Table 2: ANN Error

t	error y_1	error y_2
0.0	0.000000	0.000000
0.1	0.000075	0.000103
0.2	0.000223	0.000188
0.3	0.000336	0.000248
0.4	0.000403	0.000289
0.5	0.000459	0.000321
0.6	0.000553	0.000364
0.7	0.000725	0.000447
0.8	0.000987	0.000606
0.9	0.001321	0.000884
1.0	0.001687	0.001334

Table 3: Error at the end point t = 4, RK4 and ANN compared for dierent grid points

Grid points	ANN error	RK4 error
11	0.982	9.588
16	0.959	4.668
21	1.005	1.902
26	0.990	0.825

5.3. Advantage of ANN over the RK4 methods

We selected different sizes of uniform grid points, m = 11, 16, 21 and 26 from the domain [0, 4], and computed solutions of the system of IVPs using the two methods. The simulation in Figure 6 shows one of the significant advantage of using neural network method over other traditional mehtod-finite difference. ANN gives better performance for smaller grid pints. Also, observe that, at the end point t = 4, the ANN is more accurate than the Runga-Kutta method (3). This show that, the method could be employed for application problem requiring large data points. However, for larger grid points, RK4 is more accurate as expected.

6. CONCLUSIONS AND OUTLOOK

In this paper, we presented a vectorized algorithm for solving system of initial value problems using deep neural networks. We conducted different experiment using python code and simulated the result using graphs and tables. We have obtained some insight on the nature of the architecture for the model. We have seen that for some specific problems we can obtain a required accuracy even for a single neuron in the hidden layer. More neuron size provides more accuracy, but more iteration for learning the parameters. Moreover, arbitrary increase of neurons is not recommended. Based on the underlying problem one has to set for the best size of neurons.

We compared the ANN method with the well known forth order Runge- Kutta method. The result showed that, the ANN produced more accurate result for small number of the grid points. Moreover, for larger value of the domain, the ANN method provides better accuracy than RK4 method.

For a future work, further analytical investigation is required to strength the foundation of DNN for solving system of initial value problems including delay differential equations and stochastic differential equations. These include look- ing at stability, convergence and robustness of DNN related to solving system of IVPs. In the same one may investigate the problem using other architec- tures such as, recurrent neural networks, convolutional neural network, deep probabilistic neural network, general adversarial networks.

ACKNOWLEDGMENT

The development of the algorithms and codes were inspired by the online courses on Coursera platform by Professor Andrew Ng, Machine Learning and Neural Networks and Deep Learning. I would like to thank Coursera for the Financial Aid.

REFERENCES

Aarts, L. P., & Van Der Veer, P. (2001). Neural network method for solving partial differential equations. *Neural Processing Letters,14*, 261–271.

Asady, B., Hakimzadegan, F., & Nazarlue, R. (2014). Utilizing artificial neural network approach for solving two-dimensional integral equations. *Mathemat- ical Sciences*, *8*,117.

Basheer, I., & Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, *43*, 3–31. URL: https://www.sciencedirect.com/science/article/pii/S0167701200002013. doi:https://doi.org/10.1016/S0167-7012(00) 00201-3. Neural Computting in Micrbiology.

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, *18*.

Berg, J., & Nyström, K. (2018). A unified deep artificial neural network ap- proach to partial differential equations in complex geometries. *Neurocomput ing*, *317*, 28 – 41. doi:https://doi.org/10.1016/j.neucom.2018.06.056. Berg, J., & Nyström, K. (2019). Data-driven discovery of pdes in complex datasets. *Journal of Computational Physics*, *384*, 239 – 252. doi:https://doi.org/10.1016/j.jcp.2019.01.036.

Bouwmans, T., Javed, S., Sultana, M., & Jung, S. K. (2019). Deep neural network concepts for background subtraction: A systematic review and com- parative evaluation. *Neural Networks*, *117*, 8–66.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., &Zhang,

Q. (2018). Jax: composable transformations of python+numpy programs. URL: http://github.com/google/jax.

Bruna, J., & Dec, L. (2018). Mathematics of deep learning. *Courant Institute of Mathematical Science, NYU*, .

Calin, O. (2020). *Deep Learning Architectures*. Springer.

Chakraverty, S., & Mall, S. (2017). *Artificial neural networks for engineers and scientists: solving ordinary differential equations*. CRC Press.

Coddington, E. A., & Levinson, N. (1955). *Theory of ordinary differential equations*. Tata McGraw-Hill Education.

Dixit, P., & Silakari, S. (2021). Deep learning algorithms for cybersecurity applications: A technological and status review. *Computer Science Review*, *39*, 100317.

Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review*, *40*, 100379.

Elsken, T., Metzen, J. H., Hutter, F. et al. (2019). Neural architecture search: A survey. *J. Mach. Learn. Res.*, *20*, 1–21.

Freeman, J. A., & Skapura, D. M. (1991). *Neural networks: algorithms, appli- cations, and programming techniques*. Addison Wesley Longman Publishing Co., Inc.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* volume 1. MIT press Cambridge.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. et al. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, 354–377.

Haber, E., & Ruthotto, L. (2017). Stable architectures for deep neural networks. *Inverse Problems*, *34*,014004.

Lagaris, I. E., Likas, A., & Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, *9*,987–1000.

Lee, H., & Kang, I. S. (1990). Neural algorithm for solving differential equations. *Journal of Computational Physics*, *91*, 110–131.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234,11–26.

Malek, A., & Shekari Beidokhti, R. (2006). Numerical solution for high order differential equations using a hybrid neural network—optimization method. *Applied Mathematics and Computation*, *183*, 260 – 271. doi:https://doi. org/10.1016/j.amc.2006.05.068.

Malladi, S., & Sharapov, I. (2018). Fastnorm: Improving numerical stability of deep network training with efficient normalization.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*, 115–133.

Meade, A., & Fernandez, A. (1994). The numerical solution of linear or- dinary differential equations by feedforward neural networks. *Mathemati- cal and Computer Modelling*, *19*, 1 – 25. doi:https://doi.org/10.1016/ 0895-7177(94)90095-7.

Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., & Ter- zopoulos, D. (2021). Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Nascimento, R. G., Fricke, K., & Viana, F. A. (2020). A tutorial on solving or- dinary differential equations using python and hybrid physics-informed neu- ral network. *Engineering Applications of Artificial Intelligence*, *96*, 103996. doi:https://doi.org/10.1016/j.engappai.2020.103996.

Nouiehed, M., & Razaviyayn, M. (2018). Learning deep models: Critical points and local openness. *arXiv preprint arXiv:1803.02968* , .

Ohri, K., & Kumar, M. (2021). Review on self-supervised image recognition using deep neural networks. *Knowledge-Based Systems*, (p. 107090).

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., & Edelman, A. (2020). Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385* , . Raissi, M., Perdikaris, P., & Karniadakis, G. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse prob- lems involving nonlinear partial differential equations. *Journal of Computa- tional Physics*, *378*, 686 – 707. doi:https://doi.org/10.1016/j.jcp.2018.10.045.

Sadoghi Yazdi, H., Pakdaman, M., & Modaghegh, H. (2011). Unsupervised kernel least mean square algorithm for solving ordinary differential equa- tions. *Neurocomputing*, *74*, 2062 – 2071. doi:https://doi.org/10.1016/j.neucom.2010.12.026.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. doi:https://doi.org/10.1016/j.neunet.2014.09.003.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* ,.

Wang, Z., Huan, X., & Garikipati, K. (2019). Variational system identification of the partial differential equations governing the physics of pattern-formation: Inference under varying fidelity and noise. *Computer Methods in Applied Mechanics and Engineering*, *356*,44 – 74. doi:https://doi.org/10.1016/ j.cma.2019.07.007.

Yadav, N., Yadav, A., Kumar, M. et al. (2015). *An introduction to neural network methods for differential equations.* Springer.

Yi, H., Shiyu, S., Xiusheng, D., & Zhigang, C. (2016). A study on deep neural networks framework. In *2016 IEEE Advanced Information Manage- CEC)* (pp. 1519–1522). IEEE. *Ment, Communicates, Electronic and Automation Control Conference (IM-*

Yun, C., Sra, S., & Jadbabaie, A. (2018). A critical view of global optimality in deep learning. *arXiv preprint arXiv:1802.03487* , .

Zheng, S., Song, Y., Leung, T., & Goodfellow, I. (2016). Improving the robust- ness of deep neural networks via stability training. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4480–4488).

Zheng, Z., & Hong, P. (2018). Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks. In *Proceedings of* (pp. 924–7933). *The 32nd International Conference on Neural Information Processing Systems.*

Syndicate Session 2

AI in Practice and AI for Sustainable Development

Artificial Intelligence-based System for Diagnosis of Cardiovascular Diseases

Gizeaddis Lamesgin Simegn[1,] *, Worku Birhanie Gebeyehu[2], Mizanu Zelalem Degu[2]

[1]*School of Biomedical Engineering, Jimma Institute of Technology, Jimma University, Jimma Ethiopia*
[2]*Faculty of Computing, Jimma Institute of Technology, Jimma University, Jimma Ethiopia*
Corresponding author, e-mail: gizeaddis.lamesgin@ju.edu.et

ABSTRACT

Cardiovascular diseases are the leading causes of death worldwide and the number of people dying from cardiovascular disease is steadily increasing. The rapid economic transformation leading to environmental changes and unhealthy lifestyles increase the risk factors and incidence of cardiovascular disease. The limited access to health facilities, lack of expert cardiologists, and lack of regular health check-up trends make CVD the silent killers in low-resource settings. Computer-aided diagnosis using Artificial intelligence techniques (AI) can help reduce the mortality rate due to heart disease by providing decision support to experts allowing early diagnosis and treatment. In this paper, an AI-based system has been proposed for the diagnosis of cardiovascular diseases using clinical data, patient information, and electrocardiogram (ECG) signal. The proposed system includes an ECG processor part that allows cardiologists to process and analyze the different waveforms, a machine learning-based heart disease prediction system based on patient information and clinical data, and a deep learning-based 18 heart conditions multiclass classification system using a 12-lead ECG signal. A user-friendly user interface has been also developed for ease of use of the proposed techniques. The developed AI-based system was found to be 100% accurate in predicting health disease based on clinical and patient information, and 93.27% accurate, on average, classifying heart conditions based on a 12-lead ECG signal. The ECG processor also simplifies the analysis of important ECG waveforms and segments. The experimental results indicate that the proposed system may have the potential for facilitating heart disease diagnosis. The proposed method allows physicians to analyze and predict heart disease easily and early, based on the available resource, improving diagnosis accuracy and treatment planning.

Keywords: *Artificial intelligence, AI, Clinical data, Diagnosis, ECG signal, Heart disease*

1. INTRODUCTION

Cardiovascular diseases (CVDs) are groups of disorders of the heat and the blood vessels including heart disease, cerebrovascular disease, rheumatic heart disease and other conditions. Heart disease occurs when the heart fails to supply sufficient blood to other parts of the body to accomplish their normal functionality (Bui, Horwich et al. 2011). This could be due to blockage and narrowing of coronary arteries which are responsible for the supply of blood to the heart itself. CVD are the leading cause of death globally, taking an estimated 17.9 million lives each year and more than 75% of these deaths occur in low- and middle-income countries (LMICs) (WHO 2021). Even though evidence on the national burden of cardiovascular diseases (CVDs) is limited in Ethiopia, according to a systematic review conducted in 2014, the prevalence of CVD ranges from 7.2 to 24% (Misganaw, Mariam et al. 2014). The trend of CVD and mortality attributed to CVD is still increasing in Ethiopia (Tefera et al., 2017, Gebreyes et al., 2018).

Unhealthy diet, lack of physical activity, tobacco use and improper use of alcohol are the most common behavioral risk factors of heart disease. These can cause high cholesterol level, high blood pressure

increasing the risk of heart disease (Das et al., 2009). Identifying those at highest risk of CVDs and ensuring they receive appropriate treatment can prevent premature deaths.

Access to noncommunicable disease medicines and basic health technologies in all primary health care facilities is essential to ensure that those in need receive treatment and counselling. The risk factors can be measured at primary health facilities and diagnosis of heart disease can be made based on the laboratory results. However, complete and accurate diagnosis requires analysis and integration of many laboratory data and patient information which could be complex and the manual procedure may sometimes lead to misdiagnosis.

In developing countries, the diagnosis and treatment of heart disease is often complex, especially since diagnostic apparatus is often unavailable, as are experts and other resources, resulting in less proper prediction and treatment of heart patients (Coca et al., 2008; Yang and Garibaldi, 2015). It is essential to reduce the potential risks associated with heart disease and improve heart security by accurately and properly diagnosing heart disease risk in patients (De Silva et al., 2008). Artificial intelligence can help clinicians to make more accurate predictions for patients improving the current cardiovascular disease diagnosis and treatment by analyzing big data.

In recent years, to overcome the limitations of manual diagnosis procedure, literatures have proposed different predictive machine learning techniques based on Support Vector Machines (SVM), K-Nearest Neighbor (KNN), Naïve Bayes (NB), and Decision Tree (DT), deep learning models and others (Detrano et al., 1989; Kahramanli and Allahverdi, 2008; Das et al., 2009; Gudadhe et al., 2010; Olaniyi et al., 2015; Patel et al., 2015; Haq et al., 2018; Tomov and Tomov, 2018; Ali et al., 2019; Khourdifi and Bahaj, 2019; Latha and Jeeva, 2019; Muhammad et al., 2020). For example, Detrano et al. (Detrano et al., 1989) have used a logistic regression classification algorithm for heart disease detection and claimed a classification accuracy of 77.1%. Similarly, Kahramanli et al. (Kahramanli and Allahverdi, 2008) proposed a heart disease classification system integrating neural networks with an artificial neural network and claimed an accuracy of 82.4%. Likewise, Tomov et al. (2018) came up with a deep neural network model for heart disease prediction claiming an accuracy of 99% and 0.98 Matthews Correlation Coefficient (MCC). Ali et al. (2019) proposed an expert system using stacked SVM for the prediction of heart disease and reported a 91.11% classification accuracy. However, it is difficult to predict heart diseases easily because the data required for diagnosis related to the disease are multi-modal. To achieve high accuracy of prediction, a multimodal based method for predicting and classifying heart disease occurrence is required. Moreover, many of the automatic health disease diagnosis techniques proposed in the literature are either less accurate, dependent on clinical data, or medical imaging data or ECG signals alone. The purpose of this paper was therefore, to develop an integrated tool that allows physicians analyze ECG signals acquired from patients and get a decision support in the prediction and classification of heart disease using clinical data, patient information and standard 12 lead ECG record.

2. METHODS

In this paper, a structured patient information (age, gender, history of hypertension, etc.), streaming clinical data (heart rate, blood pressure, etc.), ECG signal data was first processed and analysed. An ECG

processor that denoises the signal, extracts THE QRS complex, ECG waves, analyzes and calculates the ECG waves amplitude and duration as well as the heart rate was developed. Then feature fusion of the structured data and streaming data was performed to train and validate a machine learning model for heart disease prediction. The 12 lead ECG data was also used to train and validate a deep learning model multi-class classification of 18 cardiac conditions. Finally, a user-friendly web-based system was developed for ease of use of the developed sub-systems for diagnosis of heart disease. Figure 3 demonstrates the proposed system flowchart.

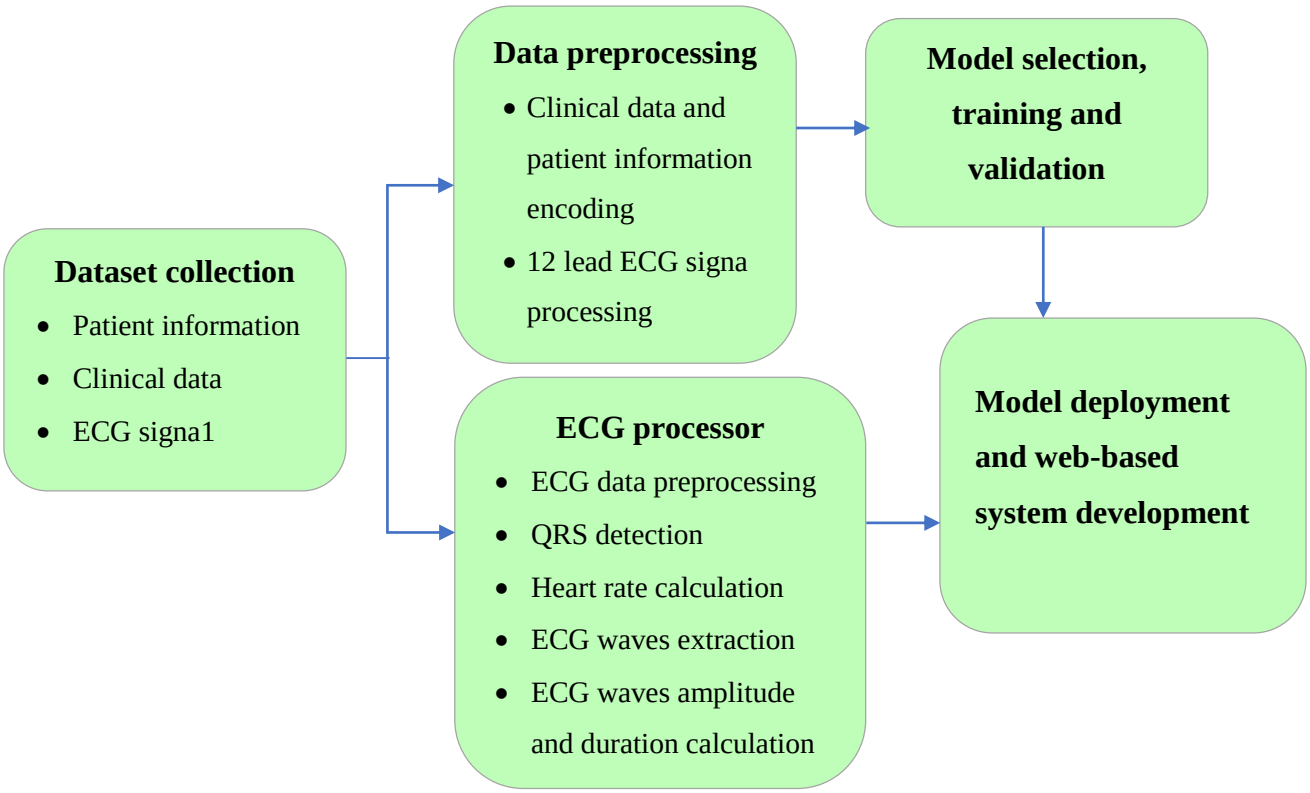


Figure 1: Flowchart of the proposed AI based heart disease diagnosis tool

2.1. Data collection

To implement the proposed system, the first step was data collection. A total of 1190 observations containing different attribute information such as age, sex, chest pain type, blood pressure, cholesterol in mg/dl, blood sugar, maximum heart rate etc. were acquired from University of California Irvine (UCI) Machine Learning Repository (Dua and Graff 2019) which was collected from 5 different heart datasets. The five datasets used for its curation include Cleveland V.A. Medical Center (303 observations), Hungarian (294 observations), Switzerland (123 observations), Long Beach V.A. Medical Center (200 observations) and Stalog (Heart) dataset (270 observation). Table 1 demonstrates the sample observations of 10 individuals. The data contains 45.5% people with heart disease and 54.5% normal people.

Table 1: Sample observations collected from 6 heart disease patients and 4 normal individuals. *(age*: the person's age in years, *sex*: the person's sex (1 = male, 0 = female), *cp*: the chest pain experienced (value 0: typical angina, value 1: atypical angina, value 2: non-anginal pain, value 3: asymptomatic), *trestbps*: the person's resting blood pressure (mm hg on admission to the hospital), *chol*: the person's cholesterol measurement in mg/dl, *fbs*: the person's fasting blood sugar (> 120 mg/dl, 1 = true; 0 = false), *thalach*: the person's maximum heart rate achieved, *exang*: exercise induced angina (1 = yes; 0 = no), *oldpeak*: ST depression induced by exercise relative to rest ('ST' relates to positions on the ecg plot), *slope*: the slope of the peak exercise ST segment (value 1: upsloping, value 2: flat, value 3: down sloping), *ca*: the number of major vessels (0-3), *thal*: a blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect), *target*: heart disease (0 = no, 1 = yes)).

S.N.	age	sex	cp	trestbps	chol	fbs	thalach	exang	oldpeak	slope	ca	thal	target
1	71	0	0	112	149	0	125	0	1.6	1	0	2	1
2	43	0	0	132	341	1	136	1	3.0	1	0	3	0
3	34	0	1	118	210	0	192	0	0.7	2	0	2	1
4	51	1	0	140	298	0	122	1	4.2	1	3	3	0
5	52	1	0	128	204	1	156	1	1.0	1	0	0	0
6	34	0	1	118	210	0	192	0	0.7	2	0	2	1
7	51	0	2	140	308	0	142	0	1.5	2	1	2	1
8	54	1	0	124	266	0	109	1	2.2	1	1	3	0
9	50	0	1	120	244	0	162	0	1.1	2	0	2	1
10	58	1	2	140	211	1	165	0	0.0	2	0	2	1

The ECG data for the heart disease classification model training and testing, which was a total of 23,924 ECG recordings labeled with 18 cardiac abnormalities, were gathered from 4 different sources: (i) southeast University, China, including the data from the China Physiological Signal Challenge 2018 (2 datasets from this source), (ii) St. Petersburg Institute of Cardiological Technics, St. Petersburg, Russia, (iii) the Physikalisch Technische Bundesanstalt, Brunswick, Germany. (2 datasets from this source), and (iv) Georgia 12-Lead ECG Challenge Database, Emory University, Atlanta, Georgia, USA. Demographic information i.e., age and sex were also included in the data. Table 2 demonstrates the heart disease/conditions and number of data collected, for each class, for model training.

2.2. Data preprocessing and visualization

During the data preprocessing, all features of the heart disease prediction dataset (patient information and clinical data) were first converted into numeric ones, and then different values were grouped into their categories. After feature conversion, the correlation between every two features was analyzed to determine whether the information among features is redundant. The correlation matrix is computed to check the linear relationship between the variables, which is used to identify the highly correlated variables. High correlation magnitudes indicate that the variables contain similar information. The correlation filtering is intended to remove the redundant variables.

All the 12-lead ECG data and the corresponding gender and age information, that was collected for heart disease/conditions multiclassification, were one-hot encoded prior to feeding to the model for training.

Table 2: The 12-lead ECG collected data and heart disease/conditions

S.No.	ECG cardiac abnormalities	Abbreviation	Number of data
1	Ventricular Premature Beats	VPB	764
2	Right Axis Deviation	RAD	38
3	Right Bundle Branch Block	RBBB	3934
4	T-Wave Inversion	TInv	120
5	Supraventricular Premature Beats	SVPB	1664
6	Prolonged QT Interval	LQT	1106
7	Atrial Fibrillation	AFL	188
8	Atrial Flutter	AF	3904
9	Left Bundle Branch Block	LBBB	1420
10	Q-Wave Abnormal	QAb	180
11	T-Wave Abnormal	TAb	3116
12	1st Degree Av Block	IAVB	2336
13	Premature Atrial Contraction	PAC	1664
14	Sinus Bradycardia	SB	1422
15	Premature Ventricular contraction	PVC	764
16	Left Anterior Fascicular Block	LAnFB	228
17	Nonspecific Intraventricular Conduction Disorder	NSIVCB	340
18	Incomplete Right Bundle Branch Block	1RBBB	736

2.3.Training and testing heart disease prediction models

In this paper, two machine learning models, XGBoost and Random Forest and a artificial neural network (ANN) deep learning model have been trained and tested. In order to evaluate the effectiveness of each model and select the best performing model, the same data was used to test XGBoost, random forest machine learning models and ANN deep learning model.

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting designed to be highly efficient, flexible and portable (Chen and Guestrin, 2016). It can be applied in prediction problems involving unstructured data (images, text, etc.), in a wide range of applications to solve regression, classification, ranking, and user-defined prediction problems. In this paper, the XGBoost model was implemented with a learning rate of 0.01, L1 regularization value of 5, L2 regularization value of 2, and 2000 number of estimators or runs (model learning iterations). Random forest is a supervised learning algorithm which is used for both classification as well as regression (Breiman 2001). Random forest, as the name implies, consists of a large number of individual decision tress that operate as an ensemble. It creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result. In this paper, the

random forest algorithm was implemented with 600 number of decision trees (estimators) and other default parameters.

For both of the heart disease prediction model trainings, initially, the data was randomly divided into training set (80%) and test set (20%). A 10-fold cross validation technique was in which the training set was split into ten parts of approximately equal size, in which nine parts are used for training and one part is used for validation. This process is repeated ten times iteratively and the average of these accuracy is taken as the expected prediction accuracy.

The ANN was implemented using a standard feed-forward back-propagation neural network (BPNN) model. The network has three layers, an input layer with 13 neuros, hidden layer with 11 neurons and a 1 neuron output layer. A uniform kernel initializer, ReLu activation function in the input and hidden layer, the sigmoid activation function in the output layer, an Adam optimizer, and a binary cross entropy (to compare the predicted probabilities to actual class output), batch size of 10 and 100 number of epochs were used in training the model. 80% of the data were used for training while 20% of the data were used for testing.

2.4.Training and testing of heart disease Classification model

For the classification of the 18-heart disease/conditions from 12-lead ECG recordings, a conventional neural network (CNN) was trained and validated. The model was designed to accept two separate inputs: (i) ECG signal and (ii) age and gender. For feature extraction of the first input (ECG signal), 3 one dimensional conventional neural networks (Conv1D) with 5000 input length and 12 steps were used. For the second input feature extraction two dense layers were used. The outputs of the first and second feature extracting blocks were then concatenated. Finally, a dense layer with 18 outputs was used for final classification. The model uses ReLu activation function for the conventional layers and sigmoid activation function for the dense layer, Adam as an optimizer, and a binary cross entropy loss function. The model was trained for 50 number of epochs and batch size of 50. Figure 2 illustrates the simplistic architecture of the proposed heart disease classification model.

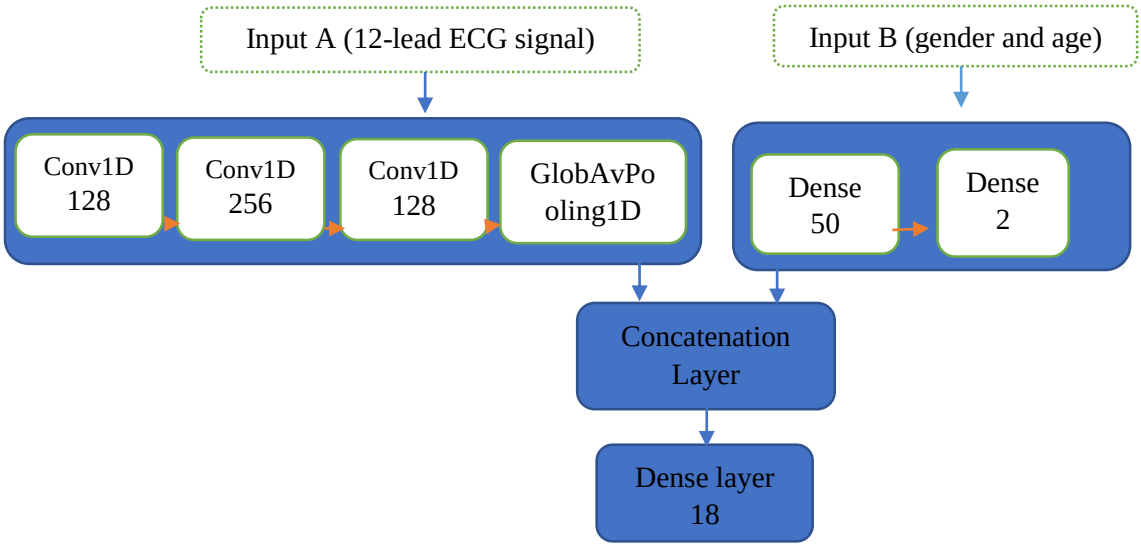


Figure 2: Simplistic architecture of the heart disease classification model. Conv1D: 1 dimensional CNN, GlobAvPooling: 1 dimensional Global Average Pooling, Dense: dense layer

2.5. ECG processor

The electrocardiogram (ECG) signal provides key information about the electrical activity of the heart. It is the most important Biosignal used by cardiologists for diagnosis of heart disease. ECG signal readings and analysis are done after signal processing. ECG signal processing techniques include de-noising or noise removal, baseline correction, wave form and parameter extraction and abnormality detection. An ECG waveform consists of five basic waves called P, Q, R, S, and T-waves and sometimes U-waves. The P-wave indicates the successive depolarization of right atria and left atria, QRS complex indicates the ventricular depolarization, T-wave represents the ventricular repolarization and the U-wave represents the repolarization of the papillary muscles. The most important part of the ECG signal analysis is the shape of QRS complex which is the combination of three of the graphical deflections seen on the typical ECG.

ECG signals have frequency range of 0.5 Hz to 100 Hz. However, the signal is exposed to contamination of different noises and artifacts during acquisition. There are mainly three artefacts/noises in ECG signal: the high frequency noise, low frequency noise and the power line interference. In this work, finite impulse response (FIR) digital filters using Kaiser window (Kaiser and Schafer 1980) were designed and implemented to remove high frequency noise, low frequency noise, and powerline interference from the ECG signals. The low pass and high pass filters were designed with 100 Hz and 0.5 Hz cutoff frequencies, respectively, and order of 100. Similarly, a notch filter with 50 Hz central frequency and order of 100 was designed for removal of the power line interference.

After noise removal, ECG feature extraction system was designed to extract important features of the ECG signal including R-peak detection, detection and delineation of PQST peaks and waves and determination of each of the ECG waves amplitudes and intervals. The Neurokit2 (Makowski, Pham et al. 2021) discrete wavelet method of ECG peaks detection package has been used to extract and delineate the ECG peaks. After extraction of the required peaks, an algorithm has been developed for calculation of ST depression, WRS duration, slope of ST segment, QT interval, amplitude of the R peak, amplitude of the Q peak, amplitude of P wave, amplitude of T wave, PR interval, corrected QT interval using Bazett formula (Bazett 1920) and the average heart rate, which are important indicators of presence of heart disease or abnormality.

3. RESULTS

3.1. Data pre-processing and visualization

In the pre-processing stage, the different attribute information used for training of the heart disease prediction model were converted into numeric values and analysed. As demonstrated in the correlation plot of Figure 3, chest pain, the maximum heart rate and slope of peak exercise ST segment are highly correlated with the target (having heart disease or not). Figure 4 demonstrates the number of people (in the collected data) with each chest pain type (angina) and the relation between the types of chest pain and heart disease. As indicated, 27.2% persons have chest pain type 0, 82% have chest pain type 1, 79.3% have chest pain type 2 and 69.5% have chest pain type 3. As demonstrated in Figure 4, those who have chest pain type 1 and chest pain type 2 are more likely to be affected by heart disease.

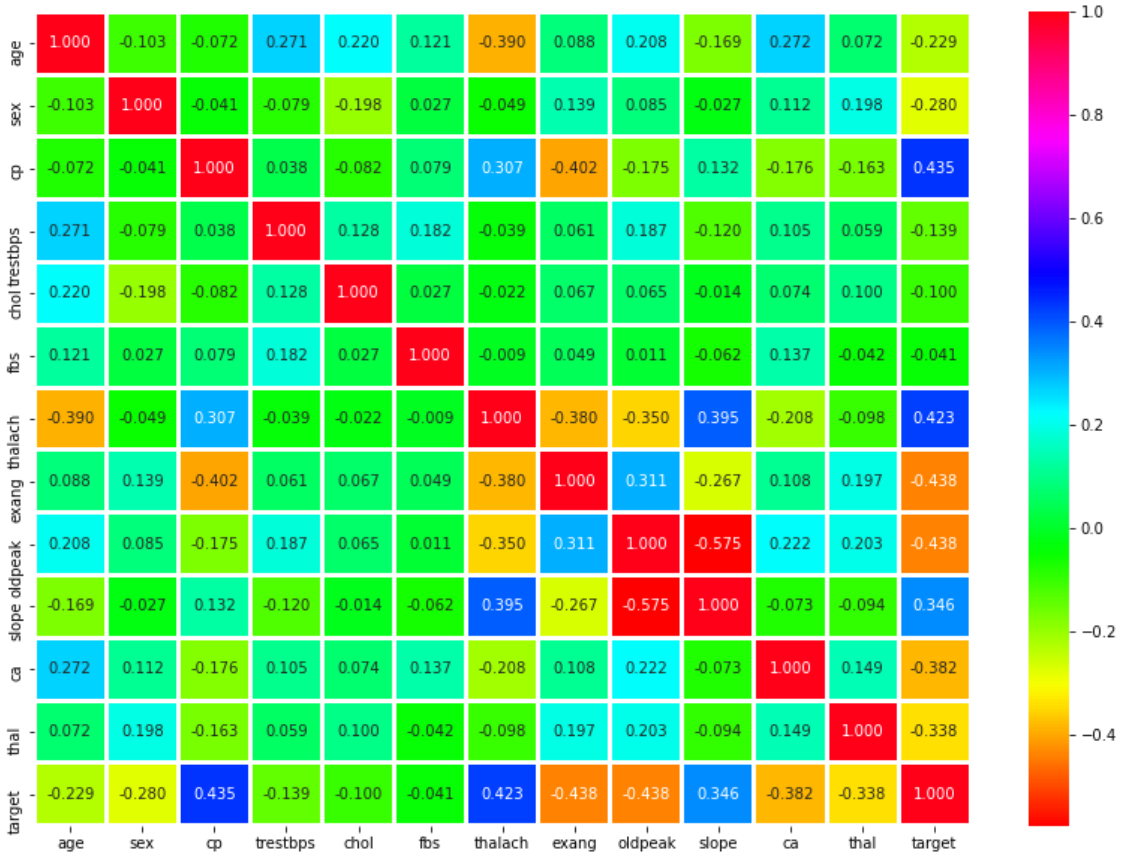


Figure 3: Correlation matrix between features

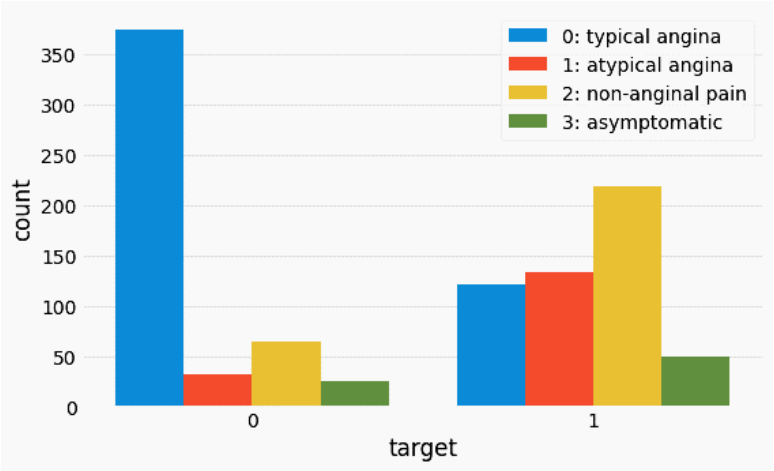


Figure 4: Data visualization demonstrating relation between types of chest pain and heart disease

3.2. Results of Heart disease prediction models

Accuracy, precision, recall, F1-score and Receiver Operating Characteristic (ROC) curve were used as performance metric for model evaluation and comparison. Accuracy, precision, recall and Fl-score are calculated from the actual and model predicted true positive, false positive, false negative, and true negative values.

Figure 5 and 6 show the ROC curves of XGBoost, ANN and random forest models trained using the patient information and clinical data for heart disease prediction. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. The area under an ROC curve (AUC) is a measure of the usefulness of a test and a greater area means a more useful test. AUC values of 0.98, 0.95 and 1 were obtained using the XGBoost, ANN and random forest models, respectively.

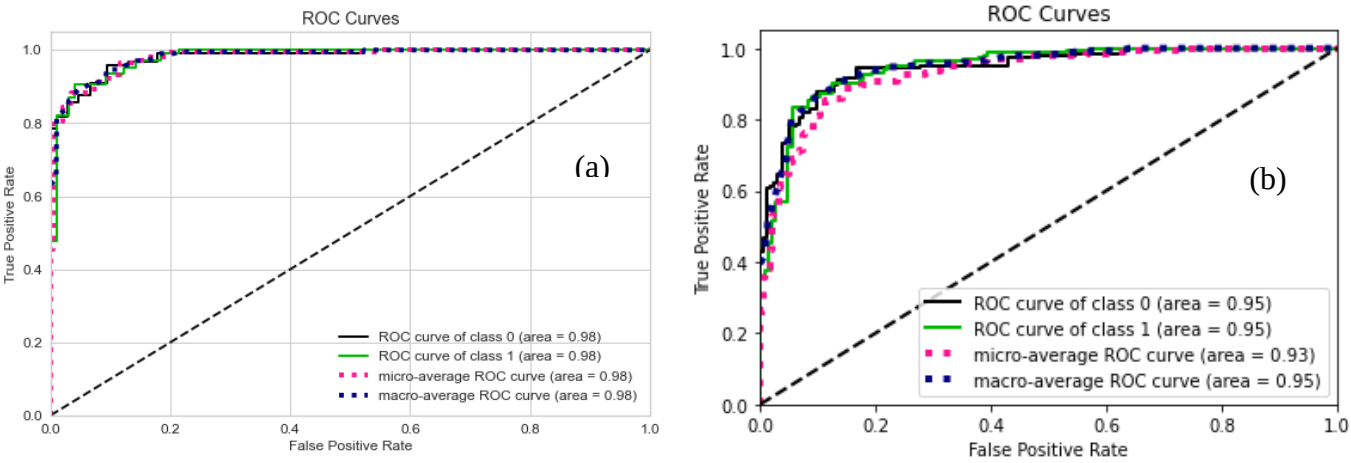


Figure 5: ROC curves of (a) XGBoost and (b) neural network models trained using patient information and clinical data for heart disease prediction

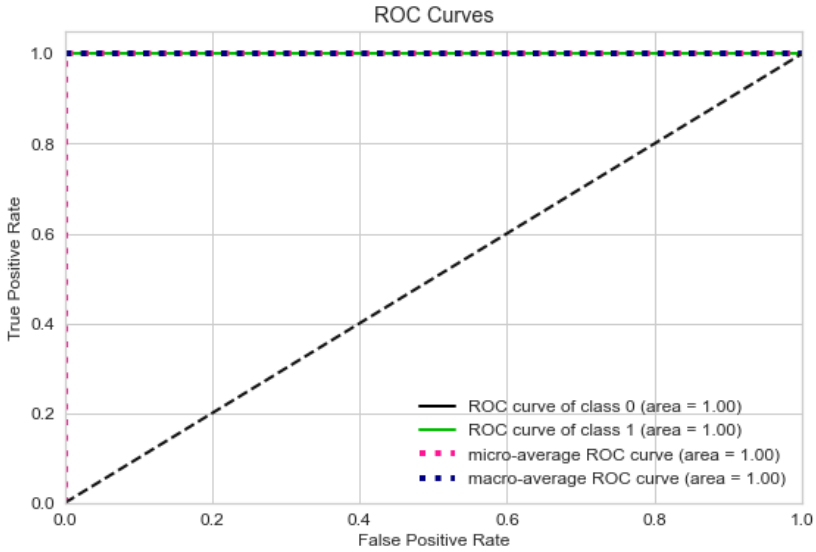


Figure 6: ROC curve of random forest model trained for heart disease prediction using patient information and clinical data

The summary of models' performances on a test data are demonstrated in Table 3. As indicated in Figure 5 and Table 2, the random forest model outperforms the other models on predicting heart disease using the given data with an accuracy of 100%. Hence, the random forest model was selected deployed in our system for heart disease prediction.

Table 2: Summary of models' performance on test data for prediction of heart disease

Performance metrics/ Models	ANN	XGBoost	Random Forest
Area under the curve (AUC)	0.95	0.98	1
Precision (%)	94.07	91.35	100
Recall (%)	79.19	91.15	100
Fl-score (%)	86.16	91.15	100
Accuracy (%)	85.71	92.19	100

3.3. Model results of heart disease classification using 12-lead ECG

Figure 7 demonstrates the normalized confusion matrix of the multi-class classifier. The correct predictions for each class are expressed in the diagonal of the confusion matrix. The values in the off-diagonal illustrate the false positives and false negative results of the model. The model was found to be 93.27 % accurate, in average, classifying the heart conditions.

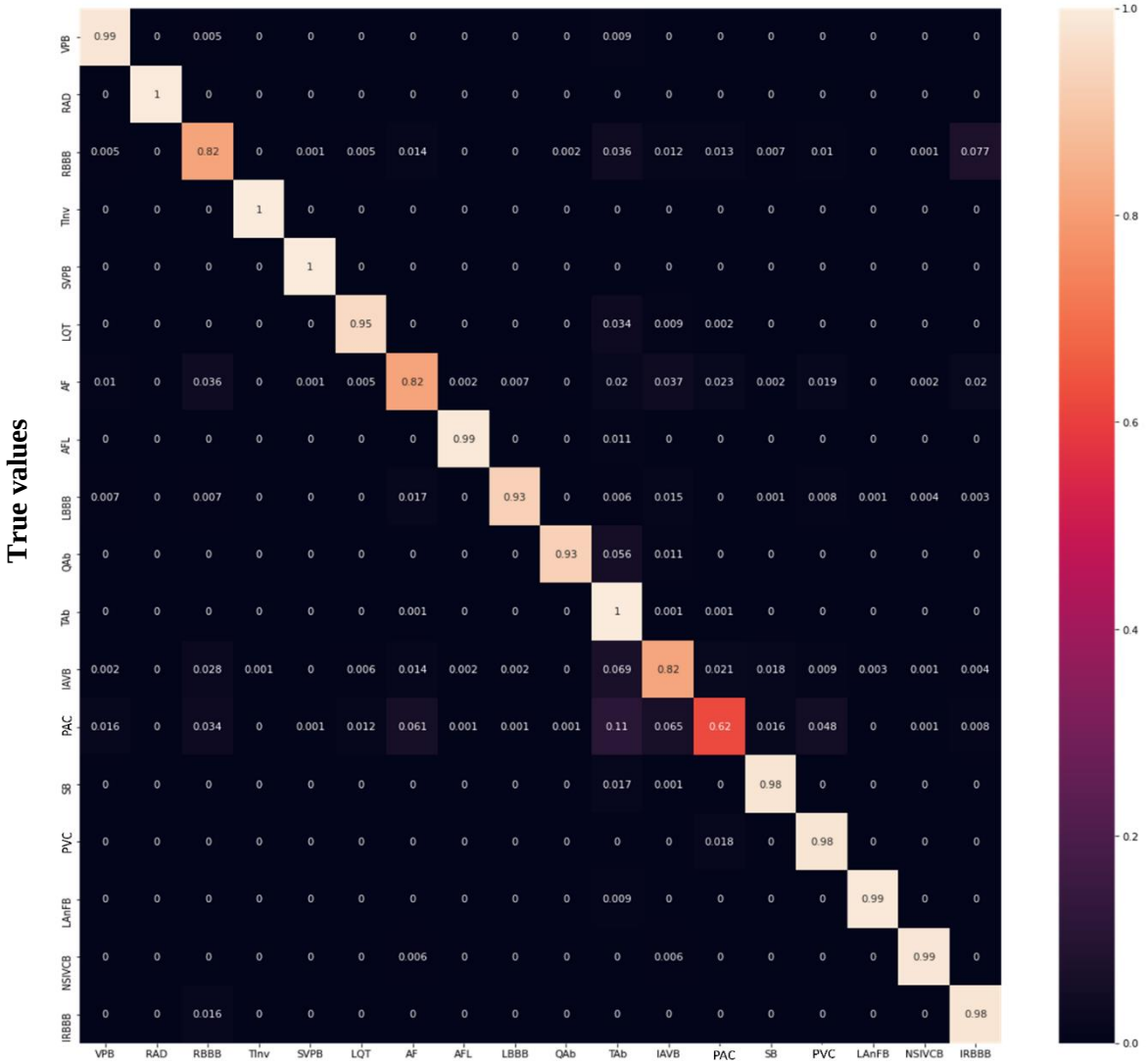


Figure 1: Normalized confusion matrix of the multi-class classifier

3.4. ECG processor

Figure 8 illustrates the raw ECG signal and the processed signal after removal of low-, high-frequency and powerline interference noises. As demonstrated, the base line drift and high frequency noises which are observed in the left signal have been removed in the processed signal. The detected R-peaks, PQST peaks, and delineation of each of the ECG waves are demonstrated in corresponding Figures 9, 10 and 11. After extraction of the required peaks, an algorithm have been developed and deployed in the web-based user interface for calculation of important indicators of heart abnormality including duration and amplitude of ECG wave segments.

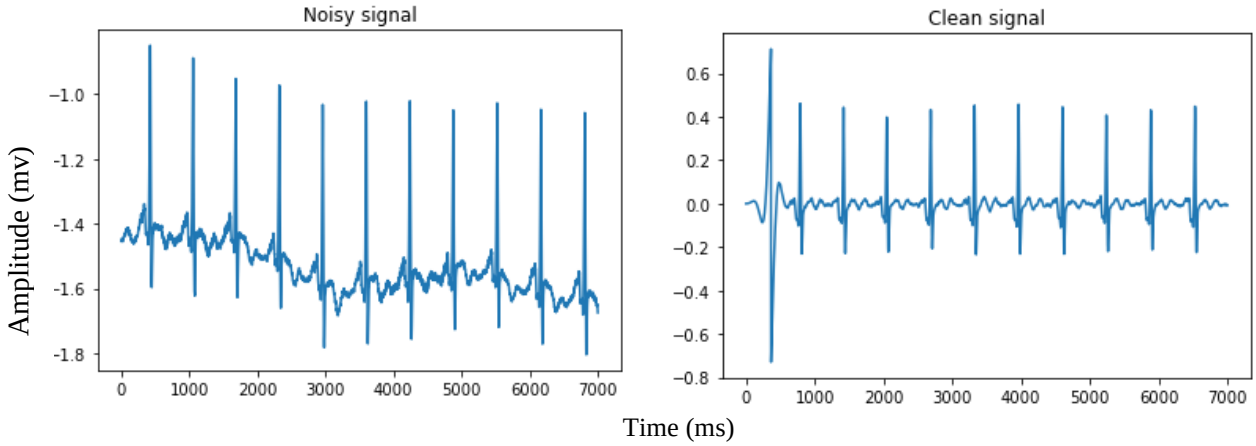


Figure 2: ECG signal noise removal

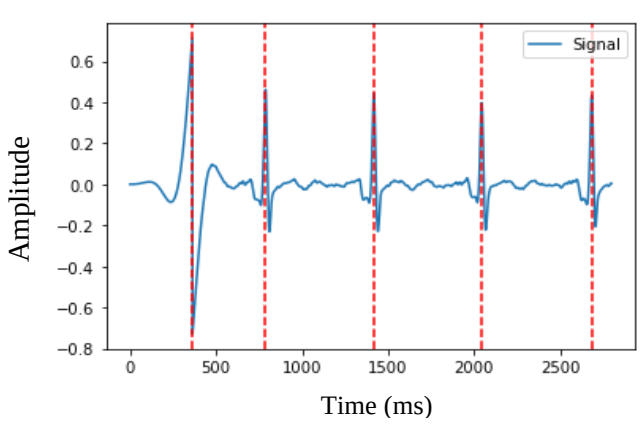


Figure 3: ECG R-peaks detection

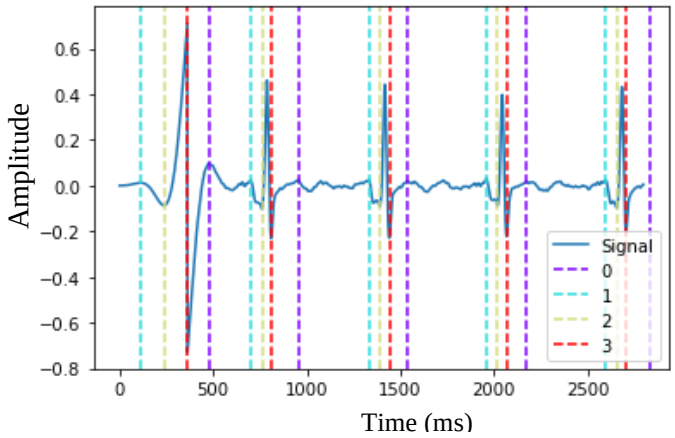


Figure 4: ECG PQST peaks detection

3.5. Web-based user interface (UI)

An integrated web-based user interface was developed for ease of use of the developed heart disease prediction models and ECG signal processor. The web-based UI was developed using Streamlit, which is a free, open-source relatively new browser-based Python framework that allows developers to turn data scripts into web apps. The developed user interface has three parts (pages), ECG processor, heart disease prediction, and heart disease classification from ECG data. Using the ECG processor (Figure 12a and b), users can upload a single lead ECG signal, enter the sampling frequency of the ECG signal, and by pressing

the 'Process' button, they can analyze the different ECG waveforms duration and amplitude for quick diagnosis.

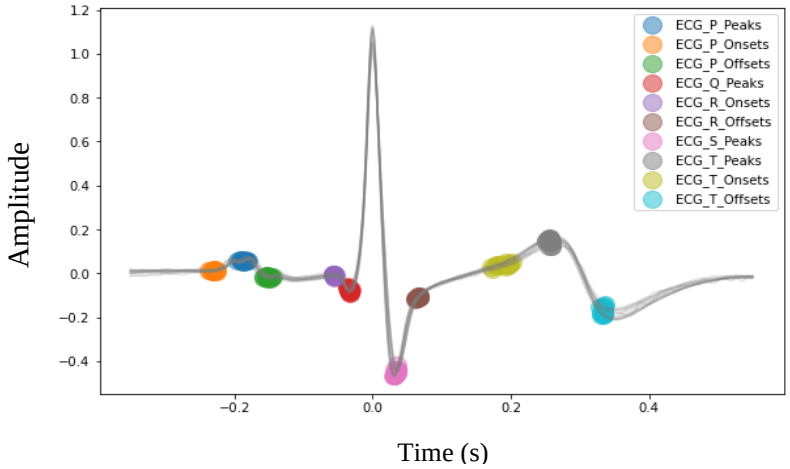
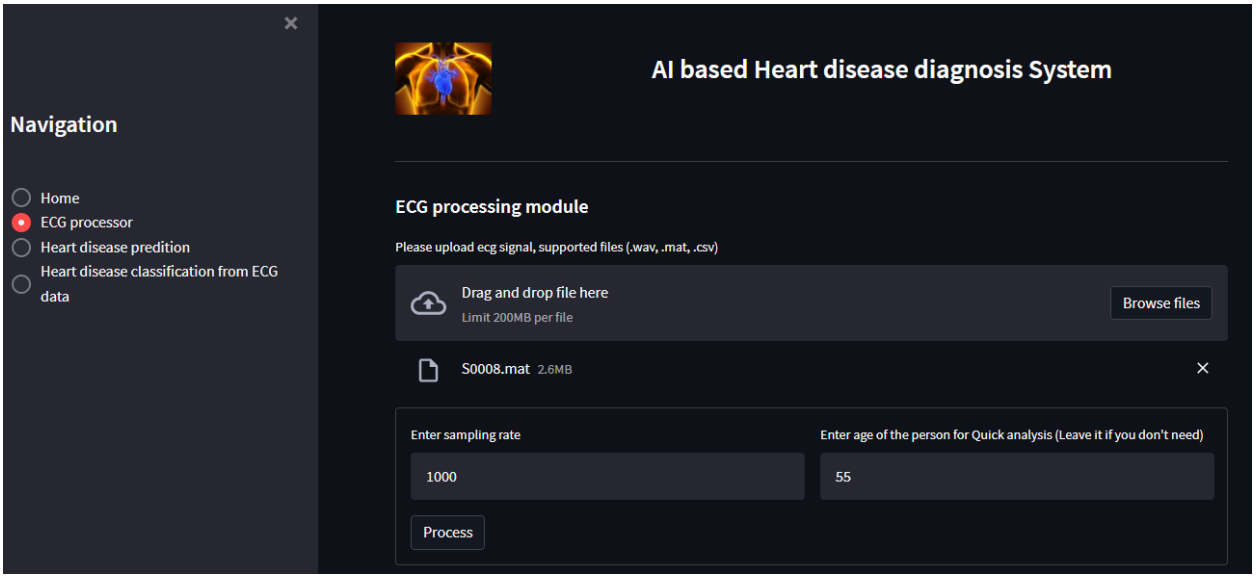


Figure 11: Delineation of ECG waves

(a)



(b)

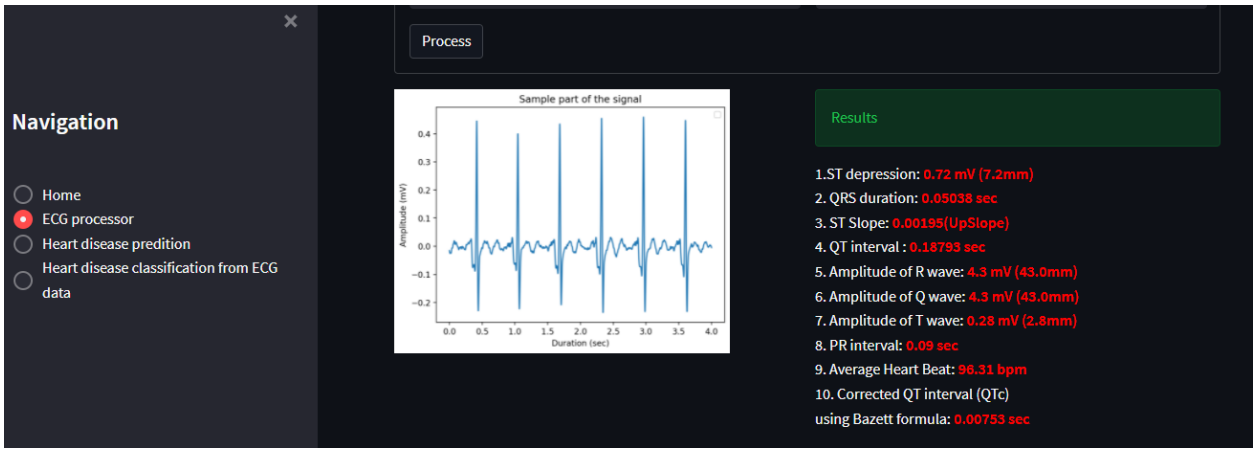
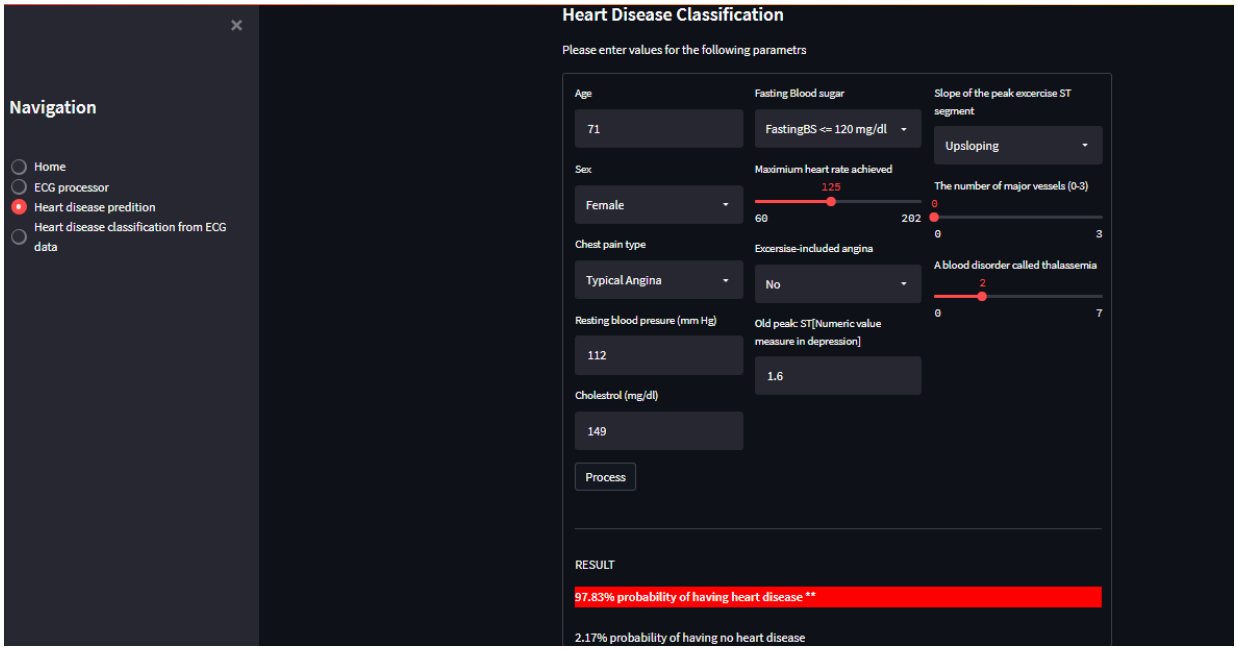


Figure 12: ECG processor page (a) signal uploader (b) quantitative analysis of ECG waveforms

The heart prediction system accepts attribute information including age, sex, chest pain type, blood pressure, cholesterol level, fasting blood sugar, maximum heart rate, exercise induced angina, ST segment depression, the slope of the peak exercise ST segment, number of major vessels and a blood disorder called thalassemia. After the required patient information and clinical data are filled, the system analyses the attributes and predicts whether the person has heart disease or not. Sample observations collected from a patient with heart disease, healthy person and the system's predictions are demonstrated in Figure 13.

(a)



(b)

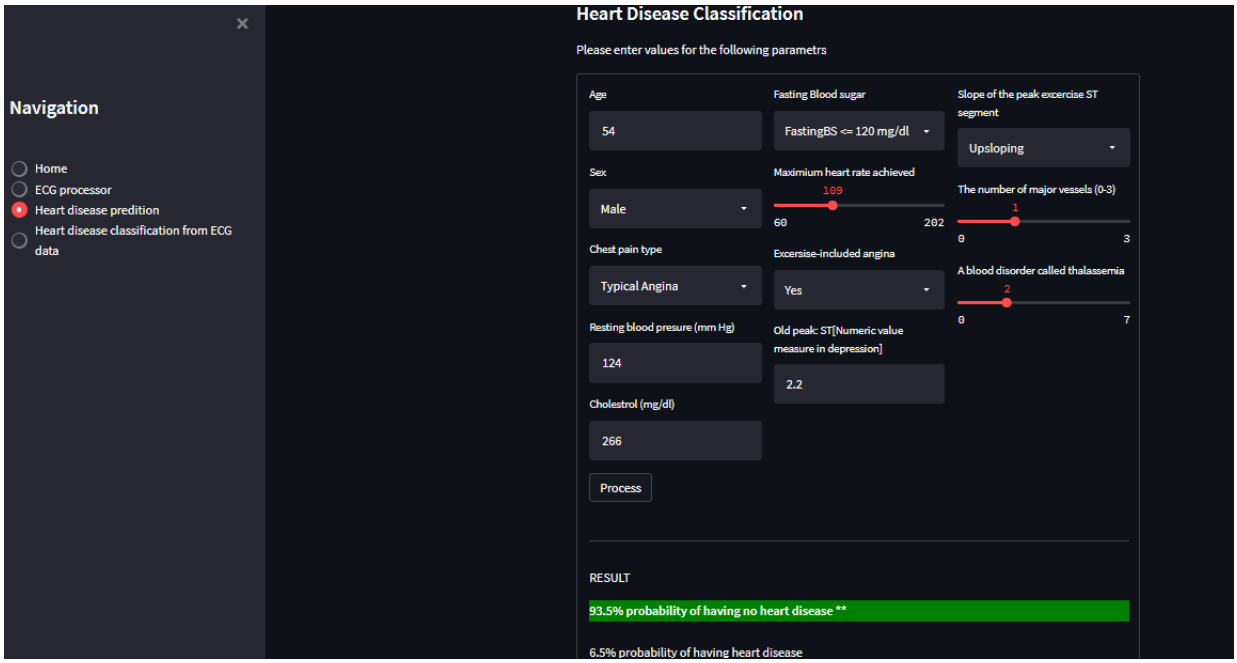


Figure 13: Heart-disease prediction user interface demonstrating typical observations (a) patients with heart disease and system's prediction (b) healthy person and system's prediction

Figure 14 demonstrates snapshot of the heart disease classification user interface part based on a 12-lead ECG and patient information. The system accepts 12-lead ECG signal, the sampling frequency, gender and sex of the patient, analyzes the entered data and provides its prediction. Top five predictions with the model's prediction percentile are displayed in the result's section. This allows the cardiologist to use their expert knowledge and the system predictions and provide the final diagnosis decision.

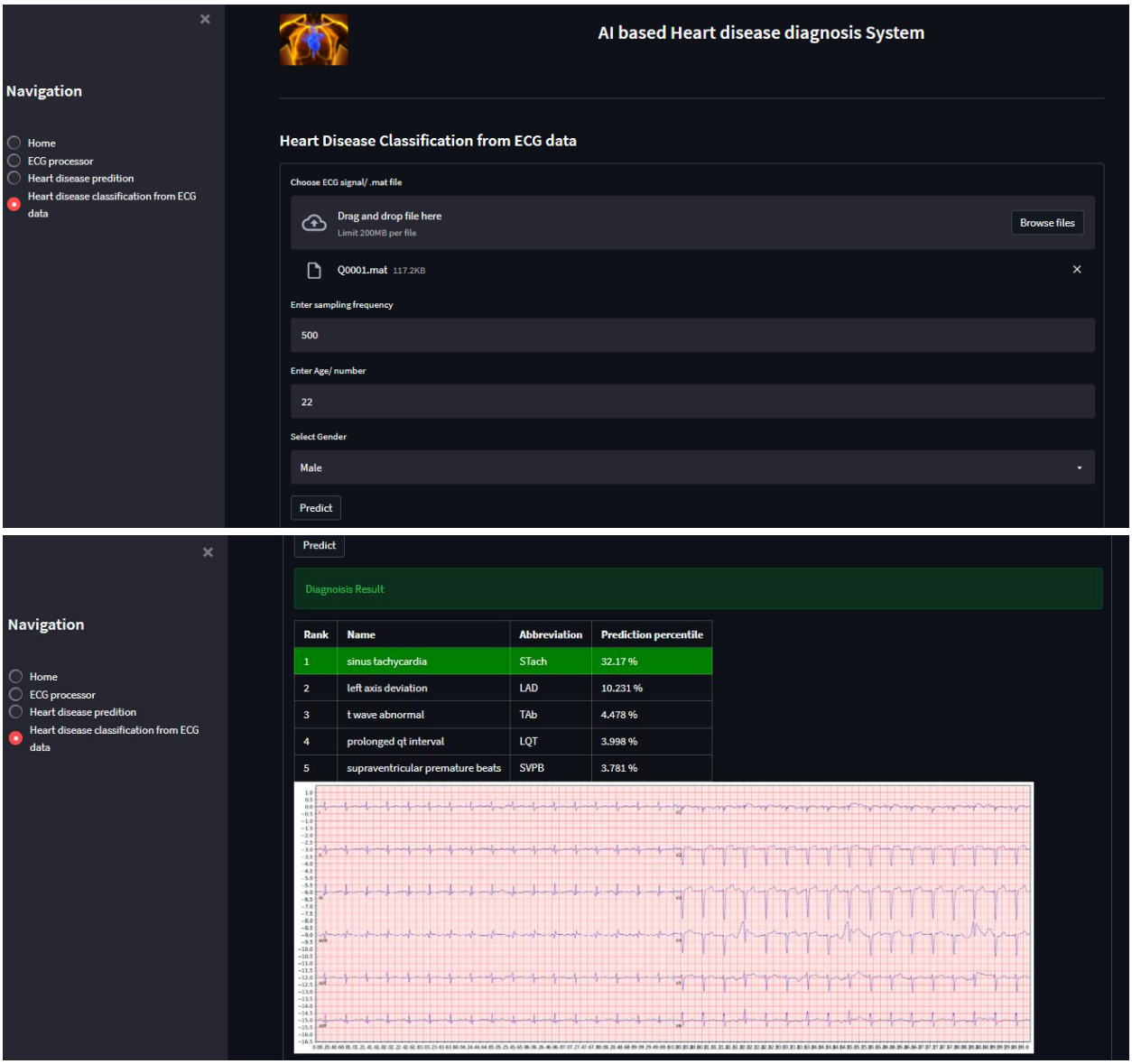


Figure 14: User interface of Heart-disease classification using 12-lead ECG signal predicting a patient with 'sinus tachycardia' cardiac condition.

4. DISCUSSION

Heart diseases are the leading cause of death in the world. They are fatal diseases that are rapidly increasing in both developed and developing countries. The major risk factors of heart disease are behavioral including unhealthy diets, physical inactivity, tobacco use and harmful use of alcohol. The

effects are manifested in terms of increasing blood pressure, blood glucose, blood cholesterol levels and weight (Hajar, 2017). The risk factors can be measured and monitored in primary health facilities. However, for efficient treatment plan different tests including laboratory, imaging or non-invasive techniques are usually required. The traditional methods that are used to diagnose heart disease are manual, complex and error-prone (Allen et al., 2012). Due to the limited availability of medical diagnosing tools and medical experts, specifically in low-resource settings, diagnosis and cure of heart disease are very complex (Yang and Garibaldi, 2015). Using of Artificial Intelligence (AI) based predictive techniques enables auto diagnosis and has the potential to reduce diagnosis errors compared to exclusive human expertise.

To overcome the limitations of traditional manual diagnosis techniques for the identification of heart disease, literatures has attempted to develop different AI based predictive mechanisms using traditional machine learning and deep learning techniques (Detrano et al., 1989; Kahramanli and Allahverdi, 2008; Das et al., 2009; Gudadhe et al., 2010; Methaila et al., 2014; Olaniyi et al., 2015; Patel et al., 2015; Samuel et al., 2017; Haq et al., 2018; Nazir et al., 2018; Tomov and Tomov, 2018; Ali et al., 2019; Khourdifi and Bahaj, 2019; Latha and Jeeva, 2019; Muhammad et al., 2020). Even though the proposed techniques and the results reported are promising, they are designed to serve either a single purpose (e.g., binary classification), or use a limited dataset type, or do not have a potential for translation or application into clinical setting.

The purpose of this work was to design and develop an integrated heart disease diagnosis system that has a flexible application based on the available resources. The developed system was deployed in a user-friendly web-based application that includes three parts: ECG processor, heart disease prediction module and heart disease multiclass classification based on a 12 lead ECG signal module.

In the ECG processor module different algorithms for signal noise removal including removal of high and low frequency noise signal removal, baseline drift correction and power line interference removal have been designed and implemented. After signal pre-processing, a mechanism for ECG feature extraction including R-peak detection, PQST peak detection, ECG waves delineation and quantitative analysis of ECG wave segments has been developed. As demonstrated in Figure 12, the ECG processor module allows users to load single lead ECG signal and perform quantitative analysis of important ECG wave segments for quick diagnosis. ECG is inexpensive, widely affordable, and it is the most useful instrument in the diagnosis and prognosis of heart disease. However, the manual interpretation of ECG signals is complex and exposed to intra- and interobserver variabilities (Allen et al., 2012). The developed system overcomes this challenge by providing an automatic quantitative assessment for informed decision making.

The second module, heart disease prediction system, uses different attribute information including age, sex and patient's clinical data or observations, which are indicators of heart disease, and predict whether the person has heart disease or not. The user interface (Figure 13) allows users fill 12 important attribute information to the system and predict the probability of having heart disease in percentage. The percentile provides information to the patients/experts the likelihood of getting heart disease with the given quantitative values of risk factors. This helps physicians to provide informed decision and perform further diagnosis and the patients to take necessary actions to reduce behavioral risk factors and prevent life threats.

The third module (Figure 14), the 12-lead ECG based multiclass classification, enables users to load a 12-lead ECG signal recorded from suspected heart disease patients and provides predictions of the type of abnormality. It performs multiclass classification to discriminate the ECG signals acquired from those of healthy individuals and patients with existing chronic heart conditions. Currently, 12-lead ECG is a standard method establishing cardiac disorders and used to determine the presence of arrhythmia, conduction defects, ischemia, and signs of structural heart disease (Kirchhoff et al., 2016). The system provides top 5 predictions, among 18 heart conditions, and their probability ranks based on the model's prediction score.

In summary, the system can be used for quick decision making based on the acquired ECG signal, or for prediction purposed based on the patient information and laboratory results, or for multiclassification of cardiac conditions based on a 12-lead ECG record or for all purposes to provide an integrated diagnosis. The proposed system is designed to overcome the challenges of current manual cardiovascular disease diagnosis, providing physicians with reliable support, helping to minimize workload pressure while maximizing efficiency, allowing experts perform informed patient specific diagnosis and treatment decisions. This work can also be used a starting point for further AI based cardiovascular disease diagnosis system developments in the context of clinical adoption of computer aided diagnosis.

5. CONCLUSION

This paper presents an integrated AI-based tool for diagnosis and assessment of cardiac conditions. Different machine learning and deep learning models were trained, evaluated and compared using variety of data collected from different sources, and best performing models were selected and deployed in a custom designed web-based user interface for prediction of heart disease and multiclass classification of cardiac conditions. The developed system can provide a reference for clinical diagnosis, remove the opportunities for human error, saves time and money, and improve the diagnosis ability of clinicians for heart disease enabling timely decision making and treatment planning.

Our experimental results demonstrate that, the developed AI-based heart disease diagnosis system has a potential to improve diagnostic accuracy, and can be used as a decision support system, especially in those areas where both the means of diagnosis and experts are scarce.

DECLARATIONS

Ethics approval and consent to participate

This research did not involve humans, animals, or other subjects. According to Jimma University's institutional review board (IRB), no formal ethics approval was required in this particular case.

Consent for publication

This research did not involve humans, animals, or other subjects

Availability of data and material

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request

Competing interests

The authors declare that they have no competing interests

Authors' contributions

GL and MZ conceptualized, designed, and implemented in collaboration with the co-investigator WB. All authors contributed to the preliminary study, the design, prototyping, and testing. The article was drafted by GL, taking into account the comments and suggestions of the coauthors. All coauthors had the opportunity to comment on the manuscript and approved the final version for publication.

ACKNOWLEDGMENTS

Resources required to conduct the study were provided by the school of Biomedical Engineering and faculty of computing, Jimma Institute of Technology, Jimma University.

REFERENCE

Ali, L., A. Niamat, J. A. Khan, N. A. Golilarz, X. Xingzhong, A. Noor, R. Nour and S. A. C. Bukhari (2019). "An optimized stacked support vector machines based expert system for the effective prediction of heart failure." IEEE Access **7**: 54007-54014.

Allen, L. A., L. W. Stevenson, K. L. Grady, N. E. Goldstein, D. D. Matlock, R. M. Arnold, N. R. Cook, G. M. Felker, G. S. Francis and P. J. Hauptman (2012). "Decision making in advanced heart failure: a scientific statement from the American Heart Association." Circulation **125**(15): 1928-1952.

Bazett, H. C. (1920). "An analysis of the time relations of electrocardiograms." Heart **7**: 353-370.

Breiman, L. (2001). "Random Forests." Machine Learning **45**(1): 5-32.

Bui, A. L., T. B. Horwich and G. C. Fonarow (2011). "Epidemiology and risk profile of heart failure." Nat Rev Cardiol **8**(1): 30-41.

Chen, T. and C. Guestrin (2016). Xgboost: A scalable tree boosting system. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining.

Coca, A., F. H. Messerli, A. Benetos, Q. Zhou, A. Champion, R. M. Cooper-DeHoff and C. J. Pepine (2008). "Predicting stroke risk in hypertensive patients with coronary artery disease: a report from the INVEST." Stroke **39**(2): 343-348.

Das, R., I. Turkoglu and A. Sengur (2009). "Effective diagnosis of heart disease through neural networks ensembles." Expert Systems with Applications **36**(4): 7675-7680.

De Silva, D. A., F. P. Woon, K. T. Moe, C. L. Chen, H. M. Chang and M. C. Wong (2008). "Concomitant coronary artery disease among Asian ischaemic stroke patients." Ann Acad Med Singap **37**(7): 573-575.

Detrano, R., A. Janosi, W. Steinbrunn, M. Pfisterer, J. J. Schmid, S. Sandhu, K. H. Guppy, S. Lee and V. Froelicher (1989). "International application of a new probability algorithm for the diagnosis of coronary artery disease." Am J Cardiol **64**(5): 304-310.

Dua, D. and C. Graff (2019). UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2019.

Gebreyes, Y. F., D. Y. Goshu, T. K. Geletew, T. G. Argefa, T. G. Zemedu, K. A. Lemu, F. C. Waka, A. B. Mengesha, F. S. Degefu, A. D. Deghebo, H. T. Wubie, M. G. Negeri, T. T. Tesema, Y. G. Tessema, M. G. Regassa, G. G. Eba, M. G. Beyene, K. M. Yesu, G. T. Zeleke, Y. T. Mengistu and A. B. Belayneh (2018). "Prevalence of high bloodpressure, hyperglycemia, dyslipidemia, metabolic syndrome and their determinants in Ethiopia: Evidences from the National NCDs STEPS Survey, 2015." PLOS ONE **13**(5): e0194819.

Gudadhe, M., K. K. Wankhade and S. S. Dongre (2010). "Decision support system for heart disease based on support vector machine and Artificial Neural Network." 2010 International Conference on Computer and Communication Technology (ICCCT): 741-745.

Hajar, R. (2017). "Risk factors for coronary artery disease: historical perspectives." Heart views: the official journal of the Gulf Heart Association **18**(3): 109.

Haq, A. U., J. P. Li, M. H. Memon, S. Nazir and R. Sun (2018). "A Hybrid Intelligent System Framework for the Prediction of Heart Disease Using Machine Learning Algorithms." Mobile Information Systems **2018**: 3860146.

Kahramanli, H. and N. Allahverdi (2008). "Design of a hybrid system for the diabetes and heart diseases." Expert systems with applications **35**(1-2): 82-89.

Kaiser, J. and R. Schafer (1980). "On the use of the I 0-sinh window for spectrum analysis." IEEE Transactions on Acoustics, Speech, and Signal Processing **28**(1): 105-107.

Khourdifi, Y. and M. Bahaj (2019). "Heart Disease Prediction and Classification Using Machine Learning Algorithms Optimized by Particle Swarm Optimization and Ant Colony Optimization." International Journal of Intelligent Engineering and Systems.

Kirchhoff, P., S. Benussi, D. Kotecha, A. Ahlsson, D. Atar and B. Casadei (2016). "ESC Guidelines for the management of atrial fibrillation developed in collaboration with EACTS: The Task Force for the management of atrial fibrillation of the European Society of Cardiology (ESC) Developed with the special contribution of the European Heart Rhythm Association (EHRA) of the ESC, endorsed by the European Stroke Organization (ESO)." Eur J Cardiothorac Surg **50**(5): e1-e88.

Latha, C. B. C. and S. C. Jeeva (2019). "Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques." Informatics in Medicine Unlocked **16**: 100203.

Makowski, D., T. Pham, Z. J. Lau, J. C. Brammer, F. Lespinasse, H. Pham, C. Schölzel and S. H. A. Chen (2021). "NeuroKit2: A Python toolbox for neurophysiological signal processing." Behavior Research Methods **53**(4): 1689-1696.

Methaila, A., P. Kansal, H. Arya and P. Kumar (2014). "Early heart disease prediction using data mining techniques." Computer Science & Information Technology Journal **24**: 53-59.

Misganaw, A., D. H. Mariam, A. Ali and T. Araya (2014). "Epidemiology of major non-communicable diseases in Ethiopia: a systematic review." J Health Popul Nutr **32**(1): 1-13.

Muhammad, Y., M. Tahir, M. Hayat and K. T. Chong (2020). "Early and accurate detection and diagnosis of heart disease using intelligent computational model." Scientific Reports **10**(1): 19747.

Nazir, S., S. Shahzad, S. Mahfooz and M. Nazir (2018). "Fuzzy logic based decision support system for component security evaluation." Int. Arab J. Inf. Technol. **15**(2): 224-231.

Olaniyi, E. O., O. K. Oyedotun and K. Adnan (2015). "Heart diseases diagnosis using neural networks arbitration." International Journal of Intelligent Systems and Applications **7**(12): 72.

Patel, J., D. TejalUpadhyay and S. Patel (2015). "Heart disease prediction using machine learning and data mining technique." Heart Disease **7**(1): 129-137.

Samuel, O. W., G. M. Asogbon, A. K. Sangaiah, P. Fang and G. Li (2017). "An integrated decision support system based on ANN and Fuzzy_AHP for heart failure risk prediction." Expert Systems with Applications **68**: 163-172.

Tefera, Y. G., T. M. Abegaz, T. B. Abebe and A. B. Mekuria (2017). "The changing trend of cardiovascular disease and its clinical characteristics in Ethiopia: hospital-based observational study." Vascular health and risk management **13**: 143-151.

Tomov, N.-S. and S. Tomov (2018). "On deep neural networks for detecting heart disease." arXiv preprint arXiv:1808.07168.

WHO. (2021). "Cardiovascular diseases (CVDs)." Retrieved October 19, 2021, from https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

Yang, H. and J. M. Garibaldi (2015). "A hybrid model for automatic identification of risk factors for heart disease." Journal of biomedical informatics **58 Suppl**(Suppl): S171-S182.

Predicting the Level of Anemia among Ethiopian Pregnant Women using Homogeneous Ensemble Machine Learning Algorithm

Belayneh Endalamaw*, Tesfamariam M. Abuhay, Dawit Shibabaw

Information science, University of Gondar, Gonda, Ethiopia
**Corresponding author, e-mail: belayneh.endalamaw@uog.edu.et*

ABSTRACT

Anemia is defined as the reduction of red blood cells (RBC) or hemoglobin in the blood. More than 115,000 maternal deaths and 591,000 prenatal deaths occurred in the world per year with anemia. The World Health Organization (WHO) divides anemia in pregnancy into mild anemia (Hb 10- 10.9g/dl), moderate anemia (Hb 7.0-9.9g/dl), and severe anemia (Hb < 7g/dl). This study, hence, aims to predict the level of anemia among pregnant women in the case of Ethiopia using homogeneous ensemble machine learning algorithms. In this study, the data were gathered from the Ethiopian demographic, health survey (EDHS) which was collected three times with five-year intervals. The data were preprocessed to get quality data that are suitable for the machine learning algorithm to develop a model that predicts the levels of anemia among pregnant. The study was conducted following a design science approach. Random forest, cat boost, and extreme gradient boosting with class decomposition (one versus one and one versus rest) and without class decomposition were employed to build the predictive model. For constructing the proposed model, twelve experiments were conducted with a total of 29104 instances with 23 features, and a training and testing dataset split ratio of 80/20. The overall accuracy of random forest, extreme gradient boosting, and cat boost without class decompositions are 91.34%, 94.26%, and 97.08.90%, respectively. The overall accuracy of random forest, extreme gradient boosting, and cat boost with one versus one are 94.4%, 95.21%, and 97.44%, respectively. The overall accuracy of random forest, extreme gradient boosting, and cat boost with one versus the rest are 94.4%, 94.54%, and 97.6%, respectively. Finally, the researcher decided to use cat boost algorithms with one versus the rest for further use in the development of artifacts, model deployment, risk factor analysis, and generating rules because it has registered better performance with 97.6% accuracy. We identified the most determinant risk factors using feature importance. Some of them are the duration of the current pregnancy, age in 5-year groups, source of drinking water, respondent's occupation, number of household members, wealth index, husband/partner's education level, birth history.

Keywords: *Anemia, Homogeneous Ensemble Machine learning, Flask, Heroku, Class Decomposition*

1. INTRODUCTION

According to [1], Anemia is defined as a decrease in the number of RBC or hemoglobin in the blood that has significant adverse health consequences, as well as adversative impacts on economic and social development. According to [2], Anemia is a public health problem among women of reproductive age, affecting both poor and rich countries overall the world. It negatively affects the social and economic well-being of a country and all its communities. According to [3][4], Anemia during pregnancy is a risk factor for poor pregnancy outcomes, such as low birth weight (LBW), preterm birth, prematurity stillbirth, intrauterine growth restriction, and impaired cognitive development.

Anemia in pregnant women can be caused by parasitic infestation, socio-demographic status, economic status, dietary practice, obstetric factors, reproductive health, and other health-related factors [5]. More than

115,000 maternal deaths and 591,000 prenatal deaths are caused by anemia disease in the world per year [6]. According to a WHO report, anemia affects 41.8 % of pregnant women worldwide, with Africa (57.1 percent) having the highest prevalence [7][8]. According to [4][9], anemia during pregnancy is the main cause of morbidity and mortality of pregnant women in developing countries like Ethiopia and has both maternal and fetal consequences such as impairment of the capacity of the blood to transport oxygen around the body, fatigue, poor work capacity, impaired immune function, increased risk of cardiac diseases, and mortality [4][10]. The burden and underlying factors of these diseases are varied even within countries [10]. Most of the women who live in the rural area of Ethiopia have been affected by this disease due to different factors like nutritional factors, parasites, socio-demographic factors, obstetric factors, reproductive characteristics, and the like [10]. According to WHO guidelines, the minimum acceptable hemoglobin level during pregnancy is 11 g/dl, during the first half, 10.5 g/dl, during the second half, and 12 g/dl for lactating women [6][10][11]. To understand and predict the level of anemia among pregnant in the case of Ethiopia, and factors that influence anemia among pregnant women, several types of research have been conducted in the world done by health care professionals. For example, [3][6][7][8][9][10][11][12][13] and [14] investigated the status of anemia among pregnant women using cross-sectional statistical methods. They also used bivariate and multivariate logistic regression methods and identified the most determinant risk factors. Most of these previous studies, however, used local clinical data that covers limited geographical areas like a single city or town only, small data set less than 500, and only focused on one of the following factors such as socioeconomic, demographic, nutritional, and reproductive, apart from health-related variables. Some of them [3][6][7][8][9][10][11][12][13] and [14] also focused on identifying the determinant risk factors of anemia among pregnant women who followed first antenatal care during pregnancy only and develop a descriptive statistics model. Besides, [3][6][7][8][9][10][11][12][13] and [14] of the previous studies were conducted using cross-sectional statistical methods which usually have limited capacity to discover new and unanticipated patterns that are hidden in data and identify cause and effect relationships [6][10][15]. These studies did not also include features that lead to anemia like the previous history of birth, previous history of abortion, history of the place of delivery, history of malaria, and nutritional variables, i.e. the factors that contribute to the occurrence of anemia among pregnant women weren't thoroughly studied. In such situations, new technologies like machine learning algorithms may help to discover hidden patterns [16]. There were machine learning-related works such as [17][18][19] and [20]. These studies aimed at developing a predictive model, but did not identify the most determinant risk factors, and generate rules that allow to development of evidence-based policies and strategies towards reducing anemia among pregnant women. This study, hence, aims to develop a model that predicts the level of anemia among pregnant women using homogeneous ensemble machine learning algorithms by investigating the following research questions: (1) what is the underlying structure of anemia among pregnant women in Ethiopia? (2) Which homogeneous ensemble of machine learning algorithms is suitable for predicting the level of anemia among pregnant women in Ethiopia? (3) What are the associated risk factors that influence the occurrence of anemia among pregnant women in Ethiopia? (4) What are the important rules that may shape policies and interventions towards reducing anemia among pregnant women in Ethiopia?

The rest of this document is organized as follows: Section II presents related works, Section III discusses materials and methods used, Section IV mentions experimental setup and result discussion, and Section V presents the conclusion.

2. RELATED WORK

Several studies such as [3][6][7][8][9][10][11][12][13] and [14] investigated the status of anemia among pregnant women and its determinant factors in different parts of Ethiopia using cross-sectional statistical methods. They used bivariate and multivariate logistic regression methods. Most of these previous studies used local clinical data, covered limited geographical areas like a single city or town only, employed small data set less than 500, and focused on different factors like socioeconomic, demographic, nutritional, and reproductive, apart from health-related variables. Some of them also identified the determinant risk factors of anemia among pregnant women who followed first antenatal care during pregnancy. Several studies were conducted using cross-sectional statistical methods which usually have limited capacity to discover new and unanticipated patterns and identify cause and effect relationships that are hidden in data [10][6][15]. These studies did not include features, such as the previous history of birth, previous history of abortion, history of the place of delivery, history of malaria, and nutritional variables, i.e. the factors that contribute to the occurrence of anemia among pregnant women weren't thoroughly studied. Dithy and Krishnapriya [17] predicted anemia among pregnant women using ANN and gausnominal classification algorithm with an accuracy of 065 % and 0.74%, respectively. M. D. Dithy and V. Krishnapriya [18] focuses on anemia selection in pregnant women by using random prediction (Rp) classification algorithm. The researcher were classified the anemia level by those classification algorithms and the performances of the predictive model shows that 0.65%, 0.76%, 0.826%, and 0.92% with ANN, gausnominal, vector neighbor, and random prediction model respectively. Besides, these studies did not considered all potential features, discussed in section I, which helps to take holistic interventions. [17][18][19] and [20] aimed to construct a predictive model, but they did not identify risk factors, and extract rules which are important to make evidence-based policies and interventions. This study, hence, motivated to fill these gaps by constructing a predictive model, identifying risk factors, extracting relevant rules towards preventing and controlling the level of anemia among pregnant women in Ethiopia, designing an innovative artifact and deploying the predictive model for the potential users.

3. MATERIALS AND METHODS

3.1.Data Collection

The data used in this research was extracted from the EDHS of the Ethiopian central statistical agency. The data were collected in 2005, 2011, and 2016, in the five-year interval.

3.2. Data Preprocessing

The extracted datasets consist of a total of 11174 instances with 34 features. As all these features are not relevant for developing a predictive model that can predict the level of anemia among pregnant women in the case of Ethiopia, data preprocessing techniques such as data cleaning, data transformation, handling class imbalance, removal of quasi constant features, and feature selection methods were applied. The missing values were handled using mode imputation techniques for categorical data. Redundant data were

removed manually. The quasi constant features were not directly removed, but we have constructed one feature and combined them into one. There were features which have more distinct values and need to be transformed for mining purposes; such as features with more categorical values such as the source of drinking water, body mass index, wealth index, marital status, and household members were transformed into discrete values using binning discretization mechanisms. Then, features selection methods such as filter and wrapper were applied to select the relevant features which are important for further process. The class level of the collected data was imbalanced which was treated using the synthetic minority over-sampling technique (SMOTE). The main reason for using SMOTE is it avoids loss of valuable information [21][22]. After conducting all the required data preprocessing tasks, a total of 29104 instances with 23 features were considered for further analysis and prediction model development. Feature selection is a technique for selecting a small subset of relevant features from a large set of relevant features by deleting unnecessary, redundant, or noisy features [23]. In this experiment, we used two types of feature selection methods (filter, and wrapper) to see which one could give us better performance. As a result, the step forward feature selection method performs better than others, see Table 1. So we used all the features selected by step forward feature selection methods.

For developing the final predictive model we have used all the features selected by step forward feature selection methods, see Table 1, and all the features that were recommended by domain experts, see Table 2.

The performance of the algorithm highly depends on the selection of Hyperparameter, which has always been a crucial step in the process of machine learning model development [24][25][26]. To this end, grid search was used to tune the Hyperparameter of each algorithm that was selected for an experiment.

3.3. Predictive model Development

To construct a model that predicts the level of anemia among pregnant women in the case of Ethiopia, homogeneous ensemble machine learning algorithms such as extreme gradient boosting, random forest, and cat boost algorithms without applying class decomposition and with applying one versus one and one versus rest class decomposition were selected for an experiment. To show that homogeneous ensemble algorithms can perform better than other supervised machine learning algorithms, we have also conducted using decision tree algorithms. The data set was split into 80/20 train-test datasets. The performance of each predictive model was evaluated using accuracy, precision, recall, and F1- score.

Table 1: Feature selection results

	Mutual information feature selection	Chi2 feature selection	F class if feature selection	Step forward feature selection	Step backward feature selection
0	Age in 5-year groups	Region	Region	Age in 5-year groups	Age in 5-year groups
1	Region	Highest educational level	Type of place of residence	Region	Region
2	Number of antenatal care visits	Source of drinking water	Highest educational level	Number of antenatal care visits	Number of antenatal care visits
3	Highest educational level	Religion	Source of drinking water	Source of drinking water	Highest educational level

4	Religion	Frequency of reading newspaper or magazine	Religion	Religion	Source of drinking water
5	Frequency of watching television	Frequency of listening to radio	Frequency of watching television	Number of household members	Religion
6	Duration of current pregnancy	Frequency of watching television	Duration of current pregnancy	Frequency of listening to radio	Number of household members
7	Birth history	Currently breastfeeding	Current pregnancy wanted	Duration of current pregnancy	Frequency of listening to radio
8	History of contraceptive use	Mosquito bed net	History of contraceptive use	birth history	Duration of current pregnancy
9	Body mass index	Husband/partner's education level	Husband/partner's education level	Current pregnancy wanted	birth history
10	Husband/partner's education level	Respondent's occupation	Respondent's occupation	History of contraceptive use	Current pregnancy wanted
11	Husband/partner's occupation	History of the place of delivery	History of the place of delivery	Body mass index	Body mass index
12	Respondent's occupation	Iron tablet during pregnancy	Iron tablet during pregnancy	Husband/partner's education level	Husband/partner's education level
13	History of the place of delivery	Had diarrhea recently	Had diarrhea recently	Husband/partner's occupation	Husband/partner's occupation
14	Vitamin a in last 6 months	Vitamin a in last 6 months	Vitamin a in last 6 months	Respondent's occupation	Respondent's occupation
15	Wealth index combined	Wealth index combined	Wealth index combined	Wealth index combined	Wealth index combined
Accuracy with RF	89.091221	76.120941	82.85518	0.91813755	0.917751321

Table 2: Features selected with domain experts

No	Features	Feature descriptions
1	m49a	Take the drug for malaria during pregnancy
2	H34	Take Vitamin A
3	V106	Highest educational level
4	M15	History of Place of delivery
5	m45	Iron tablet during pregnancy
6	V228	History of terminating a pregnancy
7	V404	Breastfeeding status

Figure 1 represents the proposed model architecture that was implemented in this study to develop a predictive model, select the best-performed model, identify risk factors, generate relevant rules, design artifacts, and deploy the final model for the potential set of users.

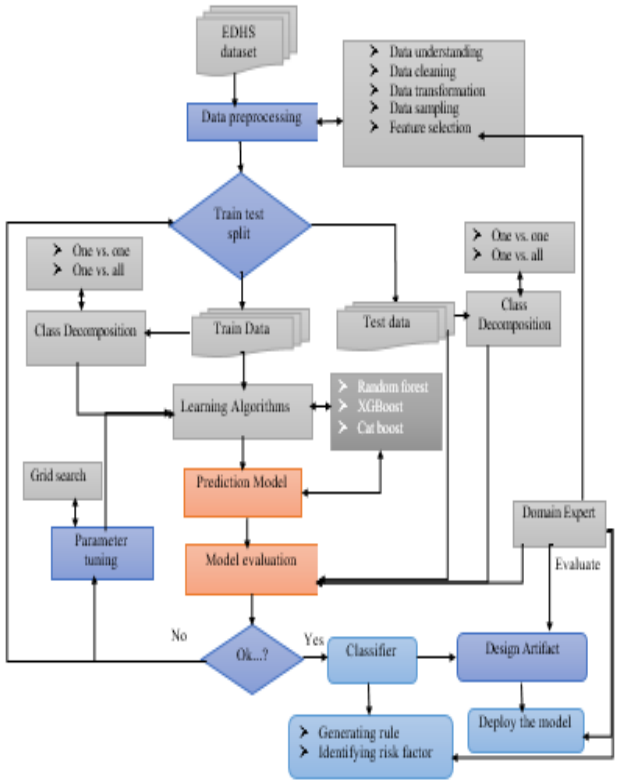


Figure 1: Proposed model architecture

4. EXPERIMENTAL SETUP AND RESULTS DISCUSSION

Here below results are discussed based on the research questions.

4.1. What is the underlying structure of anemia among pregnant women in Ethiopia?

To answer this question, we used descriptive statistics techniques to show the underlying structure of anemia among pregnant in the case of Ethiopia by considering the year, age, place of residence, region, antenatal care visit, history of the place of delivery, history of terminating the pregnancy, and wealth index with the anemia level. See graph 2 below which represents that pregnant women who live in the rural area of Ethiopia are highly affected by anemia.

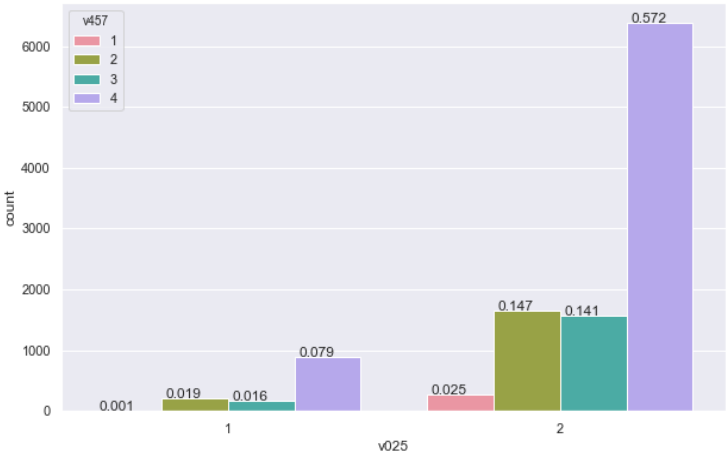


Figure 2: Prevalence of Anemia in place of residence

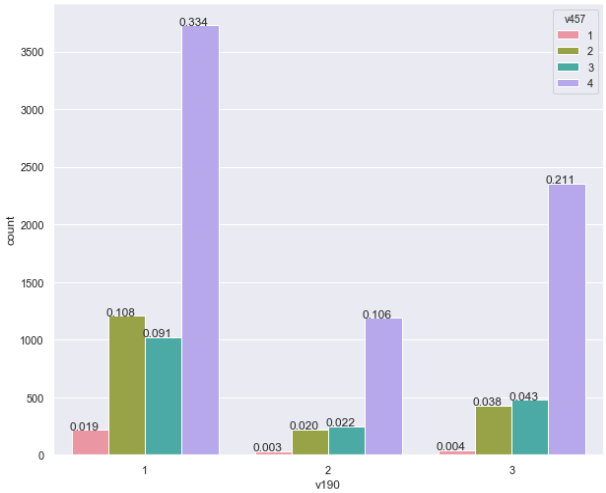


Figure 3: Prevalence of anemia in wealth index status

Graph 4 here below represents that the pregnant women which didn't follow or follow one time only during pregnancy were highly affected by anemia.

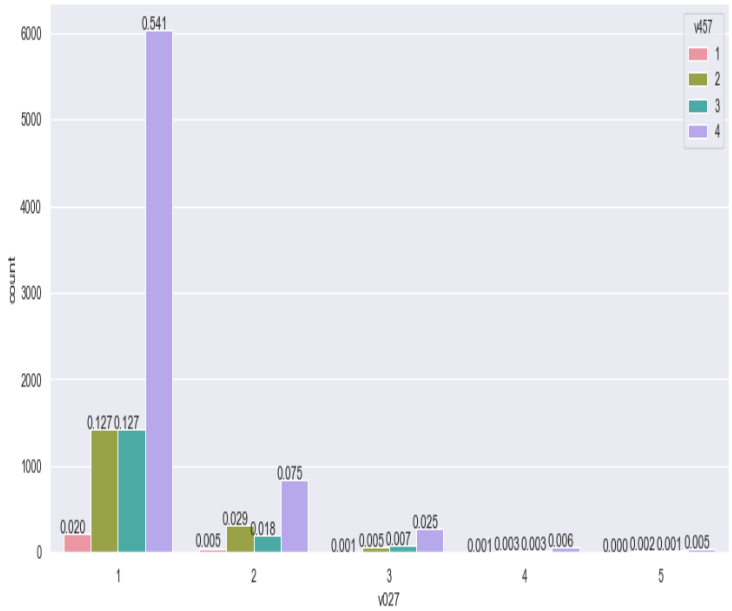


Figure 4: prevalence of anemia among pregnant women Based on antenatal care follow-up

4.2. Which homogeneous ensemble of machine learning algorithms is suitable for predicting the level of anemia among pregnant women in Ethiopia?

To answer this question, twelve experiments for three homogeneous ensemble machine learning algorithms namely random forest, extreme gradient boosting, and cat boost with class decomposition (by using one versus one and one versus rest), and without class decomposition was conducted. To show that homogeneous ensemble algorithms can perform better than other supervised machine learning algorithms, we have also conducted an experiment using decision tree algorithms. The experiments showed that the model that was developed using the cat boost algorithm with one versus rest class decomposition performs better in predicting the level of anemia among pregnant in the case of Ethiopia with 97.6% of accuracy, 97.59% of precision, 97.57% of recall, and 97.58% of f1_score, see Table 3 below.

4.3.What are the associated risk factors that influence the occurrence of anemia among pregnant women in the case of Ethiopia?

To answer this question, feature importance analysis was performed using the model that was developed with the best performing algorithm which is cat boost.

4.4. What are the important rules that can be generated from the predictive model?

To answer this question, we used all the features that we used to develop the predictive model and generate all the important rules by using the best-performed algorithms (cat boost algorithms with one versus rest class decompositions) for the level of anemia among pregnant in the case of Ethiopia. The most important rules that were also validated by domain experts are presented here below:

Table 3: Model performance

	Evaluation metrics	Without class decompositions	With one vs. one class decomposition	With one vs. rest class decomposition
Decision tree	Accuracy	79.38%	89.88%	89.09%
	precision	79.09%	89.81%	89.01%
	Recall	79.21%	89.77%	88.98%
	F1_score	79.03%	89.71%	88.96%
	Cross validation	68.48%	84.27%	83.17%
Random forest	Accuracy	91.34%	94.4%	94.4%
	Precision	91.32%	94.36%	94.37%
	Recall	91.28%	94.35%	94.35%
	F1_score	91.25%	94.34%	94.34%
	Cross validation	81.23%	89.37%	88.18%
Cat Boost	Accuracy	97.08%	97.44%	**97.595%**
	Precision	97.09%	97.438%	**97.596%**
	Recall	97.05%	97.418%	**97.574%**
	F1_score	97.06%	97.422%	**97.58%**
	Cross validation	95.94%	96.478%	**96.482%**
Extreme gradient Boost	Accuracy	94.26%	95.21%	94.54%
	Precision	94.27%	95.20%	94.53%
	Recall	94.20%	95.16%	94.48%
	F1_score	94.20%	95.16%	94.48%
	Cross validation	88.86%	91.73%	89.72%

Table 4: Identified risk factors with best fit model and feature importance

Feature	Values	Feature	Values
Duration of current pregnancy	10.3953193	Current pregnancy wanted	3.838873474
Age in 5-year groups	9.69394377	Body mass index	2.787116569
Source of drinking water	8.99369175	Number of ANC visits	2.600944933
History of contraceptive use	6.61405164	Highest educational level	2.419310637
Respondent's occupation	6.12946203	History of terminating a pregnancy	0.849814164
Number of household members	5.85914199	Currently breastfeeding	0.732357678
Wealth index	5.63211101	Type of place of residence	0.576997215
Frequency of listening to the radio	5.16045505	Vitamin A in last 6 months	0.356953114
Husband/partner's education level	5.02943094	During pregnancy, given or bought iron tablets/syrup	0.046775106
Region	4.3314029	History of Place of delivery	0.010932682
Husband/partner's occupation	3.96855455	During pregnancy took: sp/ fansidar for malaria	0.00058328
Birth history	3.87177534		

RULE1, IF given iron tablet or syrup during pregnancy == 'No' AND vitamin A in last 6 months == 'No' AND during pregnancy took sp fansidar for malaria== 'No' AND region == 'Somali' AND currently breastfeeding == 'No' AND place of residence == 'rural' AND Duration of current pregnancy == 'seven-nine-week' AND current pregnancy wanted == 'Yes' AND respondents occupation == 'did not work' AND history of place of delivery == 'Home' AND age == 'thirty - thirty four' AND educational level == 'no

education' AND husband educational level == 'no education' AND number of household== 'six-ten' AND history of terminating pregnancy== 'No' AND body mass index == 'normal' AND husband occupation == 'did not work' THEN anemia level== 'sever'.

RULE2, IF given iron tablet or syrup during pregnancy == 'No' AND vitamin A in last 6 months == 'No' AND during pregnancy took sp fansidar for malaria== 'No' AND region == 'Somali' AND currently breastfeeding == 'No' AND place of residence == 'rural' AND Duration of current pregnancy == 'seven-nine-week' AND current pregnancy wanted == 'Yes' AND respondents occupation == 'did not work' AND place of delivery == 'Home' AND age == 'thirty - thirty four' AND educational level == 'no education' AND husband educational level == 'no education' AND number of household== 'six-ten' AND History of terminating pregnancy== 'No' AND body mass index == 'normal' AND husband occupation == ' agricultural - employee' AND source of water == 'pure' AND history of contraceptive use == 'Yes' THEN anemia level== 'none anemic'.

RULE3, IF given iron tablet or syrup during pregnancy == 'No' AND vitamin A in last 6 months == 'No' AND during pregnancy took sp fansidar for malaria== 'No' AND region == 'Somali' AND currently breastfeeding == 'No' AND place of residence == 'rural' AND Duration of current pregnancy == 'seven-nine-week' AND current pregnancy wanted == 'Yes' AND respondents occupation == 'did not work' AND history of place of delivery == 'Home' AND age == 'thirty - thirty four' AND educational level == 'no education' AND husband educational level == 'no education' AND number of household== 'six-ten' AND history of terminating pregnancy== 'No' AND body mass index == 'normal' AND husband occupation == ' agricultural - employee' AND source of water == 'not pure' AND history of contraceptive use == 'Yes' THEN anemia level== 'Moderate'.

After conducting all twelve experiments and selecting the best-performed model, we have designed innovative artifacts using a flask framework with HTML and deployed the predictive model for the potential users. The artifacts were designed by using all the features that we used for model development. The developed artifacts were simply an interface designed by HTML which takes the results of the predictive model with the help of the Flask framework. The designed artifacts were deployed using Heroku-based cloud computing platforms for the potential users. All the potential users can access the results of the predictive model to evaluate the level of anemia among pregnant women. See the link here below and potential users can access it anywhere over the internet.

https://anemia-level-prediction-model.herokuapp.com/

5. CONCLUSION

Anemia is a global public health issue that affects a wide range of people of all ages. Anemia during pregnancy is a risk factor for poor pregnancy outcomes, such as low birth weight, preterm birth, prematurity stillbirth, intrauterine growth restriction, and impaired cognitive development. This study aimed to develop a predictive model for the level of anemia among pregnant in the case of Ethiopia by using homogeneous ensemble machine learning algorithms. This study was conducted by using design science methodology. The proposed model was constructed using homogeneous ensemble machine learning algorithms namely random forest, extreme gradient boosting, and cat boost algorithms with class decomposition methods and

without class decomposition methods. To conduct this study we have done a total of twelve experiments. The cat boost algorithm with one versus all class decomposition has registered the highest performance with 97.6% of accuracy, 97.59% of precision, 97.57% of recall, 97.58% of f1_score, and 96.48% of cross-validation. We have identified the best determinant risk factors with the best-performed algorithms and feature importance analysis methods. Some of the most determinant risk factors were duration of current pregnancy, age in five years group, source of drinking water, history of contraceptive use, respondent's occupation, and several household members. We have also designed artifacts using HTML as a front end and Flask framework for the back end that takes the predictive model. The researcher generates the most important rules by using the best fit model for developing policies and interventions towards maintaining anemia among pregnant women.

Finally, we recommend that future researchers conduct a predictive model for pregnant women that predicts which type of anemia is occurred within the pregnant women either Vitamin deficiency anemia, Anemia of inflammation, Aplastic anemia, or iron-deficiency anemia. A predictive model that can predict the level of anemia among neonatal based on maternal determinants during pregnancy. The determinant risk factors over time.

ACKNOWLEDGMENT

We would like to acknowledge the Ethiopian central statistics for providing us the data with a data set description that was used to conduct this study.

REFERENCES

[1] A. R. Kavsaoɨlu, K. Polat, and M. Hariharan, "Non-invasive prediction of hemoglobin level using machine learning techniques with the PPG signal's characteristics features," *Appl. Soft Comput. J.*, vol. 37, pp. 983–991, 2015, doi: 10.1016/j.asoc.2015.04.008.

[2] F. Habyarimana, T. Zewotir, and S. Ramroop, "Prevalence and risk factors associated with anemia among women of childbearing age in Rwanda," *Afr. J. Reprod. Health*, vol. 24, no. 2, pp. 141–151, 2020, doi: 10.29063/ajrh2020/v24i2.14.

[3] W. Worku Takele, A. Tariku, F. Wagnew Shiferaw, A. Demsie, W. G. Alemu, and D. Zelalem Anlay, "Anemia among Women Attending Antenatal Care at the University of Gondar Comprehensive Specialized Referral Hospital, Northwest Ethiopia, 2017," *Anemia*, vol. 2018, 2018, doi: 10.1155/2018/7618959.

[4] G. Stephen, M. Mgongo, T. H. Hashim, J. Katanga, B. Stray-pedersen, and S. E. Msuya, "Anaemia in Pregnancy : Prevalence , Risk Factors , and Adverse Perinatal Outcomes in Northern Tanzania," vol. 2018, 2018.

[5] S. K. Ndegwa and S. K. Ndegwa, "Anemia & Its Associated Factors Among Pregnant Women Attending Antenatal Clinic At Mbagathi County Hospital , Nairobi County , Kenya," vol. 32, no. 1, pp. 59–73, 2019.

[6] W. Gari, A. Tsegaye, and T. Ketema, "Magnitude of anemia and its associated factors among pregnant women attending antenatal care at Najo General Hospital, northwest Ethiopia," *Anemia*, vol. 2020, pp. 1–8, 2020, doi: 10.1155/2020/8851997.

[7] T. A. Gudeta, T. M. Regassa, and A. S. Belay, "Magnitude and factors associated with anemia among pregnant women attending antenatal care in Bench Maji, Keffa and Sheka zones of public hospitals, Southwest, Ethiopia, 2018: A cross -sectional study," *PLoS One*, vol. 14, no. 11, pp. 30–34, 2019, doi: 10.1371/journal.pone.0225148.

[8] A. Gebreweld and A. Tsegaye, "Prevalence and Factors Associated with Anemia among Pregnant Women

Attending Antenatal Clinic at St. Paul's Hospital Millennium Medical College, Addis Ababa, Ethiopia," *Adv. Hematol.*, vol. 2018, 2018, doi: 10.1155/2018/3942301.

[9] M. S. Teshome, D. H. Meskel, and B. Wondafrash, "Determinants of anemia among pregnant women attending antenatal care clinic at public health facilities in kacha birra district, southern ethiopia," *J. Multidiscip. Healthc.*, vol. 13, pp. 1007–1015, 2020, doi: 10.2147/JMDH.S259882.

[10] B. Zekarias, A. Meleko, A. Hayder, A. Nigatu, and T. Yetagessu, "Prevalence of Anemia and its Associated Factors among Pregnant Women Attending Antenatal Care (ANC) In Mizan Tepi University Teaching Hospital, South West Ethiopia," *Heal. Sci. J.*, vol. 11, no. 5, pp. 1–8, 2017, doi: 10.21767/1791-809x.1000529.

[11] F. Weldekidan, M. Kote, M. Girma, N. Boti, and T. Gultie, "Determinants of Anemia among Pregnant Women Attending Antenatal Clinic in Public Health Facilities at Durame Town : Unmatched Case Control Study," vol. 2018, 2018.

[12] M. O. Osman, T. Y. Nour, H. M. Bashir, A. K. Roble, A. M. Nur, and A. O. Abdilahi, "Risk factors for anemia among pregnant women attending the antenatal care unit in selected jigjiga public health facilities, somali region, east ethiopia 2019: Unmatched case–control study," *J. Multidiscip. Healthc.*, vol. 13, pp. 769–777, 2020, doi: 10.2147/JMDH.S260398.

[13] B. Berhe, F. Mardu, H. Legese, A. Gebrewahd, G. Gebremariam, and K. Tesfay, "Prevalence of anemia and associated factors among pregnant women in Adigrat General," *BMC Res. Notes*, pp. 1–6, 2019, doi: 10.1186/s13104-019-4347-4.

[14] D. Getaneh, A. Bayeh, B. Belay, T. Tsehaye, and Z. Mekonnen, "Assessment of the Prevalence of Anemia and Its Associated Factors among Pregnant Women in Bahir Dar City Administration, North-West Ethiopia," *J. Pregnancy Child Heal.*, vol. 05, no. 02, 2018, doi: 10.4172/2376-127x.1000367.

[15] R. C. Solem, "Limitation of a cross-sectional study," *Am. J. Orthod. Dentofac. Orthop.*, vol. 148, no. 2, p. 205, 2015, doi: 10.1016/j.ajodo.2015.05.006.

[16] A. M. Abaidullah, N. Ahmed, and E. Ali, "Identifying Hidden Patterns in Students' Feedback through Cluster Analysis," *Int. J. Comput. Theory Eng.*, vol. 7, no. 1, pp. 16–20, 2014, doi: 10.7763/ijcte.2015.v7.923.

[17] M. D. Dithy and V. Krishnapriya, "Predicting Anemia in Pregnant Women By Using Gausnominal," vol. 118, no. 20, pp. 3343–3349, 2018.

[18] M. D. Dithy and V. Krishnapriya, "Anemia selection in pregnant women by using random prediction (Rp) classification algorithm," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2, pp. 2623–2630, 2019, doi: 10.35940/ijrte.B3016.078219.

[19] S. S. Yadav and S. M. Jadhav, "Machine learning algorithms for disease prediction using Iot environment," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6, pp. 4303–4307, 2019, doi: 10.35940/ijeat.F8914.088619.

[20] P. Anand, R. Gupta, and A. Sharma, "Prediction of Anaemia among children using Machine Learning Algorithms," no. June, pp. 469–480, 2020.

[21] I. Journal and C. Science, "Class Imbalance Problem in Data Mining : Review," vol. 2, no. 1, 2013.

[22] R. P. Ribeiro, "SMOTE for Regression," no. October 2015, 2013, doi: 10.1007/978-3-642-40669-0.

[23] S. Wang, J. Tang, H. Liu, and E. Lansing, "Encyclopedia of Machine Learning and Data Mining," *Encycl. Mach. Learn. Data Min.*, pp. 1–9, 2016, doi: 10.1007/978-1-4899-7502-7.

[24] M. J. Healy, "Statistics from the inside. 15. Multiple regression (1).," *Arch. Dis. Child.*, vol. 73, no. 2, pp. 177–181, 1995, doi: 10.1136/adc.73.2.177.

[25] R. G. Mantovani, A. L. D. Rossi, E. Alcobaça, J. C. Gertrudes, S. B. Junior, and A. C. P. de L. F. de Carvalho, "Rethinking Defaults Values: a Low Cost and Efficient Strategy to Define Hyperparameters," 2020, [Online].

Available: http://arxiv.org/abs/2008.00025.

[26] m. M. Ramadhan, i. S. Sitanggang, f. R. Nasution, and a. Ghifari, "Parameter Tuning in Random Forest Based on Grid Search Method for Gender Classification Based on Voice Frequency," *DEStech Trans. Comput. Sci. Eng.*, no. cece, 2017, doi: 10.12783/dtcse/cece2017/14611.

Predicting Perinatal Mortality Based on Maternal Health Status and Health Insurance Service using Homogeneous Ensemble Machine Learning Methods and Deploy model

Dawit Shibabaw*, Tesfamariam M Abuhay, Belayneh Endalamaw

Dept. of Information Science, University of Gondar, Gondar, Ethiopia
**Corresponding author, e-mail: dawitshibabaw14@gmail.com*

ABSTRACT

Perinatal mortality in Ethiopia is highest in Africa, with 68 per 1000 pregnancies Intrapartum deaths (death during the delivery). It is mainly attributed to home delivery, which accounts for more than 75% of the perinatal deaths. Financial constraints have a significant impact on timely access to maternal health (MH) care. As a result, financial incentives, such as health insurance, can address the demand- and supply-side factors. This study, hence, aims to predict perinatal mortality based on maternal health status and health insurance service using homogeneous ensemble machine learning methods. The data was collected from Ethiopian demographic health survey from 2011 to 2019 G.C. The data were pre-processed to get quality data that are suitable for a machine-learning algorithm to develop a model that predicts perinatal mortality. For constructing the proposed model, three experiments were conducted with a 80/20 training and testing dataset split ratio. Random forest, gradient boosting, and cat boost algorithms were selected for experiment. The overall accuracy of random forest, gradient boosting, and cat boost with 17 features scored an accuracy of 89.95%, 90.24%, and 82%, respectively. We found out that perinatal mortality in Ethiopia is associated with risk factors such as mother's educational level, residence, mother age, wealth status, distance to the health facility, preterm, smoke cigarette, anemia level, haemoglobin level, community-based health insurance, and marital status.

Keywords: *Perinatal mortality, Homogenous ensembles, Machine learning, Health insurance, Insurance, Maternal health*

1. INTRODUCTION

Perinatal mortality refers to a fatal death at or after 28 weeks of pregnancy (stillbirth) and includes death within 7 days of life after birth [1][2]. According to the World Health Organization (WHO) 2019 report, there were 2.6 million newborn infants globally, but more than 8200 died within a day [3]. Among the 133 million newborn infants alive each year, 2.8 million died in the first week of life after birth/at birth, and the majority occurred in low-income level countries [3]. Given the reaching deadlines for reaching the Millennium Development Goals, the international community supports low- and middle-level income countries to renew their commitment to reducing maternal and infants mortality rates by improving access to maternal, neonatal, and perinatal health services [4].

Over 100 million individuals pay out-of-pocket (OOP) payments to get health treatments that have proven difficult to obtain for millions of poor people, resulting in increased morbidity and mortality [5]. WHO recommends community-based health insurance (CBHI) as one of the approaches for reducing OOP expenditures for registered families which, in turn, reduce mobility and mortality [6]. The association of CBHI with reduced maternal and infant mortality was apparent but it is impossible to reduce the infant mortality rate, without reducing the perinatal mortality [7].

Perinatal mortality in Ethiopia is the highest in Africa, with 68 per 1000 pregnancies Intrapartum deaths (death during the delivery) [8]. Ethiopia shared and valued the Sustainable Development Goals (SDGs) and has been trying to achieve the target of reducing neonatal mortality to below 12 per 1000 live births, by 2030 [9]. However, reduction of neonatal, infant and under-five mortalities was not realized without substantial reduction of perinatal mortality [10]. It is mostly attributed to home deliveries, which account for more than 75% of all perinatal deaths due to the lack of awareness about health insurance services during birth, and it continued to be an essential part of the third sustainable development goal which aims to end preventable children's deaths by 2030 [9].

Financial constraints have a significant impact on timely access to maternal health (MH) care, such as Antenatal Care (ANC), skilled care at delivery, access to facility-based deliveries, postnatal care (PNC), and perinatal [7]. As a result, financial incentives, such as health insurance, can address the demand- and supply factors that may possibly impacting maternal, neonatal, and perinatal health results [11].

The Ethiopian Ministry of Health has been working for years to make health services accessible for women through community and facility-based interventions to increase survival of newborn and children [9]. Despite these interventions, perinatal death remains an issue in Ethiopia, in particular; home delivery remains the challenge to reduce perinatal mortality [11]. Still, 74% of women give birth outside health institutions without skilled care attendants in Ethiopia [8] [12] [13]. This study, hence, aims to develop a model that predicts perinatal mortality in Ethiopia using homogeneous ensemble machine learning algorithms by investigating the following research questions (1) what is the underline structure and evolution of perinatal mortality in Ethiopia over time? (2) Which homogeneous ensemble machine learning methods is suitable to predict perinatal mortality in Ethiopia effectively? (3) What are the determinant factors of perinatal mortality in Ethiopia? (4) What are the important rules that may shape policies and interventions towards reducing perinatal mortality in Ethiopia?

The rest of this paper is organized as follows: Section II presents related works, Section III discusses materials and methods used, Section IV mentions experimental setup and result discussion, and Section V presents conclusion.

2. RELATED WORK

Several studies investigated perinatal mortality in Ethiopia using different methods. In study of Getachew et al. [14], investigated perinatal mortality and associated risk factors using a case-control study between 2008 and 2010. Subgroup binary logistic regression analyses were done to identify associated risk factors for perinatal mortality, stillbirths, and early neonatal deaths. In the study of Getachew et al. [14], a total of 1356 newborns (452 cases and 904 controls) were used in study sample size. The study reported that the perinatal mortality rate was 85/1000, and after or at 28 weeks of birth death accounts for 87% [14]. Adjusted odds ratios revealed that obstructed labor, malpresentation, preterm birth, death during the delivery haemorrhage, and hypertensive disorders of pregnancy was an independent predictor for high perinatal mortality.

Another study was conducted by Yemisrach et al. [15] on factors associated with perinatal mortality among public health deliveries in Addis Ababa, Ethiopia using an unmatched case-control study and

secondary data that was collected between 1st January up to 30th February 2015. According to Yemisrach [15], a total of 1113 (376 cases and 737 controls) maternal charts were reviewed and the mean age of the mothers for cases and controls were 26.47 ± 4.87 and 26.95 ± 4.68, respectively. Five hundred ninety-seven (53.6%) mothers delivered for the first time and factors that are significantly associated with increased risk of perinatal mortality were birth interval less than 2 years, preterm delivery, anemia, congenital anomaly, previous history of early neonatal death, and low birth weight. Use of partograph was also associated with decreased risk of perinatal mortality. Another study was also conducted by Bekele et al.[16] on the effect of community-based health insurance on utilization of outpatient health care services in Yirgalem town, Southern Ethiopia. This study used both quantitative and qualitative (mixed) approaches using a comparative cross-sectional study design. Randomly selected sample of 405 (135 members and 270 non-members) household heads were used for quantitative analysis. Multivariate logistic regression was employed to identify the effect of community-based health insurance on healthcare utilization. This study reveals that members of households with community-based health insurance were about three times more likely to utilize outpatient care than their non-member counterparts [AOR: 2931; 95% CI (1.039, 7.929); p-value=0.042]. Finally, the researcher concludes that community-based health insurance is an effective tool to increase the utilization of healthcare services and provide the scheme to member households. Kabudulaet et al.[17], Conducted on To Evaluation of machine learning methods for predicting the risk of child mortality in South Africa. The data was combined from two source from South Africa national income dynamics survey (NIDS) and district health barometer. They used machine learning algorithms such as Random forest, logistic regression and extreme gradient boost, with accuracy of 53.33%, 58.88% 58.89% respectively. Nyuyen et at.[18], To Evaluating statistical and machine learning methods to predict risk of in-hospital child mortality in Uganda. The surveillance project collected data from April 2010 to March 2014 across six public hospitals in Uganda: Tororo, Apac, Jinja, Mubende, Kabale, and Kanungu. They employed machine learning algorithm such as logistic regression, random forest, and gradient boost with accuracy of Scored Accuracy of 83%, 82%, and 83 % respectively.

However, in addition to the aforementioned studies [13] [15] [19] [20] [21], as they focused on identifying determinant risk factors only. Besides, these studies did not develop a predictive model, did not design an artifact, and did not generate rules that allow the development of preventive policies and measures. This study, hence,

motivated to fill these gaps by identifying risk factors, constructing a predictive model, design artifact, and generate rules that help to develop evidence-based policies and interventions towards perinatal mortality in Ethiopia.

3. METHODOLOGY

Figure 1 depicts the proposed model architecture that was implemented in this study to construct a predictive model, identify risk factors, extract relevant rules, and design artifacts.

3.1. Data collection

In this study we used secondary data, the Ethiopia Demographic and Health Surveys (EDHS) of 2011, 2016, and 2019 G.C, which was collected by the Ethiopian Central Statistical Agency in five years intervals.

The EDHSs are nationally representative household surveys that collect data for a variety of demographic, health, and nutrition monitoring and impact evaluation purposes.

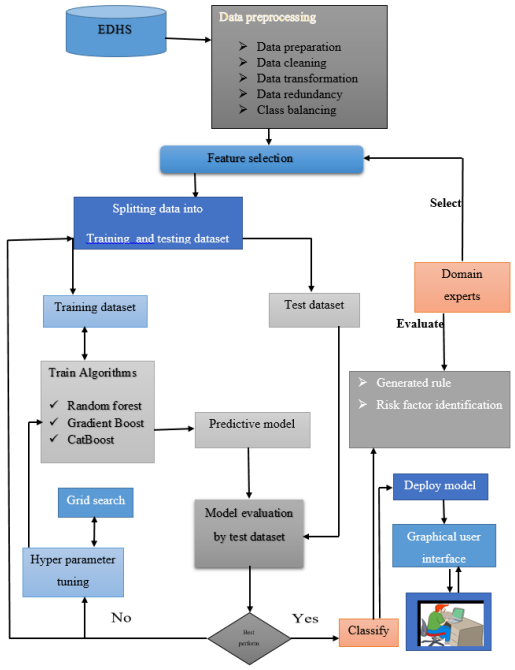


Figure 1:- The proposed model architecture

3.2. Data preprocessing

Data imputation (mode for categorical data and mean for continuous data) method was employed to substitute the missing values. Outliers were identified using a boxplot and replaced using the Interquartile Range (IQR) scores. Binning data discretization was applied to transform some of the features. For example, the feature 'education level of mothers (v106)' has 8 different values which were transformed into five different values (illiterate (1), grade 1-8 (elementary), grade9-12 (secondary), grade 12+ (tertiary), and higher education (university and college)). The synthetic minority over-sampling technique (SMOTE) was implemented to handle the class imbalance. The main reason that we use SMOTE is it avoids loss of valuable information [22][23]. The raw data contains 45 columns and 109531 instances. After applying SMOTE the data becomes 148659 and we used these data for the final experiment. Then, feature selection was conducted using filter and wrapper methods. Four experiments were conducted for selecting the relevant features for developing a perinatal mortality prediction model. As a result, different features were selected in each experiment. But, the backward selection method was registered the highest performance of 90.5% of accuracy using random forest classifiers with 13 features. The domain experts also recommended additional 4 features for further process and the total features selected for further analysis are 17 (Table 1).

4. EXPERIMENTAL SETUP, RESULTS AND DISCUSSION

4.1.What is the underline structure and evolution of perinatal mortality in Ethiopia over time?

The perinatal mortality was reduced over time in Ethiopia. This is because of increase the hospitality especially in rural areas and the emergence of community-based health insurance due to the reason pregnant

women are pushed to hospital during birth to get care of health professionals. But due to COVID 19 pandemic, the data collected in 2019 were twice smaller than other years' collected data (Figure 2).

Table 1: Features selected by sequential forward feature selection

No	Feature code	Feature description
1	Bord	Birth interval
2	V024	Region
3	V013	Maternal age
4	V190	Wealth index
5	V717	Maternal occupation
6	V457	Anemia level
7	V394	Visited health facility last 12 week
8	V501	Marital status
9	V312	Current contraceptive
10	V161	Types of cooking fuel
11	V106	Educational level
12	V228	Preterm
13	V455	Hemoglobin level
14	V025	Place of residence
15	V481a	Community/mutual health insurance
16	V463a	Smoke cigarettes
17	V463c	Chews tobacco

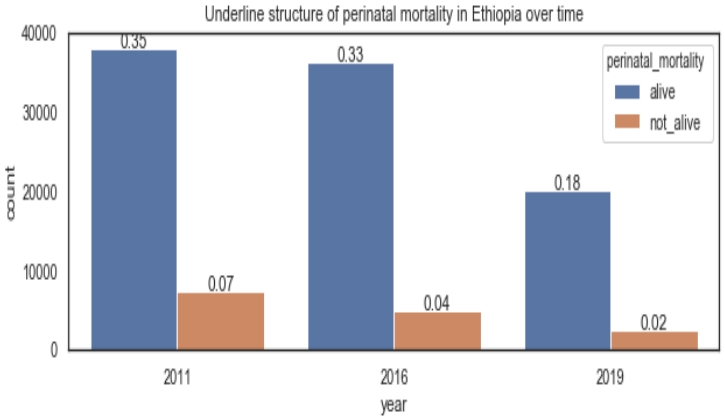


Figure 2: Perinatal mortality over time in Ethiopia

4.2. Which homogeneous ensemble machine learning methods predict perinatal mortality in Ethiopia effectively?

Three experiments were conducted to build a perinatal mortality predictive model using classification algorithms namely: Gradient Boost, CatBoost, and random forest classifiers. Grid search was applied to tune the hyperparameters of these algorithms. As a result, gradient boosting performed better with 99.72% recall, 90.24% accuracy, 92.80% f1-score, 86.96% ROC and 87.24% precision, as compared to other algorithms and batter than previous study with in overall performance of 83% using gradient boosting in pervious study. The recall indicates that there is a maximized true positive rate and a minimized false-negative rate meaning; there is a minimum false-negative rate.

Therefore, the gradient boost algorithm is selected as the best homogenous ensemble machine learning algorithm for predicting perinatal mortality based on maternal health status and health insurance service in the study area. The overall results of each experiment is shown in Table 2.

Table 2: Overall performance of models

Evaluation	Algorithms		
	Gradient Boost (%)	Cat Boost (%)	Random forest (%)
Accuracy	**90.24**	81.45	89.95
Precision	87.24	82.01	86.42
Recall	99.72	90.75	99.54
ROC	86.96	77.98	86.50
F1_Score	92.80	86.16	92.72

4.3. What are the determinant factors of perinatal mortality in Ethiopia?

Feature importance analysis was conducted to identify determinant risk factors of perinatal mortality in Ethiopia using the model developed using gradient boosting, as it has scored high accuracy than other algorithms. As a result, factors that are significantly associated with increased risk of perinatal mortality are birth interval less than 2 years, preterm delivery, anemia, congenital anomaly, educational status, family size, occupation, marital status, traveling time to the nearest health institution, perceived quality of care, the first choice of place for treatment during illness and expected healthcare cost of recent treatment, prematurity, low birth weight, previous history of perinatal death, not receiving tetanus toxoid immunization, and lack of iron supplementation, see Table 3.

Table 3: Risk factors with feature importance

No	Feature code	Feature description	Value
1	Bord	Birth interval	0.291119
2	V024	Region	0.122834
3	V013	Maternal age	0.077887
4	V190	Wealth index	0.072292
5	V717	Maternal occupation	0.055206
6	V457	Anemia level	0.054080
7	V394	Visited health facility last 12 week	0.038728
8	V501	Marital status	0.030207
9	V312	Current contraceptive	0.026625
10	V161	Types of cooking fuel	0.026059
11	V106	Educational level	0.021210
12	V228	Preterm	0.019435
13	V455	Hemoglobin level	0.016383
14	V025	Place of residence	0.009877
15	V481a	Community/mutual health insurance	0.007053
16	V463a	Smoke cigarettes	0.006037
17	V463c	Chews tobacco	0.002598

4.4. What are the important rules that may shape policies towards reducing and/or preventing perinatal mortality in Ethiopia?

The most relevant rules were generated from the best-performed algorithm (gradient boost) model, as it has registered high accuracy, and the rules were validated by the domain experts. Sample rules are presented here below:

Rule1:- if currently breast feeding and preterm == 'no' AND maternal education== 'no education' AND wanted least children == 'wanted then' AND smoke ciggrate == 'no' AND health insurance provide by employer == 'no' AND smoke Tabaco == 'never in union' AND types of cooking fuel == 'wood' AND occupation == 'not working AND wealth index== 'poorest' AND maternal age == '35-39' AND place of residence == 'rural' AND Community based health insurance == 'no' AND Then child=='Alive'

Rule 2:- if currently breast feeding and preterm == 'no' AND maternal education== 'no education' AND wanted least children == 'wanted then' AND smoke ciggrate == 'no' AND health insurance provide by employer == 'no' AND smoke Tabaco == 'never in union' AND types of cooking fuel == 'wood' AND occupation == 'not working AND wealth index== 'poorest' AND maternal age == '40-44' AND place of residence == 'urban' AND Community based health insurance == 'no' AND Then child=='Died'

Rule3:- if currently breast feeding and preterm == 'no' AND maternal education== 'no education' AND wanted least children == 'wanted then' AND smoke ciggrate == 'no' AND health insurance provide by employer == 'no' AND smoke Tabaco == 'never in union' AND types of cooking fuel == 'wood' AND occupation == 'not working AND wealth index== 'poorest' AND maternal age == '45-49' AND place of residence == 'rural' AND Community based health insurance == 'no' AND Then children=='Alive'

Rule4:- if currently breast feeding and preterm == 'no' AND maternal education== 'no education' AND wanted least children == 'wanted then' AND smoke ciggrate == 'no' AND health insurance provide by employer == 'no' AND smoke Tabaco == 'never in union' AND types of cooking fuel == 'wood' AND occupation == 'not working AND wealth index== 'poorest' AND maternal age == '45-49' AND place of residence == 'rural' AND Community based health insurance == 'no' AND Then children=='Alive'

Rule 5:- if currently breast feeding and preterm == 'no' AND maternal education== 'no education' AND wanted least children == 'wanted then' AND smoke ciggrate == 'no' AND health insurance provide by employer == 'no' AND smoke Tabaco == 'never in union' AND types of cooking fuel == 'wood' AND occupation == 'not working AND wealth index== 'poorest' AND maternal age == '15-19' AND place of residence == 'rural' AND Community based health insurance == 'no' AND Then children== 'Died'

5. DISCUSSION

As we have discussed in the experimental result section the proposed system was achieved a performance of 90.24% overall performance using gradient boosting machine learning algorithm, Which better results compared to which achieved 83% overall performance using gradient boosting machine learning algorithm in pervious study. We have deployed the model on the using Flask framework through Heroku on the web that can be available freely, visa these link: -

http://perinatal-mortality.herokuapp.com/

6. CONCLUSION

This study aims at developing a predictive model for perinatal mortality in the case of Ethiopia by using homogeneous ensemble machine learning methods. We identified the determinant risk factors of perinatal mortality with feature importance techniques such as maternal residence, level of education, birth interval, and community based health insurance. The gradient boost algorithm has registered the highest performance with 99.72% recall, 90.24% accuracy, 92.80% f1-score, 86.96% ROC and 87.24% precision. The developed predictive was correctly predicts perinatal mortality 90.24% based on objective metrics evaluation and then better than pervious study conducted using gradient boosting which scored accuracy of 83%. The most relevant rules, that helps to formulate policies towards maintaining perinatal mortality, were generated from gradient boost model, and the rules were validated by the domain experts.

Finally we recommend the implement other algorithms and techniques such as heterogeneous ensemble machine learning methods and combined methods of feature selection techniques.

ACKNOWLEDGEMENTS

We would like to acknowledge the Ethiopia central statistical agency for providing us the data.

REFERENCE

[1] A. B. Comfort, L. A. Peterson, and L. E. Hatt, "Effect of health insurance on the use and provision of maternal health services and maternal and neonatal health outcomes: A systematic review," *J. Heal. Popul. Nutr.*, vol. 31, no. 4 SUPPL.2, 2013, doi: 10.3329/jhpn.v31i4.2361.

[2] V. Jain and J. M. Chatterjee, "Machine Learning with Health Care Perspective: Machine Learning and Healthcare," no. March, 2020, doi: 10.1007/978-3-030-40850-3.

[3] R. Rasaily *et al.*, "Effect of home-based newborn care on neonatal and infant mortality: A cluster randomised trial in India," *BMJ Glob. Heal.*, vol. 5, no. 9, pp. 1–11, 2020, doi: 10.1136/bmjgh-2017-000680.

[4] J. R. Daw, T. N. A. Winkelman, V. K. Dalton, K. B. Kozhimannil, and L. K. Admon, "Medicaid expansion improved perinatal insurance continuity for low-income women," *Health Aff.*, vol. 39, no. 9, pp. 1531–1539, 2020, doi: 10.1377/hlthaff.2019.01835.

[5] E. Jukes, *Encyclopedia of Machine Learning and Data Mining (2nd edition)*, vol. 32, no. 7/8. 2018.

[6] W. Soors, N. Devadasan, V. Durairaj, and B. Criel, "Community Health Insurance and Universal Coverage: Multiple paths, many rivers to cross," pp. 1–122, 2010.

[7] N. Haven *et al.*, "Community-based health insurance increased health care utilization and reduced mortality in children under-5, around Bwindi Community Hospital, Uganda between 2015 and 2017," *Front. Public Heal.*, vol. 6, no. OCT, 2018, doi: 10.3389/fpubh.2018.00281.

[8] B. J. Akombi and A. M. Renzaho, "Perinatal mortality in sub-saharan africa: A meta-analysis of demographic and health surveys," *Ann. Glob. Heal.*, vol. 85, no. 1, pp. 1–8, 2019, doi: 10.5334/aogh.2348.

[9] P. R. Ghimire, K. E. Agho, A. M. N. Renzaho, M. K. Nisha, M. Dibley, and C. Raynes-Greenow, "Factors associated with perinatal mortality in Nepal: Evidence from Nepal demographic and health survey 2001-2016," *BMC Pregnancy Childbirth*, vol. 19, no. 1, pp. 1–12, 2019, doi: 10.1186/s12884-019-2234-6.

[10] Z. A. Hassan and M. J. Ahmed, "Factors associated with immunisation coverage of children aged 12- 24 months in Erbil / Iraq 2017-2018," vol. 24, no. 08, pp. 12222–12235, 2020, doi: 10.37200/IJPR/V24I8/PR281205.

[11] R. A. Knuppel and J. H. Shepherd, "Perinatal mortality rates," *Br. Med. J.*, vol. 280, no. 6228, p. 1376, 1980,

doi: 10.1136/bmj.280.6228.1376.

[12] B. H. Jena, G. A. Biks, K. A. Gelaye, and Y. K. Gete, "Magnitude and trend of perinatal mortality and its relationship with inter-pregnancy interval in Ethiopia: A systematic review and meta-analysis," *BMC Pregnancy Childbirth*, vol. 20, no. 1, pp. 1–13, 2020, doi: 10.1186/s12884-020-03089-2.

[13] G. T. Debelew, "Magnitude and Determinants of Perinatal Mortality in Southwest Ethiopia," *J. Pregnancy*, vol. 2020, 2020, doi: 10.1155/2020/6859157.

[14] G. Bayou and Y. Berhan, "Perinatal mortality and associated risk factors: a case control study.," *Ethiop. J. Health Sci.*, vol. 22, no. 3, pp. 153–62, 2012.

[15] Y. Getiye and M. Fantahun, "Factors associated with perinatal mortality among public health deliveries in Addis Ababa, Ethiopia, an unmatched case control study," *BMC Pregnancy Childbirth*, vol. 17, no. 1, pp. 1–7, 2017, doi: 10.1186/s12884-017-1420-7.

[16] B. Demissie and K. G. Negeri, "Effect of community-based health insurance on utilization of outpatient health care services in southern ethiopia: A comparative cross-sectional study," *Risk Manag. Healthc. Policy*, vol. 13, pp. 141–153, 2020, doi: 10.2147/RMHP.S215836.

[17] C. Kabudula *et al.*, "Evaluation of machine learning methods for predicting the risk of child mortality in South Africa," 2019.

[18] G. Nguyen, "Evaluating statistical and machine learning methods to predict risk of in-hospital child mortality in Uganda," 2016.

[19] D. D. Atnafu, H. Tilahun, and Y. M. Alemu, "Community-based health insurance and healthcare service utilisation, North-West, Ethiopia: A comparative, cross-sectional study," *BMJ Open*, vol. 8, no. 8, pp. 1–6, 2018, doi: 10.1136/bmjopen-2017-019613.

[20] K. Shiferaw, B. Mengiste, T. Gobena, and M. Dheresa, "The effect of antenatal care on perinatal outcomes in Ethiopia: A systematic review and meta-analysis," *PLoS One*, vol. 16, no. 1 January, pp. 1–19, 2021, doi: 10.1371/journal.pone.0245003.

[21] D. Prasad *et al.*, "The effect of community-based health insurance on the utilization of modern health care services : Evidence from Burkina Faso," vol. 90, pp. 214–222, 2009, doi: 10.1016/j.healthpol.2008.09.015.

[22] I. Journal and C. Science, "Class Imbalance Problem in Data Mining : Review," vol. 2, no. 1, 2013.

[23] R. P. Ribeiro, "SMOTE for Regression," no. October 2015, 2013, doi: 10.1007/978-3-642-40669-0.

Auscultation Performance Metrics Computation using Machine learning Algorithms

S. Rajkumar, V. Ellappan*, Rajaveerappa Devadas, Gemechu Dengia, and Bayisa Taye Mulatu

Department of ECE, School of Electrical Engineering and Computing, Adama Science and Technology University, Adama, Ethiopia

**Corresponding author, e-mail: ellappan.v@gmail.com*

ABSTRACT

Heart sounds (HS) and lung sound (LS) signal separation is a challenging research task for respiratory specialists and cardiologists. In this study, various performance parameters for heart and lung sound signal separation based on machine learning algorithm is evaluated and compared. Machine learning algorithm over signal source separation is a challenging signal processing problem. This is the first initiative attempt in real time signal (Heart and lung sound) to compare signal source separation based on SIX major Algorithms (JADE, Kernal, Fastica, Infomax, ExInfomax and Radical) for various performance parameters. The empirical results demonstrate the effectiveness of various algorithms with performance superiority over these reference techniques for various performance metrics. Algorithms were analyzed in terms of both classification accuracies and performance metrics.

Keywords: *Machine learning; Jade; Kernal; Radical; Fastica; Infomax and ExInfomax*

1. INTRODUCTION

Auscultation [1] is the most important and effective clinical technique for evaluating a patient's respiratory function. Auscultation of the chest is a diagnostic method used by physicians, owing to its simplicity and noninvasiveness. Lung Sound Signal (LSS) is measured and used as an aid in the diagnosis of various diseases. However, their interpretation is difficult due to the presence of interference generated by the heart sound signals (HSS). These two signals are superimposed with one another.

There has been considerable increase in interest and efforts to develop new algorithms for successful HSS and LSS separation and classification [1]. Various algorithms and techniques have been presented in the literature during last few decades and analyzed different aspects. They include high pass filter, adaptive filtering algorithms, wavelet based denoising algorithms, time–frequency filtering, modulation filtering and independent component analysis [2-5]. From perspectives of blind source separation (BSS) scenarios, machine learning algorithm play a dominant role in successful separation in various applications [1,3,8,9,15 and 22-27].

BSS is widely used to biomedical signal processing, audio and array signal processing and digital communication [2][7][10] and [14]. There are a variety of techniques [2-13] which includes Fastica [11-21], Infomax [2-8], ExInfomax [9-11], JADE [21] and Kernel [22-27]) for many biomedical applications.

Infomax Algorithm is widely used algorithm for blind source separation in EEG [12] and fMRI data analysis [2]. Blind source separations (BSS) using independent component-based analysis have been studied in-depth [3] to extract common hemodynamic sources for a group of functional magnetic resonance images (fMRI).

A new second-order Hessian-free algorithm for Infomax is introduced by [4] which achieve asymptotically quadratic convergence.

Efficient chip design for convolutive blind source separation (CBSS) adopted using the information maximization (Infomax) [5] method consists mainly of Infomax filtering modules and scaling factor computation modules.

Infomax Algorithm is also implemented in Visible Light Communication (VLC) signal separation in [6] based on the artificial neural network for the analysis of covariance of the values from signals. They are also implemented for the extraction of class-discriminant information in remote sensing hyper spectral image classification [7] and magneto encephalography (MEG)-based real-time brain computing interfaces (BCI) [8].

Various researches have been done using Extended- Infomax algorithm. They were applied to character recognition of Brain-Computer Interface (BCI) system [9] [10] based on P300 (a kind of evoked potentials). This algorithm also used in various research areas, analysis of fMRI data.

The Infomax algorithm and its extended version that adapts to sub and super-Gaussian distributions have been widely used in various research areas, including analysis of fMRI data [11]. A constrained version of the extended Infomax algorithm is used as an example to show the benefits obtained from the non-orthogonal constrained framework.

Kernel Algorithm is widely used algorithm for blind source separation in electromyogram (EMG) signals [15] [16] for diagnosing neuromuscular disorders. Blind source separation (BSS) using kernel independent component-based analysis have been studied [17] [20] in-depth. Kernel Algorithm is also implemented to calculate the utility harmonic impedance [18], porosity defect detection [19] and smart antenna systems [21].

Kernel Algorithm is widely used algorithm for blind source separation for nonlinear and non-Gaussian process monitoring [22] [24], fault monitoring [23], fault detection, nonlinear feature extraction and data driven fault diagnosis [25] [26]. Kernel is also implemented in performance monitoring the high order non-Gaussian characteristics in chemical process [27].

RADICAL (Robust, Accurate, Direct freelance part Analysis Algorithm estimates the independent sources exploitation differential entropy estimator supported 'm'-spacing estimator. Joint Approximation Diagonalization of Eigen matrices (JADE) algorithm exploits the fourth order moments so as to separate the source signals from mixed signals.

In this work, we make the following unique contributions:

- While there is an increasing demand for blind source separation techniques, to the best of our knowledge, there is no comparative study published in Auscultation field to calculate their performance metrics.
- This is the first initiative attempt to calculate their metrics based on various machine learning algorithms in auscultation separation signals.
- Heterogeneity of metrics is evaluated and compared for various algorithms.
- Various important performance metrics found in the literature [2-27] includes: Error, Absolute Error Rate (AER), Correlation Coefficient (r), Mean Square Error (MSE), Root Mean Square Error (RMSE), Normalized Mean Square Error (NMSE), Peak Signal to Noise Ratio (PSNR), Signal to

Noise Ratio (SNR), Improved Signal to Noise Ratio (ISNR), Signal to Interference Ratio (SIR), Amari-error, Frobenius error and Maximum Signal to Noise Ratio (SNR-MAX) are calculated.

- This work shows a comparative study to evaluate machine learning algorithms based on various important performance metrics which paves the way for better algorithm comparison.

This paper is organized as follows: Sect. 2 describes machine learning algorithm model and its basic concepts; Sect. 3 briefly presents various performance metrics that are evaluated in this paper; Sect. 4 will present our simulation results. Finally, Future scope and concluding remarks are provided in Sect. 5 followed by references.

2. METHODOLOGY

2.1. Machine learning model

In this machine learning models, where a set of observed random data are linearly transformed into independent component data. Here, the target is to maximize the statistical independence of the output signal. If the inputs are known to be linear instant mixture of a group of sources, then method provides to estimate the input sources.

Here, neither the original sources nor the mixture matrix is known. This can be the Blind Separation of Sources (BSS) where the aim is to get a non-observable set of signals, the sources, from another set of observable signals are considered as mixtures. The BSS problem is simply tackled by exploiting the upper higher signal statistics and improvement techniques.

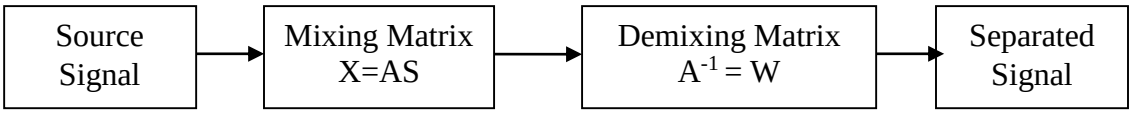


Figure 1: Schematic Illustration of the mathematical model used to perform decomposition

The original source vector **S** is of size M x N and also the mixing matrix **A** is of size **M x M**, where, **M** is that the variety of statistical independent sources and **N** is that the variety of samples in every source. The results of the separation method are that the demixing matrix **W** which might be used to obtain and acquire the estimated statistical independent sources, $\hat{S}$ from the mixtures. This method is described by Equation 1 and a schematic illustration of the mathematical model in shown in Figure 1.

$$X = AS \rightarrow \hat{S} = WX \qquad (1)$$

Preprocessing:

Some preprocessing is beneficial before making an attempt to estimate **W**.

(i) The determined signals should be focused by subtracting their mean E{**x**}

$$\tilde{X} = X = E[X] \qquad (2)$$

(ii) Then they are whitened, which implies they are linearly remodeled so the components are uncorrelated and has unit variance.

(iii) Whitening is performed via eigen value decomposition of the variance matrix, $\mathbf{V\Lambda V^T}$, **V** is that the matrix of orthogonal eigenvectors and **Λ** may be a square matrix with the corresponding eigen values. The whitening is done by multiplication with the transformation matrix **P**

$$P = V\Lambda^{1/2}V^{T} \quad (3)$$

$$Z = P\tilde{X} \quad (4)$$

Formulas:

For blind source separation of signals, we have calculated various important performance parameters [14-27] and shown in Tables 1-2.

Table 1: Performance Metrics Parameters

No	Parameter	Formula
1.	***Absolute Error Rate (AER)***	$AER = \frac{\lvert S_i - S_{\text{Ref}} \rvert}{S_{\text{Ref}}}$ (22) $\lvert . \rvert$ - Absolute value S_i - Output signal S_{Ref}- Reference Signal
2.	***Mean Square Error (MSE)***	$MSE = \frac{1}{N}\sum_{i=1}^{N}(S_{est}(i) - S_{actual}(\mathrm{i}))^2$ (23) S_{est} - Estimated signal S_{actual} - Actual signal N – Length of the signal
3.	***Root Mean Square Error (RMSE)***	$RMSE = \frac{1}{N}\sum_{K=1}^{N}\left[s(\mathrm{k}) - y(k) \right]^2$ (24) s(k) – Actual signal y(k) - output signal N – Number of samples
4.	***Normalized Mean Square Error (NMSE)***	$NMSE = \frac{\sum_{i=1}^{N}(S_{est}(i) - S_{actual}(\mathrm{i}))^2}{\sum_{i=1}^{N}(S_{noisy}(i) - S_{actual}(\mathrm{i}))^2}$ (25) S_{est} - Estimated signal S_{actual} - Actual signal S_{noisy} - Noisy signal N – Number of samples
5.	***Peak Signal to Noise Ratio (PSNR)***	$PSNR = 20\log_{10}\left(\frac{64}{RMSE}\right)$ (26)

6.	***Signal to Noise Ratio (SNR)***	$SNR_{dB} = 10\log_{10}\left[\frac{\sum_{k=0}^{N-1} S(k)^2}{\sum_{k=0}^{N-1}\left[S(k) - \hat{Y}(k)\right]^2}\right]$ (27) S(k) – Input Signal $\hat{Y}(k)$ - Estimated output signal N – Number of samples
7.	***Improved Signal to Noise Ratio (ISNR)***	$ISNR_{dB} = 10\log_{10}\left(\frac{(s(k) - x(k))^2}{(s(k) - y(k))^2}\right)$ (28) s(k) – Input Signal x(k) – Mixed Signal y(k) – Output Signal
8.	***Signal to Interference Ratio (SIR)***	$SIR_{dB} = 10\log\left(\frac{\sum_{t=1}^{T}\|s_t\|^2}{\sum_{t=1}^{T}\|y_t - s_t\|^2}\right)$ (29) s_t– Source signals s= {s1, s2...sT} y_t– Demixed signals y= (y1, y2…. yT}
9.	***Amari-error***	$d(U,V) = \frac{1}{m}\left(\sum_{i=1}^{m}\frac{\sum_{j=1}^{m}\|B_{ij}\|}{\max_j\|B_{ij}\|} + \sum_{j=1}^{m}\frac{\sum_{i=1}^{m}\|B_{ij}\|}{\max_i\|B_{ij}\|}\right) - 2$ (30) U, V – Matrices B = UV-1(It is necessary to normalize each row and of U and V) Amari error lies on [0 to (m-1)]
10.	***Frobenius error***	$d_F(\hat{W}, W_p) = \left\|\hat{W}W_p^{-1} - I_{mxm}\right\|F$ (31)

3. RESULTS AND DISCUSSION

The horizontal axis represents the samples and the vertical axis represents amplitude in all the graphs. Here sampling frequency 44100 Hz and time be 2.5 sec is used. The reference lung and heart sound signals, mixed (noise) signals and algorithm outputs are shown in the Figure 2-8 respectively are obtained from *R.A.L.E.® Research System.*

Table 2: Performance Metrics Evaluation

MSE

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	1.0386	0.6316	1.667	0.6011	0.6013	0.6011
Heart	1.0079	0.0299	0.0395	0.7982	1.2243	1.2244

RMSE

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	1.0191	0.7947	1.2911	0.7753	0.7754	0.7753
Heart	1.0039	0.1728	0.1988	0.8934	1.1065	1.1065

NMSE

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	24.3943	34.6069	34.6318	0.000101	0.0151	0.0043
Heart	0.8609	1.4856	1.3545	1.39E-05	1.1917	1.2038

PSNR

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	35.9589	38.1194	33.9041	38.3341	38.3329	38.3344
Heart	36.0894	51.3716	50.1547	37.1025	35.2446	35.2442

SNR_MAX

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	1.6205	0.9028	0.9021	0.8881	0.8884	0.8881
Heart	0.8883	1.1389	0.9167	1.6219	1.6181	1.6217

ISNR

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	-24.7953	16.7682	20.9836	2.9277	22.4222	22.4199
Heart	-17.6389	1.7841	3.0011	0.7523	18.4842	18.4841

SIR

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	0.7677	0.7362	0.7333	0.6997	0.7039	0.6835
Heart	0.7849	-0.3422	-0.0855	1.3561	0.6137	0.4696

SNR

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung	12.8735	13.4886	18.9033	0.0265	12.8736	12.8735
Heart	24.3404	-2.4221	2.6362	-0.0264	24.3414	24.3404

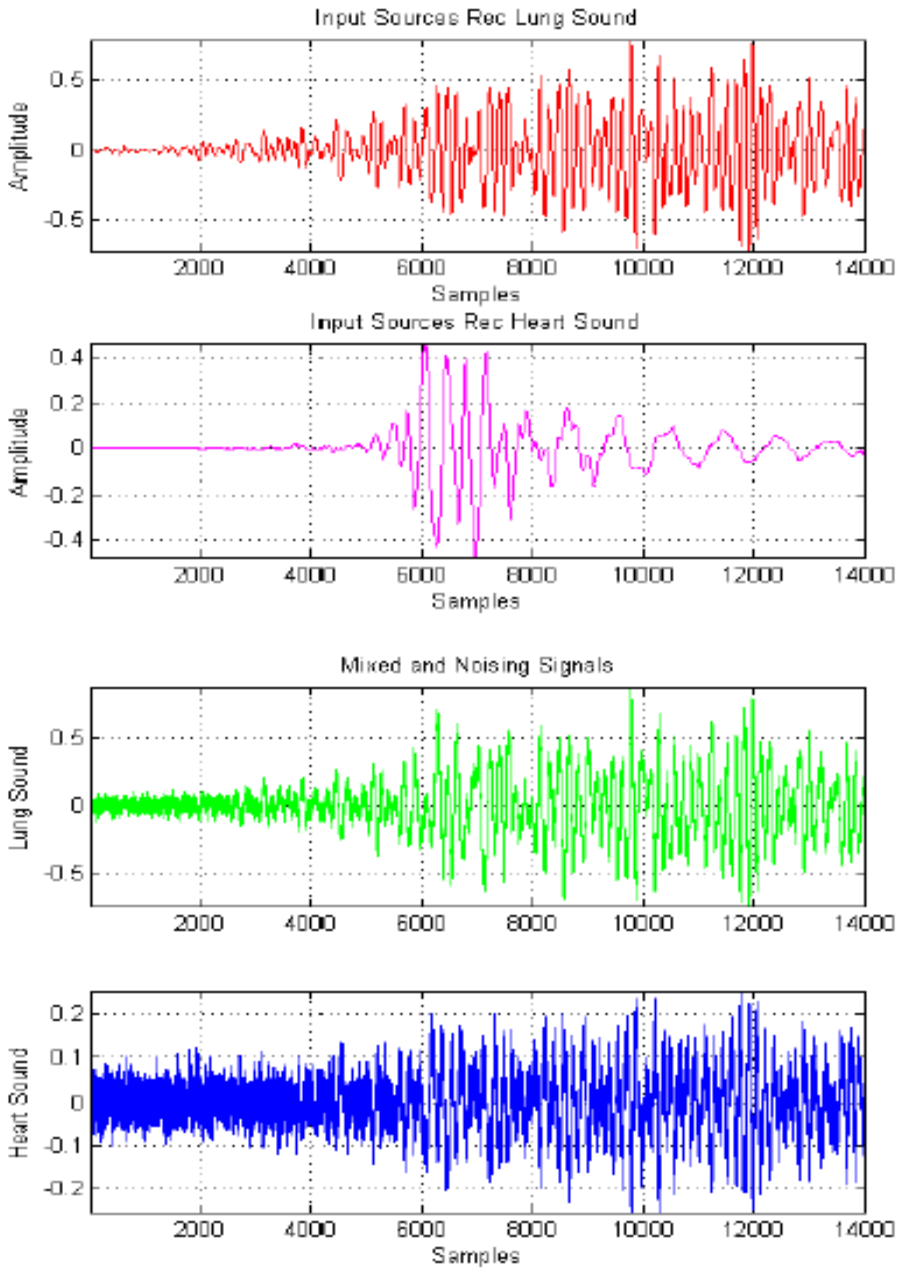
Frobenius Error

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung & Heart	0.1111	1.0064	2.0198	6.66E-06	46.4751	46.4901

AMARI Error

Sounds	Fast ICA	InfoMax	Ex-Infomax	Kernal	JADE	Radical
Lung & Heart	-0.1831	-2.4597	-1.5749	0.012	12.4976	12.211

3.1. Ex-Infomax



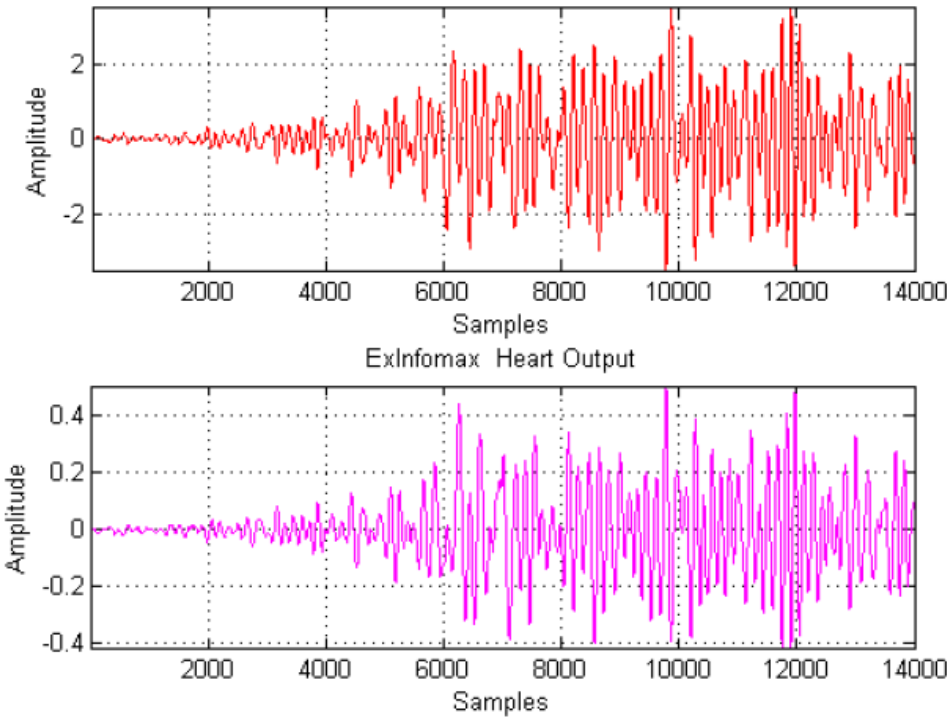


Figure 2: Simulated Output -ExInfomax model constructed on the Auscultation signal

3.2. InfoMax

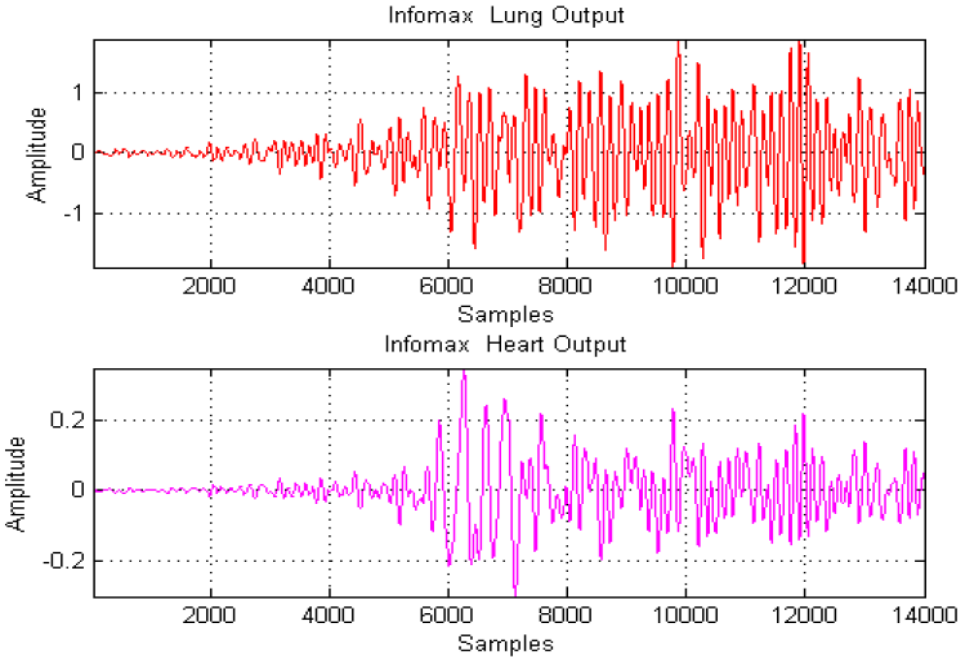


Figure 3: Simulated Output - Infomax model constructed on the Auscultation signal

3.3. Kernal Simulation Results:

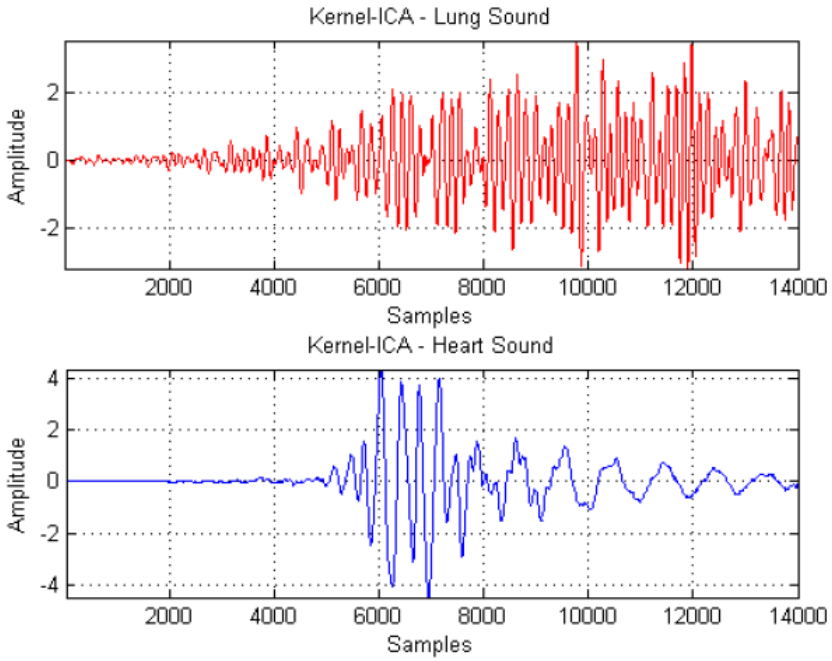


Figure 4: Simulated Output – Kernal model constructed on the Auscultation signal

3.4. Fastica Simulation Results:

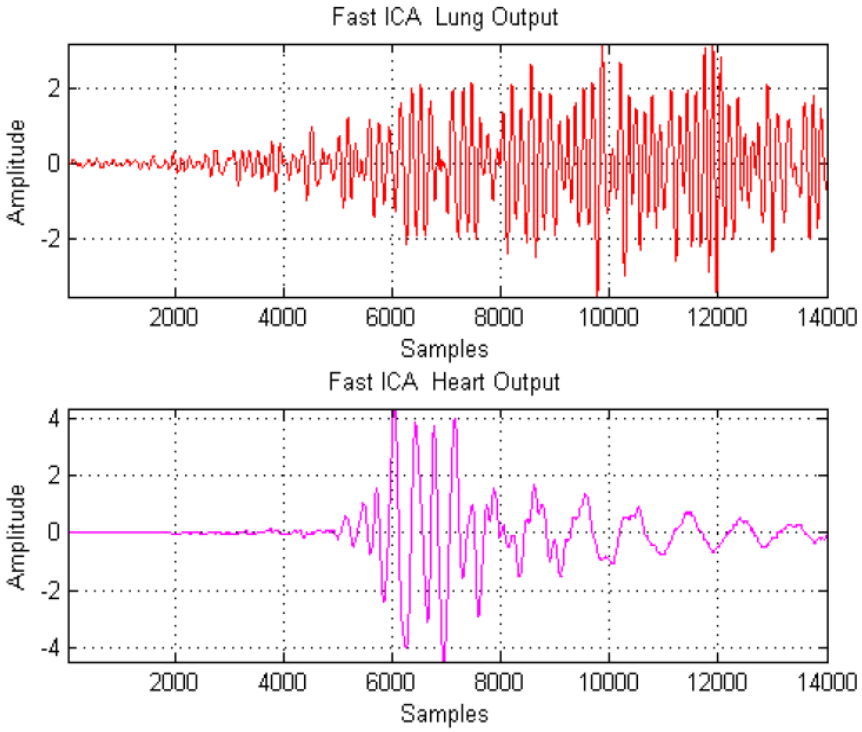


Figure 5: Simulated Output – Fastica model constructed on the Auscultation signal

3.5. JADE Simulation Results:

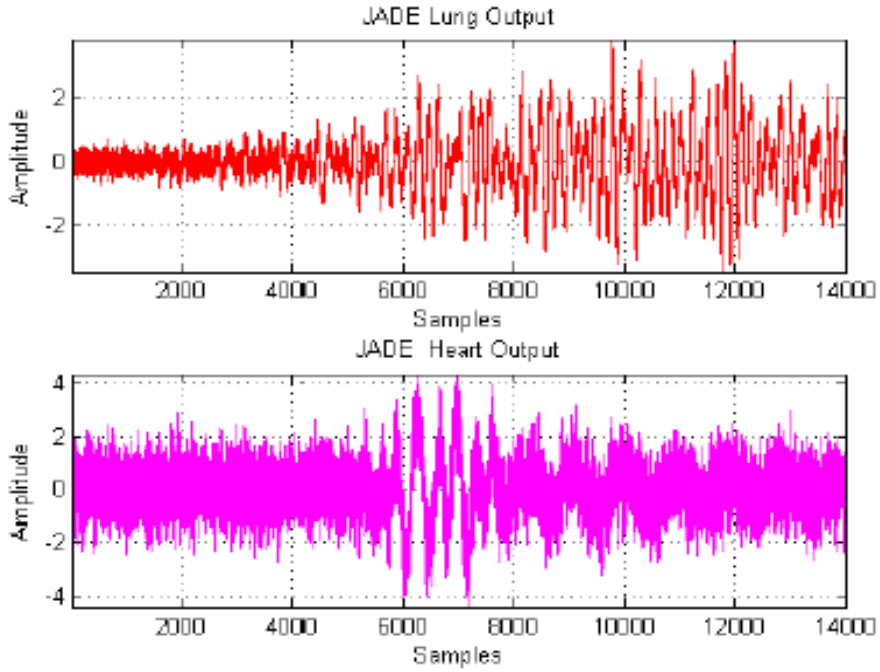


Figure 6: Simulated Output -JADE model constructed on the Auscultation signal

3.6. Radical Simulation Results:

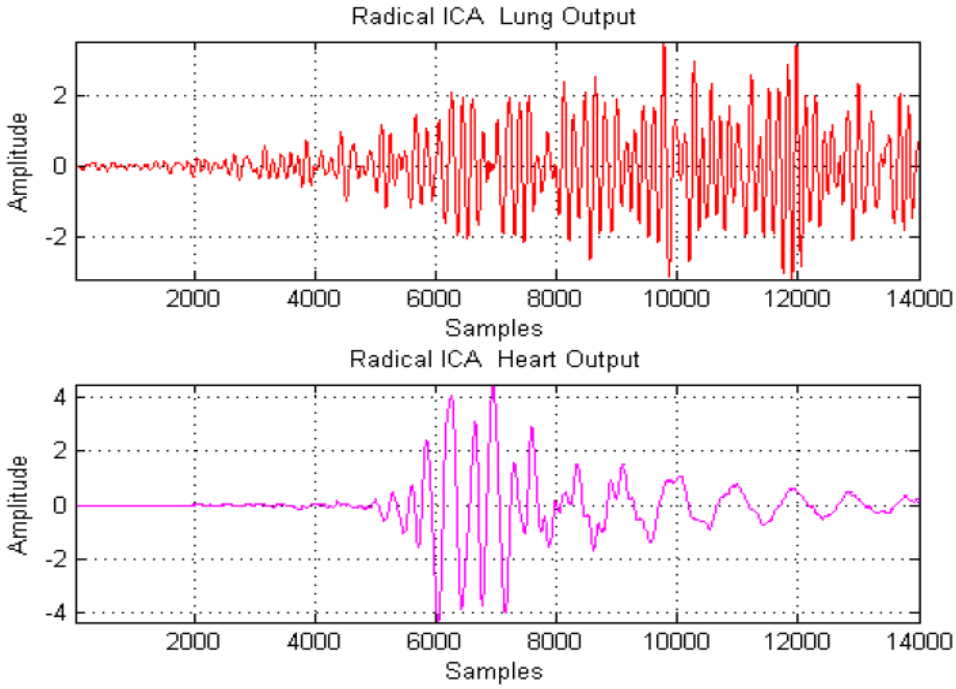
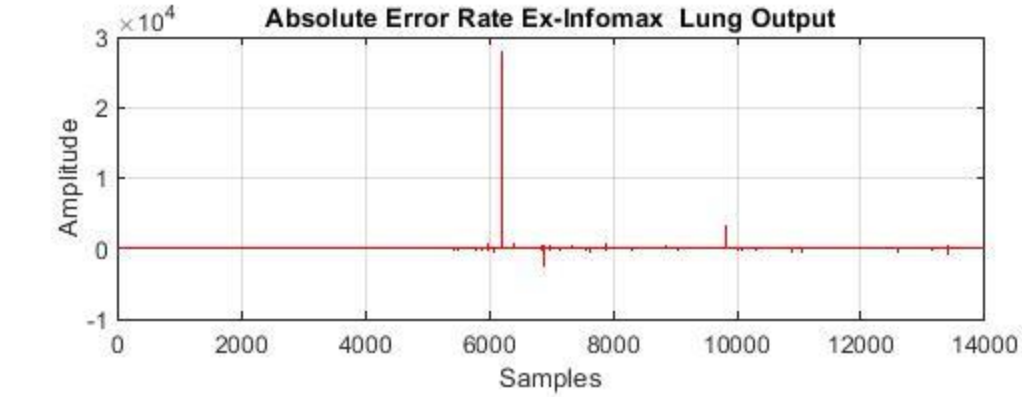
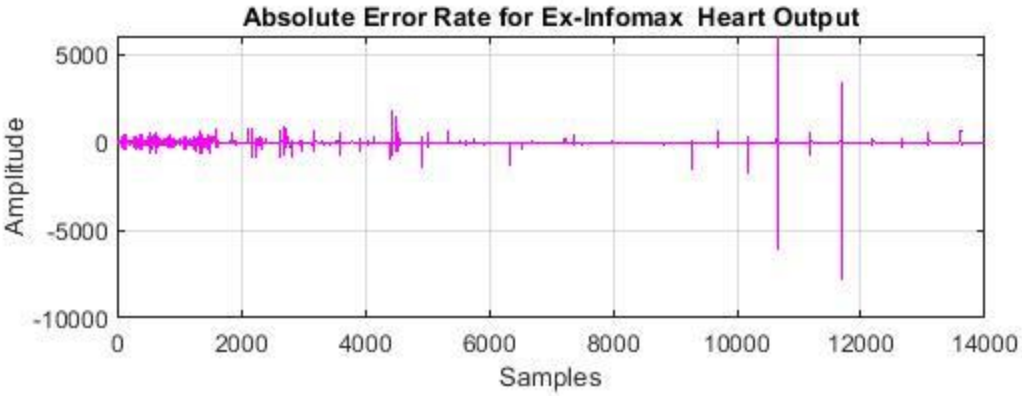
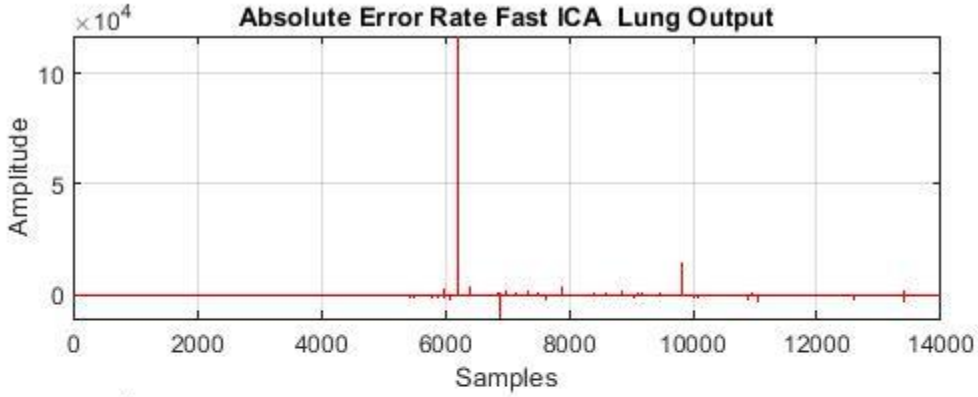
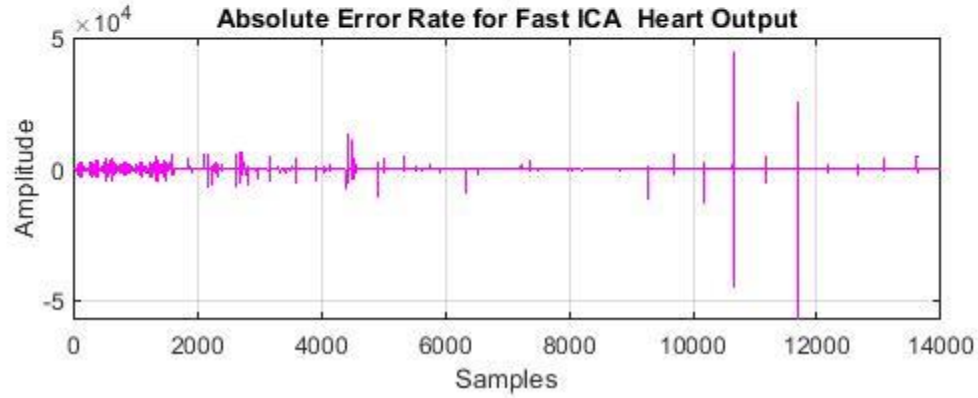


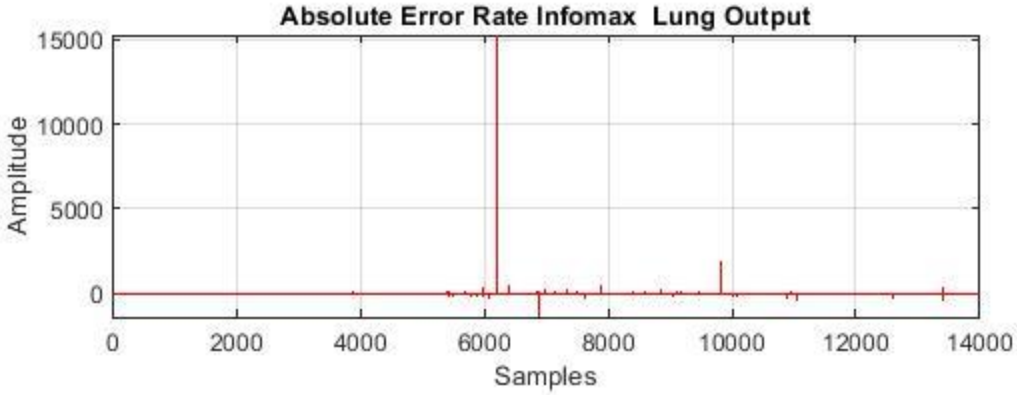
Figure 7: Simulated Output -Radical model constructed on the Auscultation signal

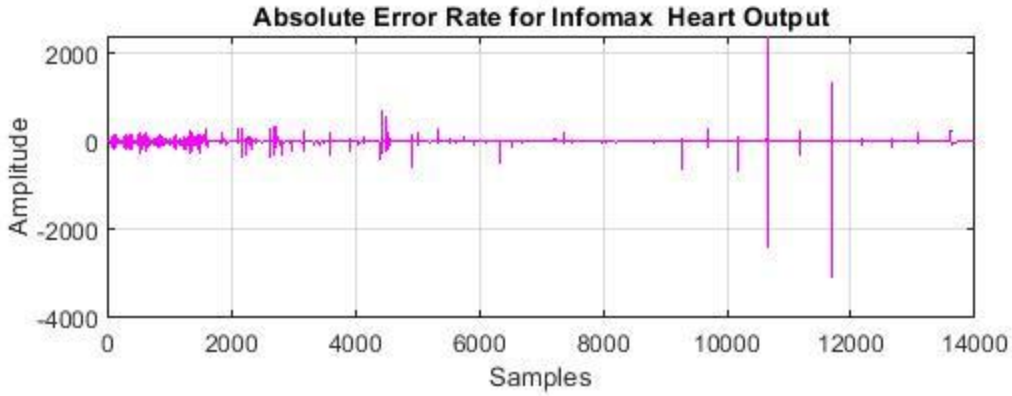

Absolute Error Rate Ex-Infomax Lung Output
×10^4
Amplitude
Samples

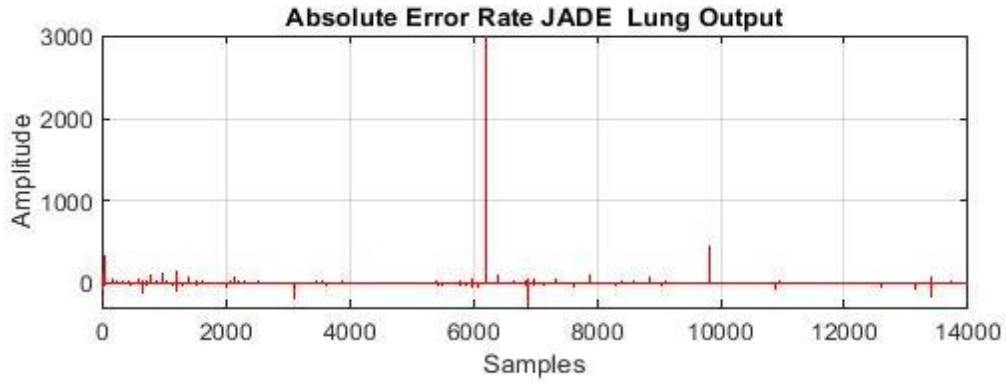

Absolute Error Rate for Ex-Infomax Heart Output
Amplitude
Samples

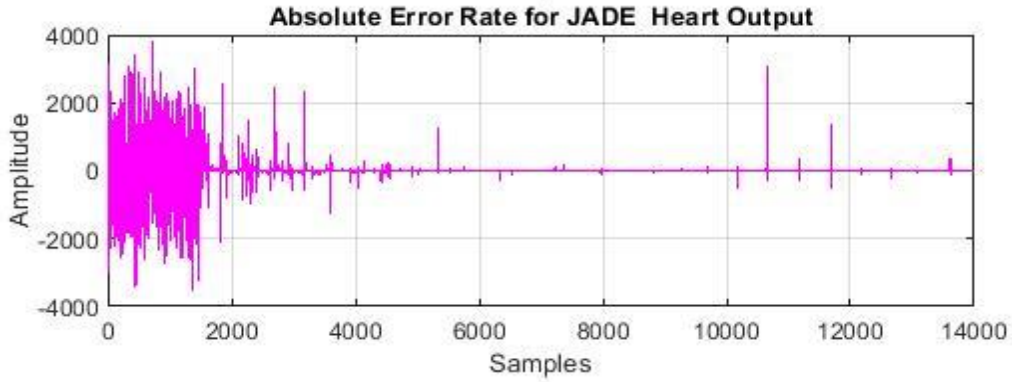

Absolute Error Rate Fast ICA Lung Output
×10^4
Amplitude
Samples


Absolute Error Rate for Fast ICA Heart Output
×10^4
Amplitude
Samples


Absolute Error Rate Infomax Lung Output
Amplitude
Samples
15000
10000
5000
0
0
2000
4000
6000
8000
10000
12000
14000


Absolute Error Rate for Infomax Heart Output
Amplitude
Samples
2000
0
-2000
-4000
0
2000
4000
6000
8000
10000
12000
14000


Absolute Error Rate JADE Lung Output
Amplitude
Samples
3000
2000
1000
0
0
2000
4000
6000
8000
10000
12000
14000


Absolute Error Rate for JADE Heart Output
Amplitude
Samples
4000
2000
0
-2000
-4000
0
2000
4000
6000
8000
10000
12000
14000

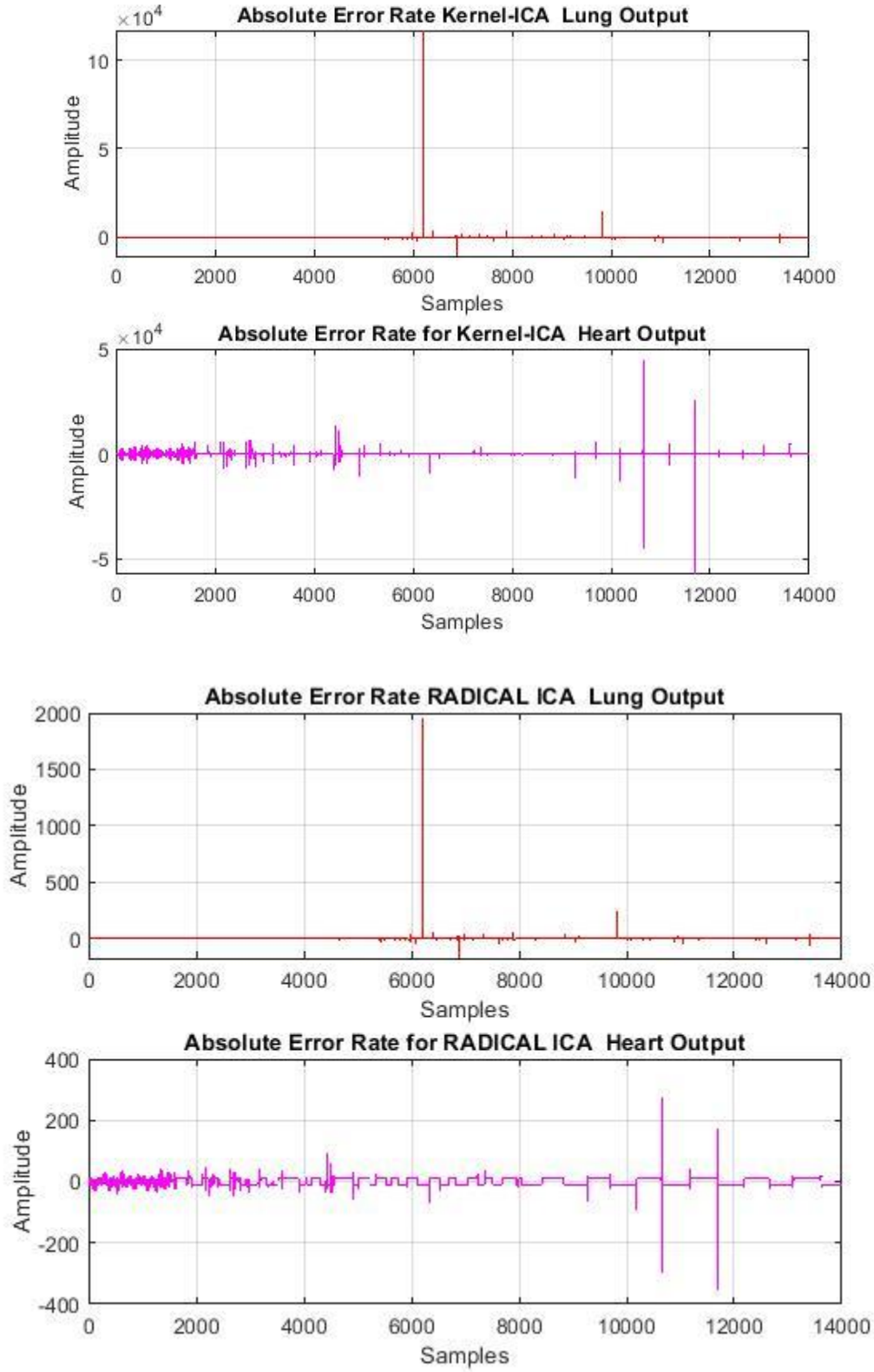
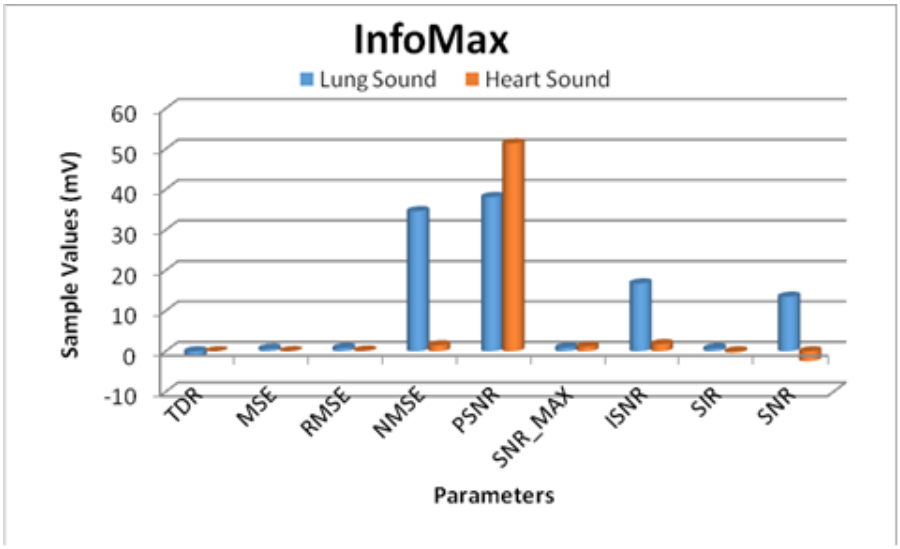
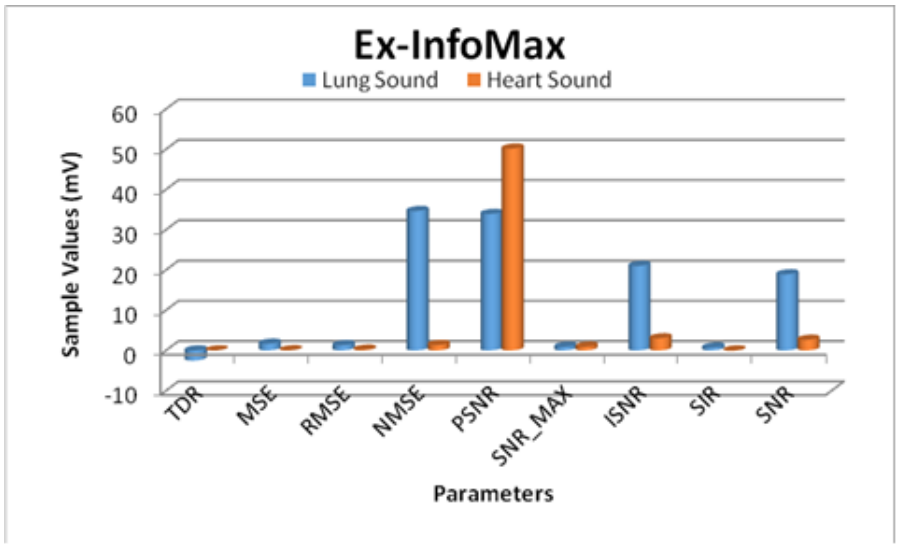
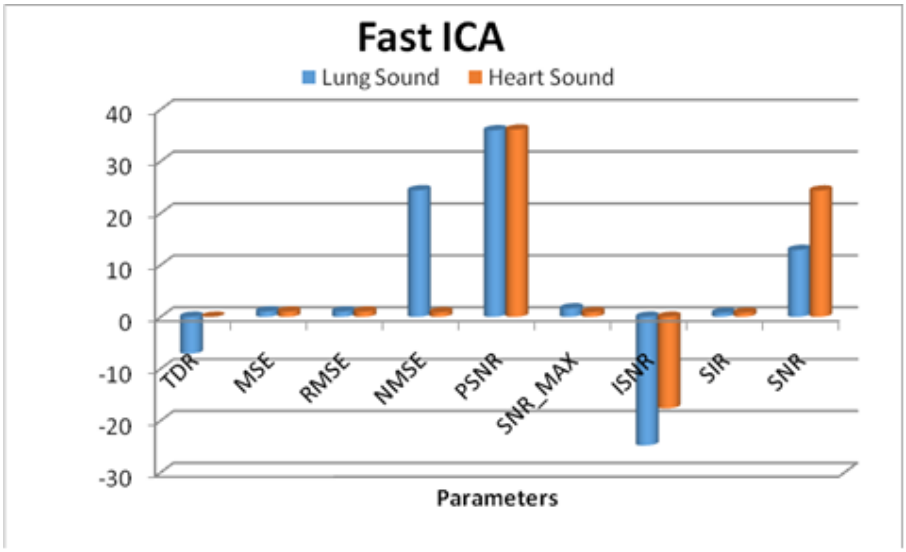


Figure 8: Absolute Error Plots of Algorithms


InfoMax
Lung Sound
Heart Sound
Sample Values (mV)
60
50
40
30
20
10
0
-10
TDR
MSE
RMSE
NMSE
PSNR
SNR_MAX
ISNR
SIR
SNR
Parameters


Ex-InfoMax
Lung Sound
Heart Sound
Sample Values (mV)
60
50
40
30
20
10
0
-10
TDR
MSE
RMSE
NMSE
PSNR
SNR_MAX
ISNR
SIR
SNR
Parameters


Fast ICA
Lung Sound
Heart Sound
40
30
20
10
0
-10
-20
-30
TDR
MSE
RMSE
NMSE
PSNR
SNR_MAX
ISNR
SIR
SNR
Parameters

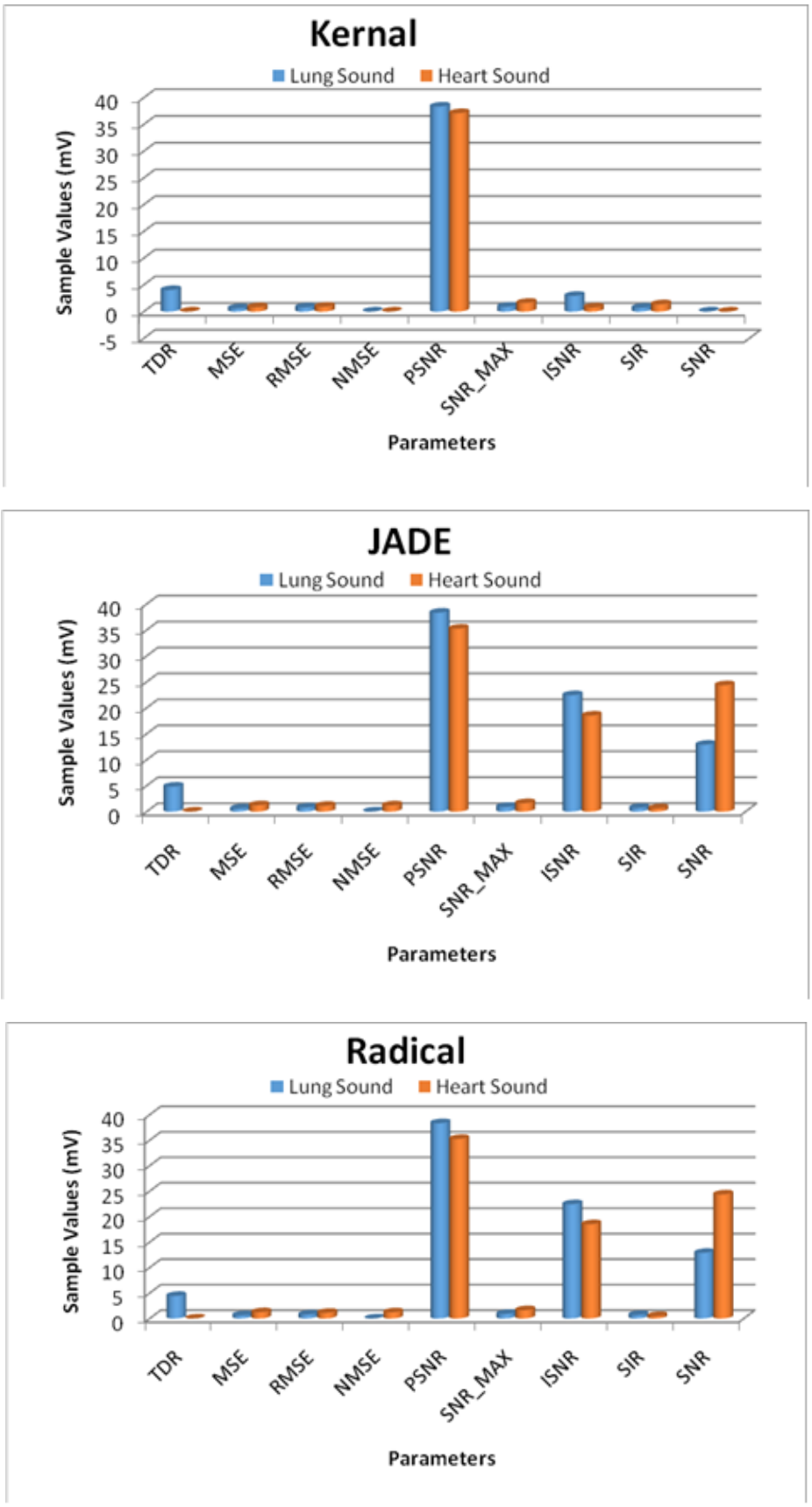


Figure 9: Comparison of the Accuracy of Algorithms - Metrics

Different metrics were involved in this study, as presented in Table 2, which poses a challenge for determining the most successful algorithm for heart-sound separation technique. Performance metrics of machine learning algorithms are calculated for both heart and lung sound consecutively and simulated signals are given in Figures 2-7. Absolute Error and Comparison of the Accuracy of machine learning algorithm is given in Figure 8 and Figure 9 respectively.

It is worth noting that, in order to use the algorithm, a mixed signal is used (Original signal + Mixed Matrix). The output recovered has better signal separation (heart and lung sound signals) besides noisy mixed signals.

From the Table 2 it is clearly pictured that Kernal dominates in MSE, NMSE, PSNR for lung sound whereas infomax shown better values in heart sound separation.

- Kernal algorithm perform best for both Frobenius Error and Amari error whereas Ex-Infomax algorithm for ISNR metrics.
- For SIR and SNR MAX metrics, Fastica for lung sound whereas Kernal shown better values in heart sound separation.
- For SNR metrics, Ex-Infomax for lung sounds whereas JADE and Radical algorithms shown higher impact in heart sounds separation.
- For NMSE Kernal algorithm performs better separation for both lung and heart sounds.

Infomax and Ex-Infomax algorithm:

Infomax algorithm favors higher than Ex-Infomax in case of MSE, RMSE, NMSE, PSNR, SNR MAX, SIR and Frobenius Error. But Ex-Infomax algorithm dominates in ISNR and SNR than Infomax algorithm. Ex-Infomax algorithm is an extended version of Infomax algorithm whose different mixing matrix and mathematical scaling tends to show their improvements in only two metrics (ISNR and SNR). Whereas the algorithm shown lesser efficient in other metrics evaluation when compared to Infomax algorithm. This descript that an extended version is efficient for some metrics only and original Infomax algorithm shown better results for many metrics which proves their efficiency.

Kernal and Fastica algorithm:

- From the Table.2 it is clearly pictured that Kernel algorithm favors higher than Fastica in case of MSE, RMSE, NMSE, PSNR, ISNR, Frobenius Error and Amari Error. But Fastica algorithm dominates only in SNR than Kernel for both lung and heart sound.
- For SNR MAX and SIR, Fastica algorithm for lung sound separation and Kernel-ICA algorithm for heart sound separation performs in an effective manner.
- **Amari error lies in the Range of [0,1].** So Kernal value (0.012) performs best than other algorithms
- Adaptability of algorithms will be varied according to our real time applications and metrics we have chosen to evaluate and estimate.
- For NMSE, infomax algorithm for lung sound separation and Ex-Infomax algorithm for heart sound separation performs in an effective manner.

JADE and Radical algorithm:

- JADE algorithm performs better for Frobenius Error, whereas RADICAL algorithm for Amari error.

- From the Table.2 it is clearly pictured that both the machine learning algorithm algorithms dominate in RMSE, PSNR, ISNR and SNR for lung sound and heart sound separation.
- For MSE and NMSE metrics, Radical for lung sound and JADE for heart sound shown better separation values.
- For SIR metrics, JADE algorithm performs better separation for both lung and heart sounds.
- For SNR MAX metrics, Radical for heart sound separation is higher, whereas both algorithms shown same value for lung sound separation.

Since a primary objective of evaluation of these metrics is used for respiratory sound research and for the selection of best machine learning algorithm for improvements to monitoring and diagnosis of respiratory disease. We can apply these algorithms to clinical application in terms of separation efficiencies.

4. CONCLUSION

In this paper, a detailed comparison among various widely used machine learning algorithm for blind source separation (BSS) was presented. A number of machine learning algorithm approaches have been used for signal analysis, and even more algorithms exist; however, the impact of using different algorithms on the results in auscultation is largely unexplored. This analysis will be used to compare and identify the best strategy for extracting auscultation signal based on the use of machine learning algorithms. While our results are not indicative of finding an optimal solution to the problem, we do feel that we have made progress.

The machine learning algorithms were evaluated in terms of performance metrics. This study opens several lines for future work. Analyzing the existing tradeoffs and evaluation of other metrics for other algorithms are some of the future works of this research. This work can be extended by following the same fashion for other signal analysis and may vary with different engineering applications.

CONFLICT OF INTEREST

Authors declare that they have no conflict of interest.

ETHICAL APPROVAL

All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

REFERENCES:

1) Rajkumar, S., Sathesh, K. & Goyal, N.K. "Neural network-based design and evaluation of performance metrics using adaptive line enhancer with adaptive algorithms for auscultation analysis", *Neural Computing& Applications* (2020). https://doi.org/10.1007/s00521-020-04864-0
2) Qunfang Long, S. Bhinge, Y. Levin-Schwartz, V. D. Calhoun and T. Adalı, "A graph theoretical approach for performance comparison of ICA for fMRI analysis," *51st Annual Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, 2017, pp. 1-6.

3) *B. Sen and K. K. Parhi*, "Extraction of common task signals and spatial maps from group fMRI using a PARAFAC-based tensor decomposition technique," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, 2017, pp. 1113-1117.
4) P. Tillet, H. T. Kung and D. Cox, "Infomax-ICA using Hessian-free optimization," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, 2017, pp. 2537-2541.
5) J. C. Wang *et al*., "VLSI Design for Convolutive Blind Source Separation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 2, pp. 196-200, 2016.
6) E. S. Juan, I. Soto, G. Salinas and P. Adasme, "Separation of VLC signals using FastIca and InfoMax," *First South American Colloquium on Visible Light Communications (SACVLC)*, Santiago, 2017, pp. 1-6.
7) N. Falco, J. A. Benediktsson and L. Bruzzone, "A Study on the Effectiveness of Different Independent Component Analysis Algorithms for Hyperspectral Image Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2183-2199, 2014.
8) L. Breuer, J. Dammers, T. P. L. Roberts and N. J. Shah, "A Constrained ICA Approach for Real-Time Cardiac Artifact Rejection in Magnetoencephalography," IEEE *Transactions on Biomedical Engineering*, vol. 61, no. 2, pp. 405-414, 2014.
9) C. Yuan and J. Zhang, "Extraction of single-trial evoked potentials with Extended Infomax ICA algorithm and its applications to BCI systems," *Proceedings of the 33rd Chinese Control Conference*, Nanjing, 2014, pp. 7139-7144.
10) Wee Lih Lee, Tele Tan, Yee Hong Leung, "An Improved P300 Extraction Using ICA-R For P300-BCI Speller," *35th Annual International Conference of the IEEE on Engineering in Medicine and Biology Socie*ty, 2013, 7064-7067.
11) P. A. Rodriguez, M. Anderson, X. L. Li and T. Adalı, "General Non-Orthogonal Constrained ICA," in *IEEE Transactions on Signal Processing*, vol. 62, no. 11, pp. 2778-2786, 2014.
12) Federico Raimondo, Juan E. Kamienkowski, Mariano Sigman and Diego Fernandez Slezak, "CUDAICA: GPU Optimization of Infomax-ICA EEG Analysis," *Computational Intelligence and Neuroscience*, pp. 1-8, 2012.
13) Lee TW, Girolami M, Sejnowski TJ. "Independent Component Analysis Using an Extended Infomax Algorithm for Mixed Sub Gaussian and Super Gaussian Sources," *Neural Computation*, vol. 11, no. 2, pp. 417-41, 1999.
14) K. Sathesh, S. Rajkumar, Neeraj Kumar Goyal, "Least Mean Square (LMS) based neural design and metric evaluation for auscultation signal separation",Biomedical Signal Processing and Control,vol. 59, pp. 1-7, May. 2020.
15) G. R. Naik, S. E. Selvan and H. T. Nguyen, "Single-Channel EMG Classification with Ensemble-Empirical-Mode-Decomposition-Based ICA for Diagnosing Neuromuscular Disorders," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 7, pp. 734-743, July 2016.
16) M. Chen, X. Zhang, X. Chen and P. Zhou, "Automatic Implementation of Progressive FastICA Peel-Off for High Density Surface EMG Decomposition," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 1, pp. 144-152, Jan. 2018.
17) S. Basiri, E. Ollila and V. Koivunen, "Alternative Derivation of FastICA with Novel Power Iteration Algorithm," in *IEEE Signal Processing Letters*, vol. 24, no. 9, pp. 1378-1382, Sept. 2017.
18) X. Zhao and H. Yang, "A New Method to Calculate the Utility Harmonic Impedance Based on FastICA," in *IEEE Transactions on Power Delivery*, vol. 31, no. 1, pp. 381-388, Feb. 2016.

19) R. Luan, G. Wen, R. Zhang, Z. Chen and Z. Zhang, "Porosity defect detection based on FastICA-RBF during pulsed TIG welding process," *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, Xi'an, 2017, pp. 548-553.
20) S. He, Z. Tong, M. Tong, S. Tang, M. Li and L. Liang, "Research on sound separation and identification of trapped miners based on fastica algorithm," *2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC)*, Macau, 2017, pp. 228-231
21) G. Fontgalland and P. I. L. Ferreira, "Combining Antenna Array Elements by Using ICA Method for Remote Sensing of Sources," in *IEEE Antennas and Wireless Propagation Letters*, vol. 16, pp. 234-237, 2017.
22) L. Cai, X. Tian and S. Chen, "Monitoring Nonlinear and Non-Gaussian Processes Using Gaussian Mixture Model-Based Weighted Kernel Independent Component Analysis," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 1, pp. 122-135, Jan. 2017.
23) Y. Zhang, W. Du and X. G. Li, "Observation and Detection for a Class of Industrial Systems," in *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6724-6731, Aug. 2017.
24) Y. Zhang and Q. Jia, "Complex Process Monitoring Using KUCA with Application to Treatment of Waste Liquor," in *IEEE Transactions on Control Systems Technology*, vol. 26, no. 2, pp. 427-438, March 2018.
25) L. Feng and R. Sun, "Dynamic kernel independent component analysis approach for fault detection and diagnosis," *2017 Chinese Automation Congress (CAC)*, Jinan, 2017, pp. 2193-2197.
26) Q. X. Zhu, Q. Q. Meng, Y. Xu and Y. L. He, "Research and application of KICA-AROMF based fault diagnosis," *2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, Taipei, 2017, pp. 215-220
27) X. Peng, Y. Tian, Y. Tang, W. Du, W. Zhong and F. Qian, "An online performance monitoring using statistics pattern based kernel independent component analysis for non-Gaussian process," *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, Beijing, 2017, pp. 7210-7216.

Effect of Forecasting of Wind Speed with input selection Using Artificial Neural Networks

Sumit Kumar Maitra[1,*], Sumit Saroha[2], Vineet Shekher[3], Mathewos Lolamo[1], Kedir Bashir[5], Priti Prabhakar[2]

[1]*Department of Electrical and Computer Engineering, Wachemo University, P.O.Box 667, Hossana, Ethiopia*
[2]*Department of Electrical Engineering, Guru Jambheshwar University of Science & Technology, Hisar, 125001, India*
[3]*Department of Electrical Engineering, BIT Sindri, Dhanbad, Jharkhand, 828123, India*
**Corresponding author, e-mail: sumit.maitra@gmail.com*

ABSTRACT

At present, deterministic times series simulation values based forecasting is preferred over physical data based wind speed forecasting (WSF). But, it is very difficult to meet out the actual requirements of wind farms because highly uncertain nature of wind speed and its associative parameters data. On the mentioned topic, this presented research develops an improved ensemble time series regression based model for day-ahead local WSF's. The proposed model input has been optimized using regression specifically Auto Correlation Function (ACF). The wind data of Hisar, India collected from National Renewable Energy Laboratory (NREL) has been utilized for the local WSF. In this, Neural Network (NN) with Levenberg Marquardt (LM) learning algorithm and Genetic Algorithms based Neural Networks (GANN) have been adopted for the forecasting simulation purpose. The results has been indicated using simulation by considering the seasonal months WSF's.

Keywords: *Forecasting, Time Series, Neural Network, Wind Speed*

1. INTRODUCTION

The power generated with the help of wind has attracted the attention of lot of academicians worldwide. This is because of random and fluctuating nature of wind speed, which also brings the problems of wind power integration with grid and its conversion. Actually, Wind speed forecasting (WSF) is very helpful of the improvement of safety and economy in order to integrate and convert that into power generation. For that purpose, the generated power must meet the demand of power and power generated by wind speed will change with respect to change in wind speed. The uncertainty in supply of power creates an imbalance between generated power and load demand (requirement) [1], [2].

The intermittent stochastic nature of wind brings lot of challenges to the safe integration and stable & reliable operation of grid. One of the effective remedy is the accurate forecasting of wind speed. The accurate WSF not only help to reduce instantaneous fluctuation of supply voltage but also can be able to adjust power dispatch on real time basis so that stable operation of grid can be ensured. Besides, accurate WSF is also helpful for the improvement in utilization and can increase the economics of wind farm. Therefore, at present the accurate WSF has become most important part of electricity industry.

Zhao et.al [3] has proposed an improved ensemble forecast model for multi step WSF in which physical data of wind and its associative parameters has been utilized. For the pre-processing of data Markov model and for simulation of data weighted average algorithm have been composed. For the improvement of forecasting accuracy of two wind farms in China ref. [4] has proposed an adaptive hybrid model in variational mode decomposition (VMD) pre-process the input data and forecasting simulation has been

done by deep belief network (DBN). Dupr et.al [5] has implemented two different downscaling models on physical data of wind speed. The prediction has been done on hourly basis from 1 h to 11 hrs for "Parc de Bonneval" France using neural networks (NN). Ref. [6] composed of hybrid strategy for WSF in which Apache Spark is applied for dividing the big data into sub group of data. The forecasting simulation is done by using a modified extreme learning machine. Aranizadeh et.al [7] has utilized WSF system for the utilization of capacitor storage system for efficient micro grid operations. WSF for Colonia Eulacio, Uruguay has been performed with the application of NN using hourly time series. The wind data has been collected using anemometer at different heights [8]. Ref. [9] composed of a combined model for WSF of Shandong Peninsula, China in which data has been processed through new pre-processing technique and parameters of NN has been optimized through modified multi-objective optimization. Recurrent NN has also been adopted for accurate time series forecasting of Wind Speed [10]. Similarly, assembled WSF of Xinjiang wind farm, China has been performed by using enhanced particle swarm optimization based algorithm with NN and wavelet based data clustering model [2].

Therefore, the objective of this paper is to investigate the performance of NN using LM learning algorithm for wins speed forecasting. The main contribution of this paper to find out the optimal number of input time lags for a NN model for better forecast. Further, the proposed NN based model with optimum number of input time lags has been implemented for all seasonal wind speed forecast of Hisar, India with one year training data sets. The selected time lags have also been adopted for GA based NN for comparison point of view. First of all the proposed technique is explained in next section in which first section describe the structure of network and then in second section the input has been optimized using regression with training selection Procedure. The wind data of Hisar, India collected from National Renewable Energy Laboratory (NREL) [11] has been utilized for the local WSF discussed with simulation results in Section 3. In the last section, there are conclusive remarks of paper.

2. PROPOSED TECHNIQUE

2.1. Neural Network Model

As suggested by the literature, the most common NN architecture is FFNN for most of the artificial intelligence and forecasting. As per the requirement, the structure of network can also be modified and the structural view of network for the proposed wind speed forecasting (WSF) problem is shown in Figure 1. Proposed network consist of three layered structure in which the input neuron at input layer is defined by auto correlation pre-processed time series of wind speed vectors. The outputs of first layer network are fed to hidden layer and which is processed through the weights & biases and initially which are set to zero. The output vector of hidden layer is processed through activation function and which is used as an input of output layer. Finally, at the end, the overall response has been taken through the output layer of network [9-10]. In the architecture of NN, *n* defines the no. of input neurons, *m* hidden neurons, and there is one neuron at output side of network, the NN model training process is described below:

(I) The output of hidden layer is calculated by using equation (2.1):

$$net_j = \sum_{i=0}^{n} w_{i,j} y_i + b_{ih} (i = 0,1,....,n;\ j = 1,....,m) \tag{2.1}$$

$$z_j = fh(net_j)(j = 1,2,...,m) \tag{2.2}$$

The output value of the j^{th} hidden node is calculated by net_j, where $w_{i,j}$ is the connection weight between input node i and hidden node j, y_i is the i^{th} input data. The output of the j^{th} hidden layer node is calculated by z_j, and f_H is the activation function of hidden layer node as given in equation (2.3), which is a tangent sigmoid transfer function (tansig).

$$f(x) = \frac{2}{1+\exp(-2x)} - 1 \qquad tansig \tag{2.3}$$

$$f(x) = x \qquad purelin \tag{2.4}$$

(II) The overall outputs response is calculated by equation (2.5):

$$F = f_0\left(\sum_{i=0}^{m} w_{j,k} y_j\right)(j = 1,2,...,m) \tag{2.5}$$

In this equation, f_O is the activation function of output layer and the function used here is pure linear transfer function (purelin) as in equation (2.4).

These are the some basic steps involved to perform the wind speed forecasting as listed below:

Step 1: As per the correlation function of time series, the number of input neurons are 6, $WP_{(t-6)}$, $WP_{(t-5)}$, $WP_{(t-4)}$, $WP_{(t-3)}$, $WP_{(t-2)}$, $WP_{(t-1)}$,the hidden neuron involved is twelve and output is processed through one neuron.

Step 2: Levenberg-Marquardt (LM) algorithm has been used for training purpose of network. The activation function at hidden layer is tangential sigmoid and at output layer is pure linear.

Step 3: The maximum number of epochs are equal to 10,000 and the performance goal of network is 0.001.

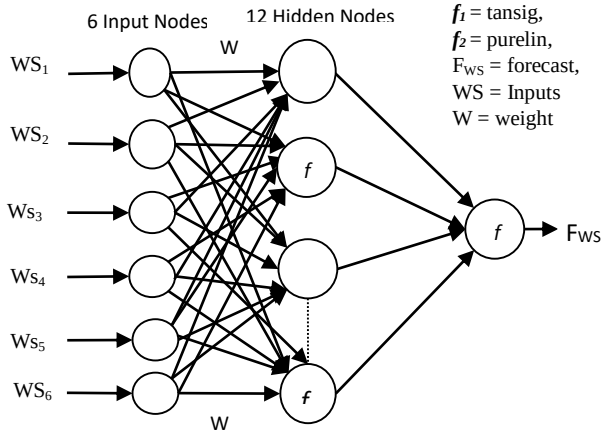


Figure 1: Architecture of Proposed Neural Network model Used

2.2. Genetic Algorithms based Neural Networks (GANN)

Genetic algorithms are a part of growing set of evolutionary algorithm that applies the search principles of natural evolution for parameters optimization. In this work, neural network parameters namely weights are biases are initialized using genetic algorithm [12]. Figure 2 shows the block diagram for the GA overall process for initialization of inputs of NN models. Steps for forecasting is similar to FFNN except that the weights and biases are initialized in this are by using GA.

The processing of GA for optimization of the input parameters of FFNN is shown in Figure 3.6 and the basic steps of genetic algorithms are given below:

- Create a population of chromosomes.
- Chromosomes are evaluated based on their fitness value.
- Based on different selection methods chromosomes are selected for performing genetic operations.
- Genetic operations of crossover and mutation are performed.
- The produced offspring replace their parents in initial population.

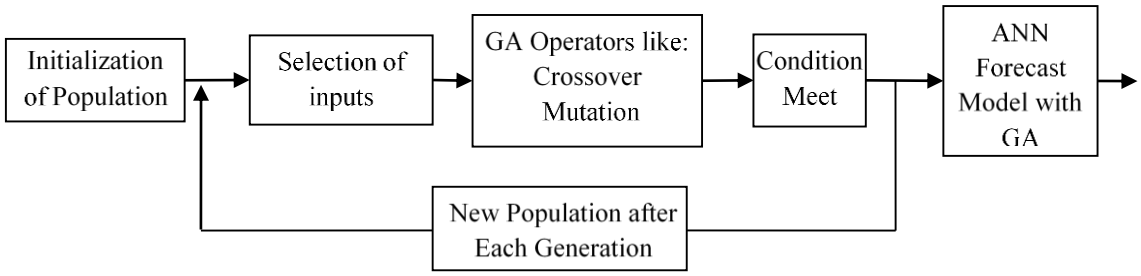


Figure 2: GA process for NN parameters optimization

2.3. Selection of Input and Training Process

For the selection of input parameters an experimental analysis has been initiated in which on the basis of error the number of input neurons are initiated. In Table 1 the number of input vectors with its error indices has been derived. In this, actually the number of time lags has been adjusted with the help of ACF. At serial number six, the error rate is minimum using Mean Absolute Percentage Error (MAPE) indices using six number of wind time series lags. For this experimentation one year training data has been utilized. Figure 3 shows the ACF of wind speed data in which higher values of ACF means higher correlation of each time lag and lower value of ACF means less correlation among each time lag. For the training purpose one month moving window with one year training data has been used as shown in Figure 4. In this, for the WSF of January 2014, one year training data from January 2013 to December 2013 has been used.

Table 1: Input Time Lag Selection on the basis of ACF

S. No.	Time Lag	MAPE
1	$WS_{(t-25)}$, $WS_{(t-24)}$, $WS_{(t-23)}$, $WS_{(t-22)}$, $WS_{(t-7)}$, $WS_{(t-6)}$, $WS_{(t-5)}$, $WS_{(t-4)}$, $WS_{(t-3)}$, $WS_{(t-2)}$, $WS_{(t-1)}$	11.97
2	$WS_{(t-24)}$, $WS_{(t-23)}$, $WS_{(t-22)}$, $WS_{(t-7)}$, $WS_{(t-6)}$, $WS_{(t-5)}$, $WS_{(t-4)}$, $WS_{(t-3)}$, $WS_{(t-2)}$, $WS_{(t-1)}$	11.97
3	$WS_{(t-23)}$, $WS_{(t-22)}$, $WS_{(t-7)}$, $WS_{(t-6)}$, $WS_{(t-5)}$, $WS_{(t-4)}$, $WS_{(t-3)}$, $WS_{(t-2)}$, $WS_{(t-1)}$	12.98
4	$WS_{(t-22)}$, $WS_{(t-7)}$, $WS_{(t-6)}$, $WS_{(t-5)}$, $WS_{(t-4)}$, $WS_{(t-3)}$, $WS_{(t-2)}$, $WS_{(t-1)}$	12.49
5	$WS_{(t-7)}$, $WS_{(t-6)}$, $WS_{(t-5)}$, $WS_{(t-4)}$, $WS_{(t-3)}$, $WS_{(t-2)}$, $WS_{(t-1)}$	12.56
6	**$WS_{(t-6)}$, $WS_{(t-5)}$, $WS_{(t-4)}$, $WS_{(t-3)}$, $WS_{(t-2)}$, $WS_{(t-1)}$**	**11.96**
7	$WS_{(t-5)}$, $WS_{(t-4)}$, $WS_{(t-3)}$, $WS_{(t-2)}$, $WS_{(t-1)}$	11.97
8	$WS_{(t-4)}$, $WS_{(t-3)}$, $WS_{(t-2)}$, $WS_{(t-1)}$	12.24
9	$WS_{(t-3)}$, $WS_{(t-2)}$, $WS_{(t-1)}$	12.34
10	$WS_{(t-2)}$, $WS_{(t-1)}$	13.28
11	$WS_{(t-1)}$	18.36

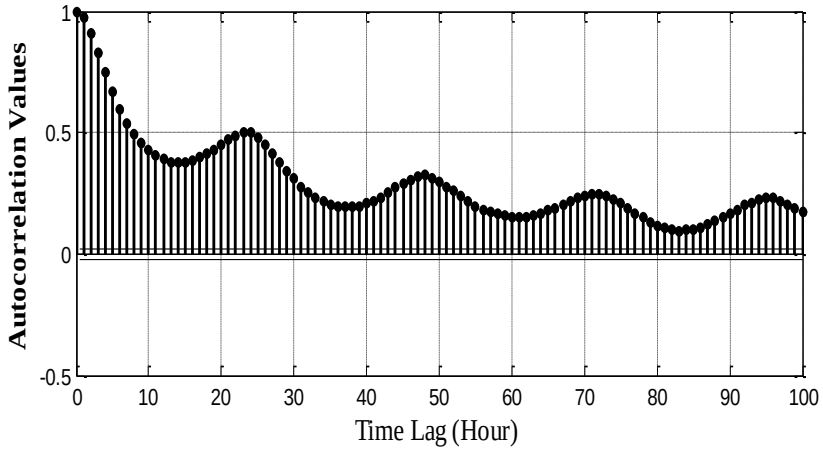


Figure 3: The ACF of Wind Speed data

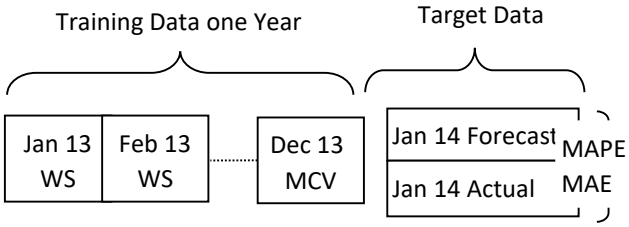


Figure 4: Training Data Used Process

2.4. Accuracy Assessment

For the analysis of results of proposed model, there are two accuracy indices used namely: mean absolute percentage error (MAPE) and mean absolute error (MAE). In this, the actual value vector is WS_t which is compared with forecasted values of wind speed obtained from NN model. Mathematically, MAPE and MAE can be written as:

$$MAPE = \frac{1}{n}\sum_{n=1}^{t}\left|\frac{(WS_t - F_t)}{WS_t}\right| \quad (2.6)$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|WS_t - F_t| \quad (2.7)$$

In both the equations, actual value of WS is indicated by WS_t and the forecasted value of WS is indicated by F_t for t^{th} hour in which the total forecasted number of hours considered to be n.

3. SIMULATION RESULTS

The WSF model accuracy has been affected by the geographical characteristics of wind farms different location. The focus of this section is to forecast wind speed on very short term time spam basis and which is actually the objective of this research work. The hourly time series data of Wind Speed has been used in order to evaluate the performance of presented forecasting models. Two years historical wind speed data between a time period of 2013 and 2014 [11] is collected from NREL for Hisar, Haryana, India. No data has been struck out from the overall data collected such as: holiday or any other special day.

Indian Seasonal Time Period

The performance of FFNN has been analysed on accuracy, simulation time and regression of forecasts. For testing of WSF results using proposed model, four Indian seasonal months [13] have been considered as given below in Table 2. The results of WSF on all accuracy measures have been obtained for the month of every season. As per table 3, it has also been analysed that higher is the accuracy of forecast higher is the regression coefficient.

The forecasting results on MAPE and MAE accuracy indices have been given below in table 3 and table 4 for FFNN and GANN respectively. The average accuracy achieved on MAPE & MAE is 8.62 % & 0.115 m/s respectively using FFNN and average accuracy achieved on MAPE & MAE by using GANN is 8.62 % & 0.115 m/s respectively. The maximum value of MAPE (11.9 %) is observed in dry season for the month of October and the minimum value of MAPE (5.93%) observed in the midway of summer and rainy season for the month of June. The 24 hours forecasting error with actual and forecasted values of WS for the first December is given below in Table 5. It has been observed that both models (FFNN & GANN) performed almost similar to each other and there are slight variations in the accuracy and other forecasting parameters. Thus, it is concluded that the proposed time lags selection technique is fit for the neural networks based models for wins speed forecasting purpose.

Table 2: Indian Seasonal Months

Season	Period	Testing Period
Summer	April-May	April
Rainy	June-September	June
Dry	October-November	October
Winter	December- March	December

Table 3: Seasonal WSF Estimated Results using FFNN

	MAPE	MAE	Simulation Time	Regression
April	9.01	0.12	15.211869	0.9772
June	5.93	0.14	26.083274	0.9883
October	11.9	0.1	11.87503	0.9717
December	7.64	0.1	20.975049	0.9845
A.V.	8.62	0.115	18.5363055	0.980425

Table 4: Seasonal WSF Estimated Results using GANN

	MAPE	MAE	Simulation Time	Regression
April	9.49	0.13	17.74303	0.9756
June	5.67	0.14	38.884627	0.9886
October	11.59	0.1	13.808296	0.9721
December	7.75	0.1	17.24818	0.9843
A.V.	8.625	0.1175	21.92103325	0.98015

Table 5: 24 hours Ahead WS Estimation Accuracy with Actual Error

		FFNN		GANN	
	Actual	Predicted	Error	Predicted	Error
1	2.713276	2.6584055	0.0548707	2.667449	0.045827
2	2.795315	2.7530006	0.0423147	2.767642	0.027674
3	2.685155	2.8043301	-0.119175	2.815134	-0.12998
4	2.62174	2.506828	0.1149118	2.495376	0.126364
5	2.665204	2.5700228	0.0951814	2.588147	0.077057
6	2.69971	2.663595	0.0361154	2.684783	0.014928
7	2.680027	2.6893557	-0.009329	2.689611	-0.00958
8	2.625622	2.6420994	-0.016478	2.635949	-0.01033
9	2.304487	2.5571717	-0.252685	2.551209	-0.24672
10	1.939406	1.999621	-0.060215	2.035505	-0.0961
11	2.619833	1.7079521	0.9118813	1.697286	0.922547
12	3.215085	3.0317069	0.1833777	2.894419	0.320665
13	3.383251	3.5088758	-0.125625	3.562028	-0.17878
14	3.627977	3.354442	0.2735354	3.386182	0.241795
15	3.881531	3.8662011	0.0153302	3.930051	-0.04852
16	4.036038	3.9537623	0.0822756	3.964404	0.071634
17	3.734478	4.0214963	-0.287018	4.032214	-0.29774
18	3.737064	3.3134966	0.4235677	3.41573	0.321334
19	3.796801	3.838249	-0.041448	3.90928	-0.11248
20	3.841455	3.7145685	0.126886	3.719732	0.121723
21	3.830432	3.8365387	-0.006106	3.860076	-0.02964
22	3.713225	3.8241265	-0.110902	3.770708	-0.05748
23	3.593368	3.5670962	0.0262716	3.543409	0.049958
24	3.475042	3.4764984	-0.001456	3.473656	0.001386
A.V.			0.0565035		0.046898

The actual and forecasted WS curves using FFNN for all Indian seasons first week has been shown in Figure from 5 to Figure 8.

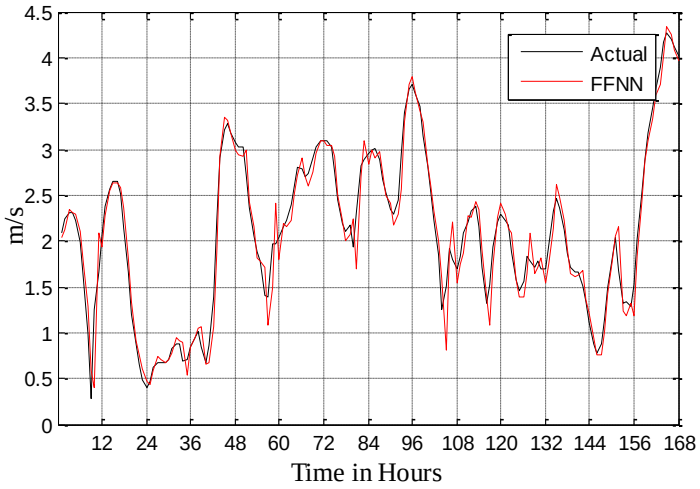


Figure 5: Actual and Forecasted WS curves for the April First week

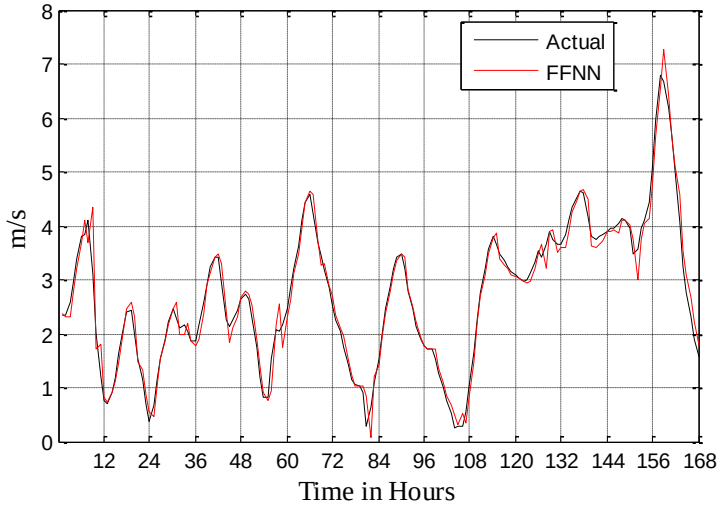


Figure 6: The Actual and Forecasted WS curves for the June First week

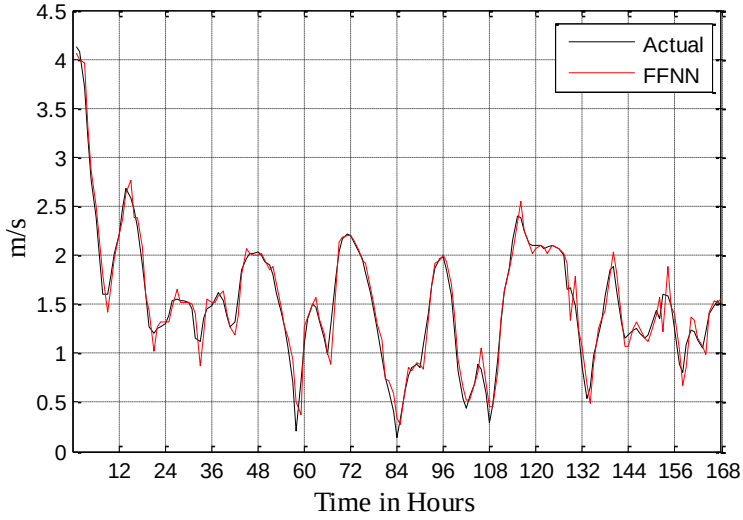


Figure 7: The Actual and Forecasted WS curves for the October First week

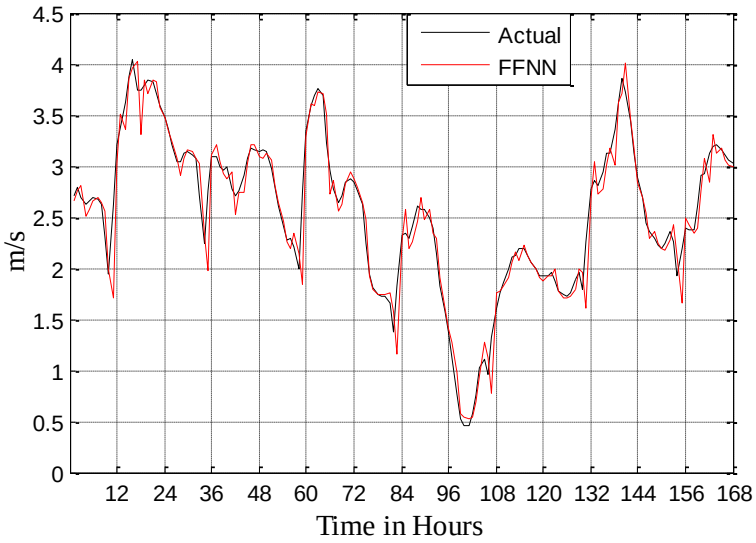


Figure 8: The Actual and Forecasted WS curves for the December First week

4. CONCLUSION

The contribution of this research is to develop an improved time series based WSF simulation model. The proposed has utilized the two years wind speed data of Hisar, India in which seasonal month forecasting of wind speed has been performed. For the simulation purpose one year training data is used as training set for one month forecasting. On the basis of experimental analysis inputs to the NN has been optimized and MAPE of 5.93 is achieved. It has also been observed that the optimization of input data is highly importance for a forecasting model and the performance of proposed ACF based time lags selection technique is satisfactory. Moreover, for real life applications, difficulties or error in forecast may occur due to the manner in which data is utilized and the simulation capability of forecasting model.

REFERENCES

1. Li Z, Wu W and Zhang B, "Adjustable robust real-time power dispatch with large scale Wind power integration, IEEE Trans Sustainable Energy, vol. no. 6, pp. 357-682, 2015.
2. Zhu Duan and Hui Liu, "An evolution-dependent multi-objective ensemble model of vanishing moment with adversarial auto-encoder for short-term wind speed forecasting in Xinjiang wind farm, China", Energy Conversion and Management, vol. no. 198, pp. 111914, 2019.
3. Jing Zhao, Jianzhou Wang, Zhenhai Guo, Yanling Guo, Wantao Lin, Yihua Lin, "Multi-step wind speed forecasting based on numerical simulations and an optimized stochastic ensemble method", Applied Energy, vol. no. 255, pp. 113833, 2019.
4. Jinliang Zhang, Yiming Wei and Zhongfu Tan, "An adaptive hybrid model for short term wind speed forecasting", Energy, vol. no. 190, pp. 115615, 2020.
5. Aurore Dupr, Philippe Drobinski, Bastien Alonzo, Jordi Badosa, Christian Briard and Riwal Plougonven, "Sub-hourly forecasting of wind speed and wind energy", Renewable Energy, vol. no. 145, pp. 2373-2379, 2020.
6. Yinan Xu, Hui Liu and Zhihao Long, "A distributed computing framework for wind speed big data forecasting on Apache Spark", Sustainable Energy Technologies and Assessments, vol. no. 37, pp. 100582, 2020.

7. Aranizadeh, A. Zaboli, O. Asgari Gashteroodkhani and B. Vahidi, "Wind turbine and ultra-capacitor harvested energy increasing in microgrid using wind speed forecasting", Engineering Science and Technology, an International Journal, vol. no.22, pp. 1161-1167, 2019.
8. P.J. Zucatelli, E.G.S. Nascimento, G.Y.R. Aylas, N.B.P. Souza, Y.K.L. Kitagawa, A.A.B. Santos, A.M.G. Arce and D.M. Moreira, "Short-term wind speed forecasting in Uruguay using computational intelligence", Heliyon, vol. no. 5, pp. e01664, 2019.
9. Zhenkun Liu, Ping Jiang, Lifang Zhang and Xinsong Niu, "A combined forecasting model for time series: Application to short-term wind speed forecasting", Applied Energy, Article in press.
10. Ignacio A. Araya, Carlos Valle and Héctor Allende, "A Multi-Scale Model based on the Long Short-Term Memory for day ahead hourly wind speed forecasting", Pattern Recognition Letters, Article in press.
11. https://www.nrel.gov/
12. D Liu, D Niu, H Wang, L. Fan, "Short-term wind speed forecasting using wavelet transform and support vector machines optimized by genetic algorithm", International Journal of Renewable Energy, vol. no. 62, pp. 592-597, 2014.
13. L M Saini and M K Soni, "Artificial neural network-based peak load forecasting using conjugate gradient methods", IEEE Transactions on Power Systems, vol. 17, no. 3, pp. 907-912, August 2002.

Machine Learning Approach for Green Usage of Computing Devices

Mulualem Bitew Anley, Rediet Bereket Awgichew

College of Informatics department of Information Systems, University of Gondar, Gondar, Ethiopia
**Corresponding author, e-mail: mulualem.bitew@uog.edu.et*

ABSTRACT

Information Communication Technology (ICT) is an engine and source of development for the better and modern lifestyle of today's society. With the number of devices is increase the problem is tremendously increase. Among those factors improper usage contributes high portion. Managing user's computer usage patterns and computer power properly reduces the emission of CO_2 (carbon dioxide) and power consumption. This study used server user access log files and joule meter power reader data for the machine learning model building. Based on the data, an experiment has been carried out to classify the power consumption patterns of the user to make the intelligent power mode decision of the device. The study used python for data analysis, model establishment, and experiment. The experiment was done using five machine learning algorithms. From those algorithms, the SVM machine learning algorithm classified 99.6% the power state of the computers correctly.

***Keywords**: Machine Learning, Green Usage, Machine learning for green computing, Carbon emission, Power Consumption, Computing devices power consumption*

1. INTRODUCTION

Information Communication Technology (ICT) is becoming a major building block of modern society and contributing to enable people to live a better life, in making organizations productive and making the communication or interaction of people much easier (Asadi, Dahlan, et al., 2017). ICT has different definitions in different disciplines and it is a bit difficult to give a precise definition for the term ICT (Zuppo, 2012). However, for this study, we used Zhang's (Zhang et al., 2008) definition of ICT as a working definition that is "ICT is the application of science to the processing of data according to programmed instructions in order to derive results. In the widest sense, ICT includes all communications, information and related technology". Devices that are referred to as ICT or computing devices are PCs, desktops, laptops, handheld devices and other types of wireless or cable-connected equipment (Zuppo, 2012).

Computing devices' electric power consumptions are estimated to be 50 percent of the global electric power consumption in the coming 2030 (Technologies, 2015). High power is going to consume by the computing devices this is the high carbon emissions indirectly facing the health problem to give high attentions.

Green use refers to reducing the energy consumption related to ICT equipment and using them in an environmentally sound manner (Shuja et al., 2017). Minimizing the power consumed by computing devices is a major solution for reduction of carbon dioxide emissions and their impact on environment and global warming (Murugesan, 2008). According to (Uddin et al., 2017) more than 75 percent of computer are switch on without doing nothing and consuming power. Therefore, this study targets to apply a machine learning approach to minimize the idle computers left ON without affecting the performance on services of the user.

Power consumption and greenhouse gas emission are parallel (Murugesan, 2015). When the amount of electric power consumed by computing devices is minimized so does the CO_2 emission and its impact on our environment (Murugesan, 2008). An individual computer that is in use can produce a ton of CO_2 every year (Murugesan, 2008). Reducing energy consumption is equal to reducing greenhouse gas emission and reducing operational costs for ICT sector (Nordman et al., 1997). Increasing in greenhouse gas emission results in climate change and global warming which is becoming a very serious and causing damage on our country, continent and planet. The CO_2 emission of the ICT sector throughout the world is increasing from time to time. Among CO_2 emissions of different computing devices, the first rank is the CO_2 emission of PCs and monitors which is 40% from the whole computing devices (Asadi et al., 2017). This percentage calls the attention of the world to work towards greening the design, production, usage and disposal of computing devices especially personal computers and monitors.

An energy-efficient computer that is always powered-on consumes more energy than a less energy-efficient computer that is regularly turned off. Looking at typical usage patterns of computers and monitors provides a clearer picture of how their total energy consumption can be reduced than simply looking at energy requirement (Han & Gnawali, 2012). Usage patterns are typically separated into three period's nighttime usage, weekends, and daytime usage. From this research on average, only less than one-third of all computers and monitors are turned off at night. According to the data we collected at the University of Gondar more than seventy-five percent of the laboratory, library and office computers are switch on without doing anything.

Machine learning is a sub-components of Artificial intelligence that use mathematical and statistical techniques to analysis data. Machine learning is a data- driven approach that builds a model to predict or support decisions based on the pattern they learn from data. For efficient power management and energy efficiency machine learning is highly applied on data centers. Machine learning contributes a significant impact on the green usage of computing devices. Besides on this study made an effort in using a machine learning algorithm model to classify the computer power state by learning from the data.

2. LITERATURE REVIEW

Green use focuses on how Information Technology (IT) can be green based on our computer usage mechanisms. Making IT green in usage patterns is about reducing the energy consumption of computers and using them without causing any impact or minimized impact on the environment. Green use refers to the application of computers and other computing devices in an environmental friendly way to reduce energy consumption (Tiwari, 2012).

Greening IT systems start from the designing part and the design that serves as an input for greening the manufacturing process which is the next step. Then the greening comes to usage of computers injudicious way which includes purchasing certified products, using different power-saving techniques and using usage manuals. Then finally disposing of old or unwanted computers using 3 R's that are reusing, refurbishing and recycling can lead to greener IT. Reusing is all about using old computers by replacing a functional component from other retired devices while refurbishing is upgrading old computers by replacing their parts

by buying a new components from the market. After considering reuse and refurbishing, if the computer cannot be used the last measure is recycling it properly (Murugesan, 2008).

There are many potential approaches of green computing in which they can be applied to minimize the environmental impact of ICT. These approaches can be referred to as approaches for green computing. There are different approaches in green computing some of them are product longevity, algorithm efficiency, reduce energy consumption, sleep and hibernate mode, shutdown and server virtualization. In different pieces of literature, scholars have been referring the way towards green computing or ways that enable green IT as approaches, countermeasures or strategies which all have the same intention with a similar concept.

On green usage of the computer system sleep mode and hibernate are the best options designed to save power consumption of computers. In the case of sleep mode, the work and the settings are saved in memory whereas hibernate mode puts the open programs and documents in the hard disk. From the two power management features, hibernate can save more power than sleep mode. According to different studies sleep mode can save up to 80% of power consumption while hibernate saves energy up to 96% (Tiwari, 2012). Most users do not want to shut down their computer since it can make them wait some time. So they prefer simply leaving it "ON" which is a major cause for waste of power. When we think of the energy that can be saved by shutting down computers while we are not using them, it really worth waiting.

There are some studies that are available in the area of green computing. Starting from 1992 most respected scholars, big companies and individuals has been given much attention to green computing or green IT. Another study by Mann et al. proposed an implementation framework aimed at helping organizations in determining the extent to which they should consider investing in green IT to get benefit out of it. The practical implementation strategies are data center reconfiguration, energy saving computers, software upgrades and recycling. The researcher used a feedback loop as an assessment tool. A study by Molla proposed a green IT readiness framework that enable organizations to implement or deploy environmentally sustainable IT systems. As per the researcher, practice readiness refers how the company is ready to convert policies into action. Another component or concept of the framework is technology which is about acquiring greener technologies.

A study by Girma Tememe (2014) investigated operating system level power management methods using visual studio 2010 sp1 and SQL server 2008 R2 tools. They have declared that 34.61% of energy was saved on the provided solution. However, according to Lin et al. (2016) machine learning recorded high accuracy results on many decision support systems so does on power management.

Machine learning has a great contribution on different area including climate change control and CO_2 emission reductions. Machine learning aims in designing and developing of algorithms that can solve severe problems by learning from data. Machine learning can support the identification of complex patterns, support understanding the cause of events and recommend efficient solutions based on the model built from data. Now a day's machine learning is applied in different complex application areas. As well as tackling climate change and carbon emission reduction (Rolnick et al., 2019).

Green usage of ICT devices is highly minimizing the carbon emission and power conceptions of the lab and data center computers. According to Han & Gnawali (2012) green practices in the data center, on desktop pc and on printing devices needs technological solutions to reduce carbon emissions. They also propose a framework for ICT device carbon footprint. However, the framework is needs to be more in practical and in user behaviors of the device are more effective.

In a study Lin et al. (2016) designed and developed a reinforcement learning based framework for green computing to the data centers to reduce the power conception of data centers and a green computing power solution for computer use. The designed system was server pool average energy conception with reasonable job response time for level power management solution. A study by Saleh et al. (2016) used SVM to predict the carbon emission of the device. Results indicate that SVM machine learning algorithm can be effective to predict the carbon emission of the device. However, the issues need more practical applications than predictions.

Moreover, according to Rolnick et al. (2019) machine learning is contributing too much to tackling the greenhouse gas emissions in different sectors so does computer laboratory- based user behaviors. Therefore this study used a machine learning based power mode classification model. The idea is the model learns from the server user access log file which is collected from the datacenter and the power conceptions of the device collected by using the joule meter software. The model then supports the decision of the operating system to sleep, hibernate or shut down by indicating the user accessing behaviors.

3. METHODOLOGY

3.1. Dataset and Preprocessing

In this study, we have used University of Gondar (UoG) laboratory computers and servers on the data center. In UoG there are more than 7,000 Desktop computers, 2000 Laptops and more than 60 Physical and Virtual Servers. The number is increase time to time that the university moving forward to E-University. For this study, we have collected data from the computer in which the users are staff, student and library users. As well the data was collected on three-time frames during lunch, day and night.

Joule meter software was used to collect the computer's power usage status. We have also collected the users' access log files from the data center. Joule meter software was installed to measure the power consumption of the devices based on this tool was installed on 60 computers in which computers are selected from the three types of users that are staff, students and library users randomly. In three months, more than 76 million log files and the power log data is collected from the three time range of the day which are lunchtime, night time and the rest of the day.

We have used server access log files data to track user accessing sessions. The data that integrates with the joule meter data with the users accessing sessions and computer. The data set comprises information collected from multiple users from different brands and models. After the data is preprocessed and converted into CSV file format, we have used the data for the experiments.

4. EXPERIMENTS

After the data preparation task is completed and prepared to make it suitable for the python machine learning algorithm, we continued the experiment to build a classification model. To do that, the researcher

conducted eight different experiments on the server log files and power consumption record datasets to select the best model. Those experiments were done by using python 3.3 using the machine learning classifier algorithms. In the experimentation, we have separated the data as a training set and test set. The default value of parameters is planned for every classifier algorithm since it permits accomplishing better precision compared with changing the default parameter values.

Among the machine learning classification algorithms used for this study, supervised learning algorithms which are decision tree, logistic regression, nave Bayes, Support Vector Machine (SVM) and k-Nearest Neighbors are used for the experiment. The following sections discuss the features of machine learning algorithms that can apply to this effort.

The overview of the machine learning model to power state classification is depicted on Figure 1.

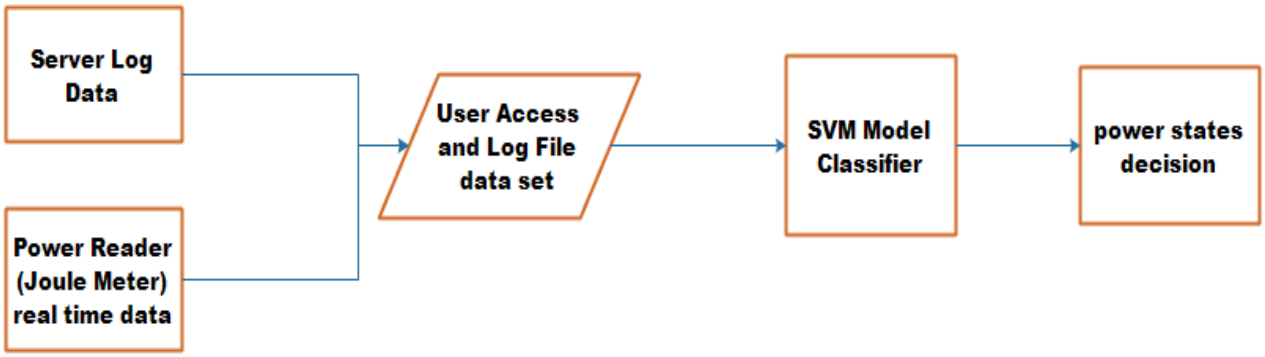


Figure 1: An overview of the SVM based green usage of an ICT device

The server log files is used to manage power on a device in a method adjusted to the individual user data about how that user utilizes their device must be captured. This is done by using server log file and joule meter real-time power consumption data. The integrated dataset that was collected has more supported the SVM real time decision. SVM model classifier is an intelligent software component that run and install in the background process. Power state decision is done by classifier decision support and the operating system.

4.1. Computer Power Usage Pattern

As shown on Figure 2, more power consumptions are recorded on a desktop computer than a laptop. This is due to the different usage patterns of laptop and desktop computers. Whereas desktop computers spend most of the time in active mode, laptop computers spend most of the time in low power mode. It is assumed that desktops and laptops spend the same amount of time when they are turned off.

Figure 3 shows that the high power consumption is in the daytime of the desktop computers. This is because the computers are active as well different applications and programs are executed. As the night time desktop computers are active and consumes power on their idle time

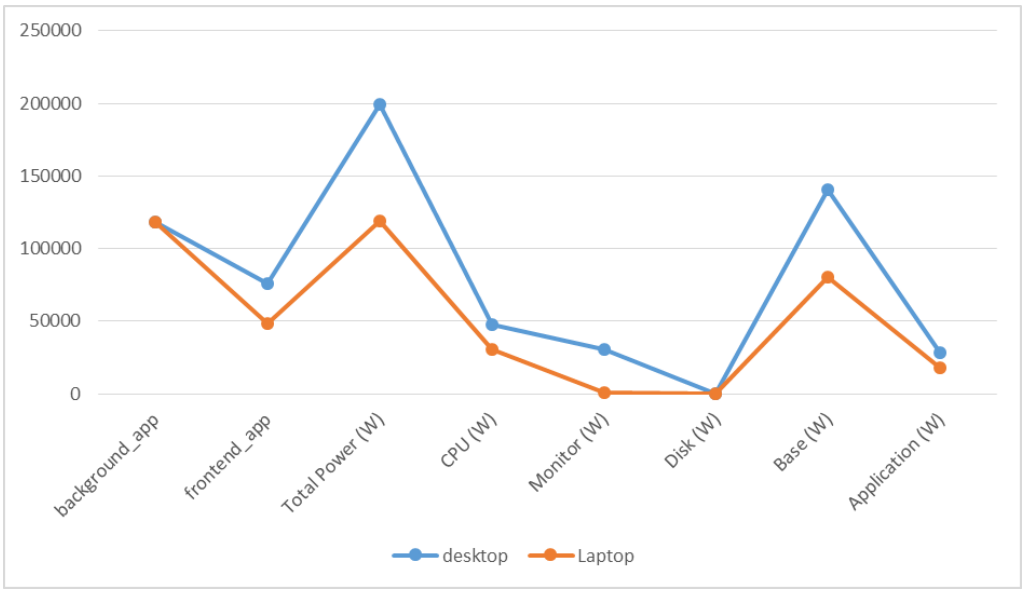


Figure 2: Power conception per computer type (Source: Collected Data)

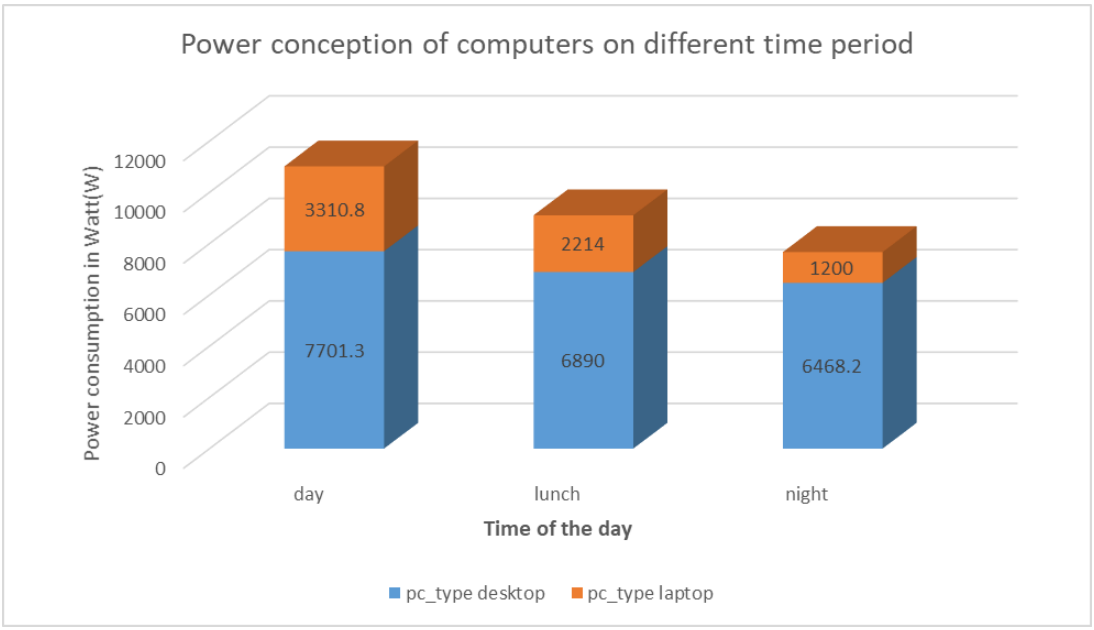


Figure 3: Power consumption of the computer device on the different time group

5. RESULT AND DISCUSSION

To select the best model for the power model adjustment and saving the power, the decision tree, logistic regression, naïve Bayes, support vector machine and KNN classifier approaches using training set, test set, cross-validation (10-folds) and percentage split (66%) were used for conducting experiments. We have used 80 percent for model training and 20 for testing from the total dataset. The summary of experimental results for those classification algorithms is presented in Table 1 below.

This experiment was done by using a full training set, testing set option belongs on python Jupiter notebook software. The logistic regression algorithm correctly classified (98.85 %) and incorrectly classified are (1.15 %) from the training set. KNN algorithm correctly classified (98.81%) and incorrectly

classified are (1.29 %) from the training set CART algorithm correctly classified (99.3 %) and incorrectly classified are (0.7 %) from the training set. NB algorithm correctly classified (86.8%) and incorrectly classified are (13.2%) from the training set and SVM algorithm correctly classified (99.4%) and incorrectly classified are (0.6%) from the training set Detail of the resulting model accuracy by class depicts on Table 1. The detail of the correctly classified, incorrectly and accuracy experiment detail results were found on Table 1.

Table 1: Experiment Result of the Algorithm

Algorithm	Correctly classified	Incorrectly classified	Accuracy by %
Logistic regression	0.985377	0.011341	98.85%
KNN	0.981092	0.017090	98.81%
CART	0.93160	0.007154	99.3%
NB	0.868718	0.039390	86.8%
SVM	0.994116	0.009693	99.4%

In algorithms evaluation, from above Table 1, it is clearly observed that SVM algorithm better than other algorithms. As a result, it is reasonable to conclude that SVM algorithm is better than other algorithms for this machine learning based decision support system for computer device carbon emission reductions and power consumption reduction.

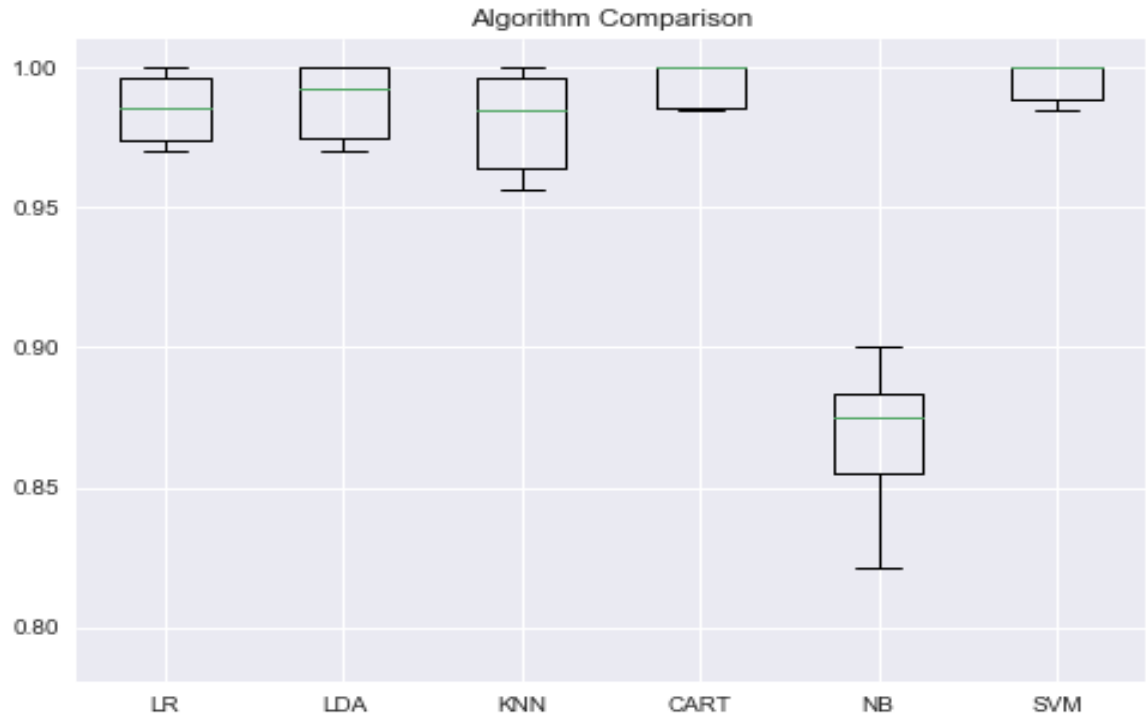


Figure 4: Machine learning algorithm comparison

Therefore, the model which is developed the SVM with training set test option classification techniques is considered as the selected working model for the next use in the development of intelligent power state decision of the computing devices.

6. CONCLUSION AND RECOMMENDATION

This research obtained that machine learning algorithms have an effort on power management in computing devices by considering the user log and power consumption of the devices. An analysis of the computer type power consumption and how this affects carbon emissions is also discovered. The detailed power consumption of the device on time frame of the day also carried out.

To assess the effectiveness of the model, we have developed java user interface that used SVM machine learning model prototype. We have installed these prototype software on 10 computers in the laboratory. After implementing the prototype and power measurement on a similar computer, around 10 kW power is saved per week on average and then 520 kW per year on each computer. Therefore, this proposed SVM machine learning based approach has able to save 520 kW energy per year from a single computer.

Based on the above learnings, a custom solution was designed and developed to maximize energy savings and carbon emission reduction in computer devices. The machine learning model proposed an SVM classifier to classify the power state. The model was evaluated by the precision, recall and f-measure and obtain 99.6% accuracy. We conclude that machine learning algorithms based on green usage have the potential to increase power saving and reduce the carbon emission of computing devices.

REFERENCES

Asadi, S., Dahlan, H. M., & others. (2017). Organizational research in the field of green IT: A systematic literature review from 2007 to 2016. *Telematics and Informatics*.

Asadi, S., Hussin, A. R. C., & Dahlan, H. M. (2017). Organizational research in the field of Green IT: A systematic literature review from 2007 to 2016. *Telematics and Informatics*, *34*(7), 1191–1249. https://doi.org/10.1016/j.tele.2017.05.009

GIRMA TEMEME TAKELE. (2014). *Green Computing Power Solution in Computers Use: The case of University of Gondar*. University of Gondar.

Han, D., & Gnawali, O. (2012). Understanding desktop energy footprint in an academic computer lab. *Proceedings - 2012 IEEE Int. Conf. on Green Computing and Communications, GreenCom 2012, Conf. on Internet of Things, IThings 2012 and Conf. on Cyber, Physical and Social Computing, CPSCom 2012, November 2012*, 541–548. https://doi.org/10.1109/GreenCom.2012.77

Lin, X., Wang, Y., & Pedram, M. (2016). A reinforcement learning-based power management framework for green computing data centers. *Proceedings - 2016 IEEE International Conference on Cloud Engineering, IC2E 2016: Co-Located with the 1st IEEE International Conference on Internet-of-Things Design and Implementation, IoTDI 2016*, 135–138. https://doi.org/10.1109/IC2E.2016.33

Murugesan, S. (2008). Harnessing green IT: Principles and practices. *IT Professional*, *10*(1), 24–33. https://doi.org/10.1109/MITP.2008.10

Murugesan, S. (2015). *Harnessing Green IT : Principles and Practices*. *March*. https://doi.org/10.1109/MITP.2008.10

Nordman, B., Piette, M. A., Kinney, K., & Webber, C. (1997). User guide to power management for PCs and monitors. *Environmental Energy Technologies Division, Lawrence Berkeley National Laboratory, University of California*, *August 2014*, 5. https://doi.org/10.2172/486126

Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., Ross, A. S., Milojevic-Dupont,

N., Jaques, N., Waldman-Brown, A., Luccioni, A., Maharaj, T., Sherwin, E. D., Karthik Mukkavilli, S., Kording, K. P., Gomes, C., Ng, A. Y., Hassabis, D., Platt, J. C., … Bengio, Y. (2019). Tackling climate change with machine learning. *ArXiv*.

Saleh, C., Dzakiyullah, N. R., & Nugroho, J. B. (2016). Carbon dioxide emission prediction using support vector machine. *IOP Conference Series: Materials Science and Engineering*, *114*(1). https://doi.org/10.1088/1757-899X/114/1/012148

Shuja, J., Ahmad, R. W., Gani, A., Abdalla Ahmed, A. I., Siddiqa, A., Nisar, K., Khan, S. U., & Zomaya, A. Y. (2017). Greening emerging IT technologies: techniques and practices. *Journal of Internet Services and Applications*, *8*(1). https://doi.org/10.1186/s13174-017-0060-5

Technologies, H. (2015). *Trends to 2030*. 117–157. https://doi.org/10.3390/challe6010117

Tiwari, S. (2012). Need of Green Computing Measures for Indian IT Industry. *Journal of Energy Technologies and Policy*, *1*(4), 18–25. http://www.iiste.org/Journals/index.php/JETP/article/view/1186

Uddin, M., Okai, S., & Saba, T. (2017). Green ICT framework to reduce carbon footprints in universities. *Advances in Energy Research*, *5*(1), 1–12. https://doi.org/10.12989/eri.2017.5.1.001

Zhang, P., Aikman, S. N., & Sun, H. (2008). Two types of attitudes in ICT acceptance and use. *Intl. Journal of Human--Computer Interaction*, *24*(7), 628–648.

Zuppo, C. M. (2012). Defining ICT in a boundaryless world: The development of a working hierarchy. *International Journal of Managing Information Technology*, *4*(3), 13.

Examining Data Mining Techniques to Analyze Outbreak Surveillance and Response System: In case of Ethiopia

Yimer Mohammed

Addis Ababa University, Addis Ababa, Ethiopia
E-mail: yimoh_fast@yahoo.com

ABSTRACT

In the past, when sanitary conditions were poor, lifestyles were very traditional, and diseases were little understood, epidemics occurred periodically and killed thousands of people especially in the tropics, including Ethiopians. In Ethiopia, there are public health sectors that work all over the country, but due to lack of adequate performance assessment and data quality measure its emergence surveillance and response systems are still unproductive to deliver the right evidence to tackle the problems aptly. To that end, simple statistical analysis results were the only valuable source to make decisions. The aim of this study is, therefore, to show the applicability of data mining techniques and algorithms on the existing surveillance and response system databases using descriptive and predictive data analysis machine learning methods. To do so, the study incorporated three data mining applications, including classification, clustering and association rules mining. Consequently, an attempt was made to investigate five chosen epidemic-prone disease outbreaks using 8796 usable records and found to have significant results. So, applying data mining techniques on emerging and re-emerging disease outbreak management activities are vitally important to develop well performing descriptive as well as predictive model. By suggesting the significant of quality data to be held and application of machine learning tools and techniques on the datasets of the sector, the study provided potential contributions for the planning, preparedness, decision making, disease control and prevention measures using.

Keywords: *Data mining, Surveillance, KDD, Healthcare, Epidemic, Outbreak, Association, Cluster, Classification, Public health*

1. INTRODUCTION

In the past, when sanitary conditions were poor and diseases were little understood, epidemics occurred periodically and killed thousands of people. One of the largest epidemics ever recorded was the outbreak of bubonic plague that raged throughout Europe, Africa and Asia between 1347 to 1350G.C. and killed one-third of Europeans. An outbreak of influenza in 1918G.C. also killed over 20 million people around the world. However, during the past 70 years, there has been a dramatic fall in the incidence of infectious diseases, particularly in developed world (Mulugeta, 2004). Regarding this, there were several factors including: the provision of immunization, Anti-microbial chemotherapy, improved nutrition, better sanitation and housing. As stated by Grant and Spring (2011), the morbidity and mortality associated with infectious disease outbreaks, which are directly or indirectly linked to ecologic or climate factors and the subsequent trends pose a growing problem for global public health. In less developed countries, however, especially in the tropics, infectious diseases continue to be one of the commonest causes of death, particularly in children (Mulugeta, 2004). For example, Ethiopia as part of the developing world, has big

health problems, especially from communicable diseases outbreaks that account about 60% to 80% of the health problem in the country.

Unfortunately, such types of diseases could be prevented by simple sanitary measures and nutritional programs. However, they are still becoming the major causes of morbidity, mortality, and disability in the country (Abera and Ahmed, 2005). In addition, the communicable disease prevalence is high in the country, because of the poor socioeconomic development, environmental factors, and lack of access to safe and adequate sanitation facilities. As a result, the problems are becoming inherent for centuries and affect three-fourth of the children in the county (Krusche and Wasse, 2007). One of the most challenging tasks facing an epidemiologist is working in a public health environment to investigate disease outbreaks (US Department of Health and Human Services, 2005). Surveillance, risk assessment and outbreak response capacity are a prerequisite for effective management of emerging disease outbreaks and other acute public health events. Effective national surveillance systems can generate reliable information for timely risk assessment strategy to respond to public health problems (WHO, 2010). In 1996, as part of the response to the growing public health problem, especially for communicable diseases, Ethiopia introduced an Integrated Disease Surveillance and Response (IDSR) strategy and later called Public Health Emergency Management (PHEM) focusing on 20 selected priority diseases to the surveillance and response interventions, in which out of the them only five diseases were weekly reportable (i.e. routine surveillance).

To better identify disease clusters, track trends, assess the effectiveness of interventions in explaining and predicting emerging and reemerging disease outbreaks require quality of surveillance datasets that could provide timely results (CDC, 2010) for effective decisions. In that regard, proper collection and storage of disease outbreak data are very important to develop well-performing predictive model that can predict future occurrences of disease outbreaks (Institute of Environmental Science & Research, 2002). The present PHEM sector under study stores data about disease outbreaks using MS-excel worksheets and analyze through simple statistical tools like epi-info, and SPSS. The rationale is that, if there are adequate and relevant data sources, one could easily characterize disease outbreaks by time, place, and person (US Department of Health and Human Services, 2005). However, characterizing an outbreak in that manner would only provide descriptive results that could only show what happened in the society after the disease had already occurred. To that end, the sector was attempting to perform interventions in order to limit the problems using descriptive statistical analysis of data, including percentage, mean, median, normal distribution, and standard deviation via simple visualization techniques like tables, graphs, charts, normal curves, which are entirely traditional data analysis and presentation approaches. However, the above statistical approaches are unproductive to analyze huge datasets of several years as observed in Ethiopian PHEM sector. Therefore, the ability to extract useful knowledge hidden in the datasets of such type is becoming progressively essential in today's competitive world. To that end, applying data mining tools and techniques could provide better solutions to describe the existing situations and predict the future occurrences of outbreak cases there by suggesting a more rigorous descriptive as well as predictive models. For the current study, therefore, testing the application of data mining tools and techniques is unquestionable to generate strong decision-making alternatives in case of epidemic-prone outbreak diseases.

Finally, the primary goal of this study was to analyze the outbreak surveillance and response system of Ethiopia using data mining applications to point out the possibilities of applying data mining techniques and tools to manage the problems by developing better descriptive and predictive model to escalate the decision making efforts of the sector in early warning, planning, preparedness, and response activities. In doing so, three data mining applications such as classification, clustering and association rule mining were tested to explore their applicability using the PHEM dataset pools. So, the study would address three important questions as shown below: 1) to what extent data mining techniques and tools apply on the surveillance and response datasets of Ethiopia? 2) Could it be possible to develop more efficient predictive and descriptive models for emerging and reemerging disease outbreak cases? 3) Are there any associations among emerging and reemerging disease outbreak cases in the PHEM dataset? To address the above research questions, a research method that could help in discovering important knowledge from the surveillance datasets would be applied accordingly.

2. METHODS

The target population were all reported disease outbreak cases gathered from all over the country. This study considered all weekly reportable disease outbreak cases collected within Ethiopian public health emergency management (PHEM) sector. The study, particularly, would incorporate and analyze only five weekly reportable disease cases from the 20 epidemic-prone priority diseases identified in the surveillance database, which were collected between the years 2004 to 2012, including Malaria, Meningitis, Relapsing Fever, Typhoid Fever and Epidemic Typhus. The study used the Hybrid KDD data mining process model as shown in the Figure-1 below to analyze and evaluate the problems of the existing surveillance systems by understanding the data and discovering important patterns. It is because, the discovered knowledge and their usability could support to determine the applicability power of data mining techniques and tools to investigate outbreak surveillance datasets

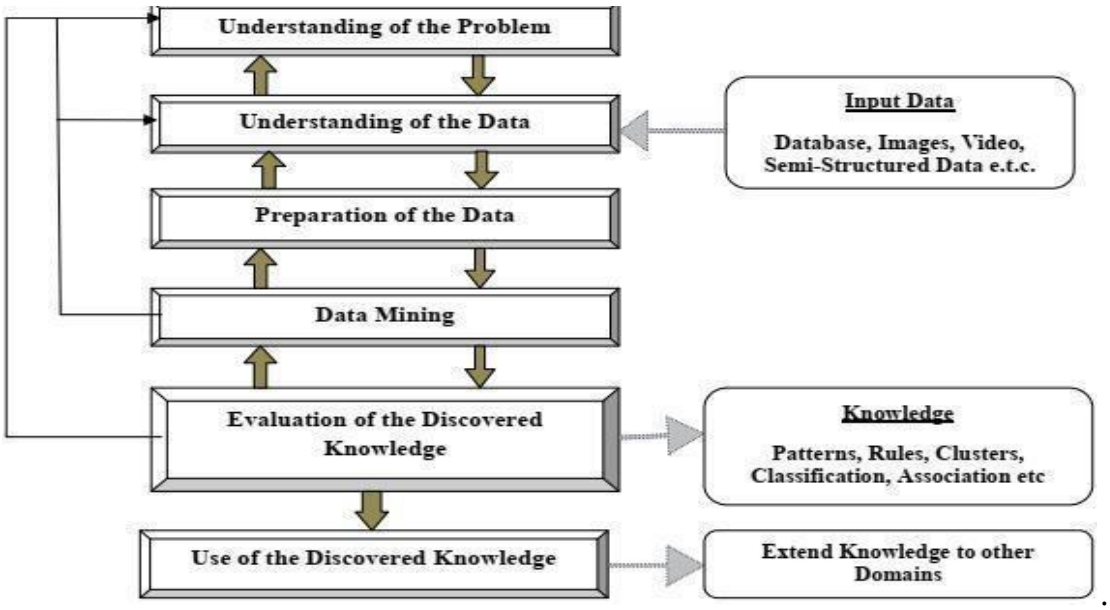


Figure 1: The six- Step of hybrid KDD process

Accordingly, Apriori association rule mining was applied for pattern discovery to see the co-occurrence of outbreak disease cases; classification algorithms were used to predict the future occurrence of disease outbreaks with regard to time and place dimensions (here, the decision tree and Naïve Bayes classifiers were applied to predict Epidemic Typhus disease); and lastly Simple KMeans clustering algorithm was examined to see how disease outbreak cases were grouped together to describe outbreak prevalence across the country within the defined time period. Here, data transformation or pre-processing were performed in the first step after the data collection process to convert the data into useable formats. Since, the surveillance system database was stored in MS-Excel file format while collecting data, we transformed them into data mining tools acceptable format like Comma Separated Value (.CSV) text format and Attribute-Relation File Format (.ARFF). Consequently, the data prepared to be analyzed by applying data mining techniques with the use of Weka 3.6 and Weka 3.7 software tools were executed to investigate the datasets and discover the hidden patterns from the surveillance datasets. So, the findings of the study were planned to conduct more rigorous investigation of the surveillance and response system using the recent datasets to develop strong predictive and descriptive models that fits with the aim of explaining emerging disease surveillance and response systems of the country by supporting the decision making process of the sector as well as assuring the important of quality data to outbreaks. Finally, the study would use the various steps of hybrid data mining process model as shown in the Figure 1 to better understand the problems and discover important patterns and rules in the process by properly preparing the preprocessing and filtering of the datasets.

DM, KD Process Model, and Hybrid KDD

KDD helps humans make sense of huge amounts of data by mining patterns and important relationships (Fayyad et al., 1996). In addition, KDD has evolved, and continues to evolve, from the intersection of research fields such as: machine learning, pattern recognition, databases, statistics, AI, knowledge acquisition for expert systems, data visualization, and high-performance computing to list some. In general, the driving force behind KDD is the database field, however, combining different models together with KDD can provide better insights on a given problem area. Therefore, to this specific study, the combined educational and industrial knowledge process model called hybrid KDD is adapted for data analysis and knowledge discovery processes. It is because of two important perspectives. In one hand, the hybrid KD Process Model stretches from the process of understanding the problem domain and data, through data preparation and analysis, evaluation, understanding, and application of the generated results. On the other hand, the hybrid KDD model has been widely used in medicine where this study was coined. The proposed model emphasizes the iteration of activities within the various phases of data mining processes via many feedback loops that are triggered by a revision process as shown in Figure 1.

3. APPLICATION OF DATA MINING IN HEALTHCARE

As stated by Soni et al., (2011), medical data mining applications have great potentials to explore hidden patterns of medical data, so that these patterns could be utilized for clinical diagnosis, pattern analysis, disease investigation, clinical decision making, disease outbreak investigation, and so on. The importance

of decision support system in the delivery of managed healthcare can hardly be overemphasized (Ranjan et al, 2007). In fact, the existing raw medical datasets are widely distributed, heterogeneous in nature, and voluminous in size. That means the healthcare industry collects huge amounts of healthcare data which, unfortunately, are not "mined" well in order to discover hidden knowledge. However, medicine is highly sensitive to information distortion and data quality matters and its effects with life threatening potentials (Shillabeer and Roddick, 2007). In addition, medical diagnosis is regarded as one of an important, yet complicated task that needs to be executed accurately and efficiently (Soni et al., 2011). So, healthcare data in any form should be collected in an organized form and then integrated together within the healthcare information systems to support the discovery of important patterns (may be using machine learning tools (e.g. data mining applications)), because the automation of such systems would be extremely advantageous for proper decision making alternatives. However, comparative analysis study of the various available tools and techniques support the strength of the model developed as a result. Here, in this study the aim was to analyze the PHEM sector disease outbreak datasets by applying data mining algorithms and techniques to drive effective predictive as well as descriptive results as recommended by Soni et al., (2011). Even if there is a wealth of data available within the healthcare systems, as stated in (Srinivas et al., 2010), the healthcare environment is generally perceived as being 'information rich' yet 'knowledge poor'. However, there is a lack of effective analysis tools and techniques to discover the hidden relationships and trends among the datasets of the surveillance and response databases. As already mentioned above, data mining applications can greatly benefit the healthcare industry without much limitations. However; according to Bedane (2010) and Koh and Tan (2005), healthcare data mining applications are limited by four major factors such as: 1) accessibility of compiled data, 2) data quality problems, 3) successful application of data mining requires knowledge of the domain area, and 4) data mining methodology and tools. Besides, lack of standard clinical vocabularies is a serious hindrance to apply data mining on health care data (Koh and Tan, 2005).

In general, Healthcare surveillance is an essential component of evidence-based decision-making practices in the public health systems. As mentioned above nowadays, investigations of diseases are more complex than they were in the past, because of emergency of new pathogens, risk factors and outbreaks, which cross jurisdictions and national boundaries, often raising political and economic burdens. The outbreaks of infectious disease and the recurrent incidences of natural disasters remind the importance of public health systems that encompass the government and private sector, academia, NGOs, associations and development partners. According to WHO (2001), effective communicable disease control relies on effective response systems and effective response systems rely on effective disease surveillance. As defined by Federal Ministry of Health (FMoH) (2009) and WHO (2012), Healthcare surveillance can be described as "the tracking and forecasting of any health event or health determinant through the continuous collection of high quality data, integration, analysis and interpretation of data results into surveillance products, thereby disseminating such products to whom they need to know [to address] as a means of "Information for Action" (Federal Ministry of Health (FMoH), 2009). In that regard, proper use of data mining techniques are not questionable due to tha fact that healthcare data mining projects can fail for a variety of reasons such as lack of management support, unrealistic user expectation, poor project management, and much

importantly erroneous data and inadequate data mining expertise and many more (Bedane, 2010). So, information technology professionals and public health professionals have to work cooperatively to enhance the decision support power of health-related cares using more sophisticated approaches. To that end, classification algorithms like a decision tree (DT) and Naïve Bayes (NB) were used to forecast future occurrences of epidemic typhus disease outbreak. Besides, descriptive data mining via association rule mining, specifically, Apriori algorithm were run to show the co-occurrence of disease outbreaks. Lastly, disease segmentation via cluster analysis method, in particular, with Simple-K-Means algorithm was performed to address the as solutions for the research questions raised. To that end, there were tremendous efforts performed to narrow down such public health gaps to maintain adequate intervention and limit the consequences of natural and human-made disasters (Federal Ministry of Health (FMoH), 2009), including infectious disease outbreaks, even if, the problems are still very high. Therefore, this specific study would implement different data mining tools and algorithms in order to evaluate the existing datasets of the sector as well as to see how predictive and descriptive model would be developed in order to enhance the effectiveness of the sector.

4. EXPERIMENTATION FOR MODEL DEVELOPMENT

Accessing and processing of the real emerging disease outbreak data for experimentation is a very difficult task because of national security and confidentiality problems. However, based on the decision made by public health experts of the sector, the sector allowed us only weekly reportable routine surveillance datasets for investigation. The data comprises of five selected epidemic prone disease cases of outbreak nature, such as Malaria, Typhoid Fever, Epidemic Typhus, Relapsing Fever, and Meningitis. So, we would develop the model of this study based on the data available on the surveillance database. A total of 9 years of 18,600 records was collected initially from PHEM and preprocessed using the hybrid KDD process model methodology. After the preprocessing phase a total of 8796 usable records were obtained to feed for data mining experimentations. Out of the total usable records, the study used 4703 records from IDSR system database and 4093 records from PHEM system database recorded from the year 2004 to 2008 and from 2009 to 2012 respectively. For prediction purpose, Tenfold (i.e. 10-fold) cross-validation was used to test the performance of a decision tree and Naive's Bayes classification algorithm. In addition, the study tested simple K-means algorithm to group the data of disease outbreaks regarding their prevalence and occurrence through the lenses of place and time setting. Finally, the study examined the applicability of Apriori association rule mining algorithm to see whether there was an association between different diseases to occur together or not at the same time and the place orientation. To see the association of disease co-occurrences, through Apriori algorithm rule interestingness and usefulness evaluation was performed with a minimum support of above 20% and a minimum confidence of above 90%. The reason of using a small support (i.e. 20%) when to develop interesting rules are to include even the rarely occurring disease cases (like meningitis) that were relatively less prevalent to occur but important to public health actions. As a result, strong rules are those rules that satisfy both minimum support threshold (min-sup) and minimum

confidence threshold (min-conf). For the sake of convenience, we put support and confidence values to appear between 0% to100%, rather than 0 to 1.0 by choice to this study.

Experimentation on association rule mining

As indicated previously the study used two different Weka software versions to generate some interesting rules using the Apriori association rule mining algorithm from the unsupervised datasets of all usable records. For experimentation, the study included all the five disease outbreak cases to show the co-occurrences and non-co-occurrences of such disease cases. Some rules that were generated from the experiment and repeated on the subsequent rules were automatically dropped out. To that, all the three datasets (i.e. IDSR, PHEM, and the combination of IDSR and PHEM) were involved in the process of investigation. As a result, comparative analysis of Weka results of the five subsequent experiments from the three datasets indicated that there were some repeated rules to be eliminated from that. Even if, the purpose of the study was to see how data mining techniques and tools are applicable in the surveillance and response systems of Ethiopia, this approach permitted us to focus and discuss only on some interesting rules that could perform well. Therefore, rules as shown below in Table 1 were the most important once obtained from the five subsequent experiments of the three datasets.

Table1: Comparison among the three datasets for Apriori association rules mining result

Exp. No.	Total Rules in the experiment	IDSR Datasets		PHEM Datasets		Combined Datasets (PHEM & IDSR)	
		Repeated Rules	Rules after removal of repeated Rules	Repeated Rules	Rules after Removal of repeated Rules	Repeated Rules	After removal of repeated Rules
E1	10	-	10	-	10	-	10
E2	15	9	6	10	5	9	6
E3	20	14	6	13	7	16	4
E4	25	12	13	9	16	10	15
E5	30	29	1	27	3	27	3
Total Rules	100	64	36	59	41	62	38

The results of the association rule mining showed that disease outbreaks were occurring or non-occurring together at some areas of the country at a point in time. So, the study projected which disease outbreak occurred with the occurrence of which types of other disease outbreaks. In addition, the experiments have also revealed that when there were non-occurrences of some disease outbreak cases some other disease outbreaks were not occurring. As a result, some interesting common rules, which were getting greater acceptance from the domain experts, were considerately chosen for the sake of rule interestingness selections and comparison purposes shown as follows in Table 2 below.

Here, the remark part in the table above showed that rules were accepted or not accepted according to their interestingness measures regarding the minimum support and minimum confidence threshold of the rules as defined in the rules' interestingness measure by the researcher. The results of the study implied that rules were accepted whenever they were interesting and not accepted whenever they were not interesting based on the given defined acceptance threshold. Even though, there were variations among the three

datasets in providing interesting rules, the average support values of over 20% to all the three datasets were accepted. In short, the average support and confidence values of the six chosen rules were greater than the required minimum defined threshold (i.e. Min-Sup=20% and Min-Conf=90%). Finally, the researchers had manually removed rules that were repeated and again appeared on the subsequent iterations to make proper analytical decisions on the remaining rules. Data mining applications, in general, association rule mining techniques, in particular, were highly significant and very applicable to analyze the disease outbreak datasets of Ethiopian surveillance and response system database.

Table 2: Selected rules and their respected level of support and confidence

Dataset Name	Rule No.	Best Rules Produced				Measure of Association		Remark
		Antecedence		Consequence				
		Occur	Not occur	Occur	Not occur	Confidence	Support	
IDSR	1.		TF, RF		ET	99%	18.71%	Not accepted
	2.	ET		TF		99%	21.01%	Accepted
	3.	Mal, ET		TF		98%	18.86%	Not accepted
	4.	Mal, RF		TF		94%	28.86%	Accepted
	5.	RF		TF		94%	32.87%	Accepted
	6.	TF		Mal		91%	72.12%	Accepted
PHEM	1.		TF, RF		ET	99%	26.34%	Accepted
	2.	ET		TF		99%	31.57%	Accepted
	3.	Mal, ET		TF		99%	31.3%	Accepted
	4.	Mal, RF		TF		98%	29.12%	Accepted
	5.	RF		TF		96%	29.83%	Accepted
	6.	TF		Mal		97%	70.29%	Accepted
(IDSR+ PHEM) Datasets	1.		TF, RF		ET	99%	22.27%	Accepted
	2.	ET		TF		99%	25.97%	Accepted
	3.	Mal, ET		TF		99%	23.31%	Accepted
	4.	Mal, RF		TF		96%	29.02%	Accepted
	5.	RF		TF		95%	31.46%	Accepted
	6.	TF		Mal		94%	71.29%	Accepted

N.B: - ET = Epidemic Typhus disease case, Mal = Malaria disease case, Men = Meningitis disease case, RF = Relapsing Fever disease case, TF = Typhoid Fever disease case.

N.B: Rule numbers were given manually for the sake of discussion purpose alone

Experimentation on Epidemic Typhus classification

Classification algorithms were also tested to classify disease outbreaks regarding their current occurrences and future incidences at a point in time within a certain area in the nation. Here, the aim of the study was only to predict the occurrence of disease cases to show the future occurrence of new or reemerging incidences at a certain district at a certain time. To that end, the study incorporated decision tree with J48 and naïve Bayes classifiers on the newly established infectious disease outbreak PHEM datasets using the 10-fold cross-validation testing approach. To that end, for the prediction purpose, attribute evaluator (i.e. feature selector) called Gain Ratio feature evaluator was used for testing by the search method of Attribute Ranker to select the best attribute from the available attributes in the dataset. So, the study incorporated ranked list of attributes according to their gain ratio of each attribute's value to predict the chosen class as shown below in table 3. After the pre-processing phase, we analyzed around 4093 usable

records using the seven chosen attributes (Region, Zone, Woreda, Year, Month, and Epidemic Weeks) to predict the class called occurrence and non-occurrence of Epidemic typhus. Finally, the occurrence of an outbreak was represented by 1 and non-occurrence by 0. Later on, we converted the value 1 to 'YES' for the occurrence and 0 to 'NO' for the nonoccurrence for convenience. Thus, the analysis results of decision tree classifiers showed better statistical values than the naïve Bayes classifiers as presented in the Table 3 below.

Table 3: Attribute Selection using Gain Ratio feature evaluator called Attribute Ranker

Attribute name	Gain Ratio value	Rank
RegionName	*0.110265*	1
ZoneName	*0.084928*	2
WoredaName	*0.078706*	3
Month	*0.000375*	4
Epidemic typhus	Class	Class

Discussion of Decision Tree J48 algorithm results (Sample Classes)

In decision tree classifier the results were analyzed in the form of IF-THEN rules. Since the research was conducted for testing the PHEM datasets to see whether the surveillance and response system database is quality and applicable to data mining tools and techniques or not, all generated rules were not explained here. The success ratio (rate) or the confidence to outbreak disease occurrence and nonoccurrence were calculated from the number after the leaf of the tree. So, Yes for occurrence and No for nonoccurrence were adapted. Accordingly, the success ratio was measured by the ratio of correctly classified disease cases with the total number of disease cases in that area.

For example, some rules were extracted and discussed as below:

1. RegionName = Addis Ababa

|WoredaName = Kolfe Keraniyo: Yes (33.0)

The above rule implies that:

The rule stated that, out of the total numbers of disease cases classified by the decision tree J48 classifier, all the records (i.e. the 33 records as shown in the rule) in the datasets were shown the disease occurrences, so it can be calculated as the success ratio = 33/33*100=100%. It can be discussed as, if the region = Addis Ababa and WoredaName = Kolfe Keraniyo, then Epidemic Typhus occur with 100% success ratio from the reported disease outbreak data. 'Yes' implies Disease occurrence and 33.0 implies the total number of occurrences classified correctly.

2. RegionName = Oromia

| ZoneName = Adama Special Zone: No (29.0/6.0)

The above rule implies that:

If region = Oromia and ZoneName = Adama Special Zone, then Epidemic Typhus do not occur with 82.86% success ratio from the given dataset.

Classification Metrics for predictive model development

After a model has been developed to predict disease outbreaks using the current PHEM datasets, performance measure was done using the training datasets so as to utilize model testing activities in the upcoming data of the surveillance system. In that regard, the future epidemic typhus disease outbreaks would be accurately predicted when and where outbreaks could occur. To that end, the predictive performance of the decision tree and Naive Bayes Classifiers using confusion matrix was empirically depicted as shown in table 4. As described by Chaudhary et al. (2008) and Bramer (2007), classification models were evaluated using the conventional machine learning metrics such as Precision, Recall, F-Measure, TP Rate, FP Rate, and ROC Area. Because, as of Han and Kamber (2006), the *confusion matrix* is a useful tool for analyzing how well your classifier can recognize tuples of different classes. In addition, the confusion matrix is more commonly named contingency table (Chaudhary et al., 2008) as shown like table 3 below. It states that the number of correctly classified instances is the sum of diagonals in the matrix; all the others are incorrectly classified.

Table 4: Measurement evaluation for the two decision tree classifiers for comparison

No.	Type of Measure	Epidemic Typhus disease Case occurrence	Naïve Bayes	Decision Tree with J48 Algorithm
1	TP Rate	No	0.798	0.915
		Yes	0.923	0.795
		Average	0.838	0.877
2	ROC Area	No	0.939	0.935
		Yes	0.939	0.935
		Average	0.939	0.935
3	Sensitivity		0.6814	0.8141
4	Specificity		0.9570	0.9049

As shown in the table above, one could see that the occurrence of disease cases were measured by four important prediction performance tests. Since the study was planned to see the best performing classification algorithms to predict the occurrence of Epidemic typhus outbreaks at a certain place and a future point in time, performance was evaluated by the average true positive (TP) rate to evaluate Decision Tree with J48 classifier and the Naïve Bayes classifier. From the experiment, the average TP rates were 83.8% and 87.7% on Naïve Bayes and Decision Tree J48 classifiers respectively. In principle, the better the average TP rate the better the performance will be. The second one was the ROC area which is the combination of the sensitivity and specificity measure. The average values of ROC area were 93.9% for Naïve Bayes and 93.5% for Decision Tree classifiers, which was greater than 90% and nearly equal, showed that both algorithms generated better performance. Since specificity and sensitivity are the most important performance measures in the field of healthcare related studies. As a result, the two measures were compared and found that the sensitivity of the Naïve Bayes algorithm was 68.14% and that of the Decision tree j48classifier was 81.41%. Based on the measures of sensitivity, the true occurrences of disease cases were to be classified as an occurrence. From the above test results, one could understand that decision tree had a better performance impact than Naïve Bayes classifier to predict the future occurrences of Epidemic Typhus outbreaks. In contrast, specificity measure showed that the non-occurrences of Epidemic Typhus

outbreaks were classified as non-occurrences with the test results of 95.7% and 90.49% for Naïve Bayes and decision tree classifiers respectively. Even though, both classifiers had shown high specificity measures with a relative higher performance measure of Naïve Bayes classifier, as per the objective of the study, outbreak disease cases were evaluated not to test the non-occurrence as can be measured by specificity, but the occurrences as can be measured by sensitivity. So, important classification measures like sensitivity, TP rate and the ROC area measures were taken into consideration to make decisions on what classification algorithm better performance was attained. Moreover, Decision tree classifier (using an IF-THEN rule representation) had classified data instances correctly with a 87.66% accuracy rate, but that of the Naïve Bayes classifier was with 83.78%. Even if, the aim of the study was to show and understand how data mining techniques and tools are applicable in case of disease outbreak surveillance and response databases, based on the discussions and justifications above decision tree with j48 was selected for model development as having better performance than Naïve Bayes classifier.

Experimentation to analyze outbreak clusters

The study used the combined datasets of 8796 usable records for clustering techniques. The results were also interesting to design new hypotheses and conduct empirical studies in the field of machine learning applications, including data mining techniques within the PHEM datasets. Because, the above implication could provide a measurable plan of action for decision makers to better prepare for and respond to outbreak incidences. In fact, we had two options to find the number of clusters (i.e. K) from the whole dataset: 1) using the total numbers of years in which data collection process in the sector was performed (i.e. 9 years of datasets); and 2) the total number of regions in the country in which data were reported (i.e. 11 total regions indicated in the datasets). To that end, we discussed with domain experts to productively decide the number of Ks and we agreed to cluster the datasets based on the number of regions in the country (i.e. 11 clusters). Cluster analysis showed that the year 2011 was indicated as having more outbreak disease occurrences than the other sampled years and appeared within 4 different places across the country. As per the objective of the research, better results and interesting clusters were observed from Simple K-means clustering algorithm experiment results as shown in Table 5 below. So, data mining has brought greater applicability to correctly describe the outbreak surveillance and response system datasets of the country. Except the years in 2004 and 2012 when we couldn't find any cluster, all the other years in the experiment appeared exactly one each. So, the surveillance and response system experts should have to cross check the prevalence of outbreak disease cases against their plan of actions on that specified year and places of occurrences.

5. LIMITATION OF THE STUDY

Even though, there were promising results from the study that meet its objective, there were some important limitations as other studies have. The first limitation is that the datasets were poor quality and having very shallow dimensionalities. So, using such datasets to provide significant implications may not be fully applicable. It is because, data quality and adequate volume and dimensionality are the upmost important things to use machine learning technologies, including data mining tools and techniques. In

addition, the datasets were collected many years back from 2004 to 2012 with a lot of missing values within it, so, the current world reality dynamics might be different from the time when the data were collected. So, providing recommendation based on such datasets might not work well today. The surveillance system was lacking consistent and proper data reporting format while gathering data from all over the districts in the the country. For example, outbreak data reporting formats of different regions were very different. In addition to that, there are diverse data mining algorithms and techniques with different predictive and descriptive powers and applicability, but this specific study was only relied on some common algorithms, including decision tree with j48, Naïve Bayes, Apriory, and Simple K-means from the set of several techniques. Therefore, the potential power of data mining applications might not be fully utilized.

Table 5: Outbreak clustering with simple K-Means algorithm results

Attribute	Full data	Cluster #										
		0	1	2	3	4	5	6	7	8	9	10
RegionName	SNNPR	SNNPR	Tigray	Oromia	Addis Ababa	Oromia	Addis Ababa	Tigray	Oromia	Harari	Oromia	Addis Ababa
ZoneName	Central Tigray	Segen	South Tigray	Arsi	Arada	East Hararge	Kolfe Keranio	Central Tigray	West Shoa	Harari	West Hararge	Yekatit Hospital
Year	2011	2011	2011	2010	2011	2005	2009	2010	2006	2011	2007	2008
Month	Mar	Jan	May	Sep	Jun	May	Jan	Jun	Apr	Jul	Oct	Dec
MalariaOutbreak	92%	98%	98%	100%	93%	99%	43%	99%	99%	63%	97%	70%
TyphoidFever Outbreak	76%	98%	73%	100%	99%	81%	97%	26%	96%	15%	83%	61%
RelapsingFeverOutbreak	33%	30%	4%	87%	72%	31%	76%	3%	66%	63%	11%	13%
EpidemicTyphusOutbreak	26%	25%	2%	87%	95%	3%	86%	2%	11%	0%	5%	12%
MeningitOutbreak	19%	20%	4%	58%	15%	43%	9%	5%	32%	10%	5%	7%
Total number of cluster instances involved	**8796**	**1257**	**887**	**798**	**773**	**683**	**344**	**1135**	**767**	**480**	**1110**	**562**
Percentage of instances	**100%**	**14%**	**10%**	**9%**	**9%**	**8%**	**4%**	**13%**	**9%**	**5%**	**13%**	**6%**

6. CONCLUSION

The PHEM surveillance and response system were suffering from a serious shortage of quality data and a consistent data reporting mechanisms across the various regions of the country. In addition, number of attributes (e.g. lack of unseen variables) in the PHEM datasets were not adequately investigated for applying data mining. So, the absence of quality data in the sector might adversely affect the quality of both the predictive and the descriptive power of the generated models. Disease prediction or classification has shown that some places of the country were more vulnerable than others for the incidence of Epidemic typhus disease cases. Decision tree J48 algorithm was better than Naïve Bayes classifier for the data at hand to prepare prediction model in relation to time and place settings. It was found that diseases were associated to occur together and not occur together. Therefore, association rule mining with Apriori algorithm was an important means to show the real association of disease cases (but here the associations didn't show causalities). Based on the datasets, the more affected and the less affected areas of the country were clearly

identified via clustering techniques for the 9 successive years. In general, data mining techniques (i.e. classification, clustering and association rule mining) were important and highly applicable to analyze the emerging and reemerging disease outbreak cases. Having that in mind, outbreak disease planning, preparedness, and response would be much easier if data mining applications are becoming a means of data analysis parts in the face of quality and voluminous datasets as expected from such sectors. Apriori algorithm couldn't detect the infrequently occurring rear disease cases like meningitis even with the defined minimum support value of 20%. The simple K-Means clustering algorithm showed that some disease cases such as: Malaria and typhoid fever were more frequently occurring than others throughout all regions of the country in all datasets. From the two datasets incorporated, one can conclude that most prevalent disease cases such as: malaria and typhoid fever were also highly prevalent in the former IDSR datasets. It indicated that the two outbreaks were not effectively controlled by the newly established PHEM surveillance and response system. However, in general, the current PHEM surveillance system of the country was better than the former IDSR system in providing interesting patterns with data mining techniques and tools

REFERENCES

Abera K. and Ahmed A. (2005). "An overview of environmental health status in Ethiopia with particular emphasis to its organization, drinking water and sanitation, A literature survey," Department of Community Health, Medical Faculty, Addis Ababa University, Ethiopia.

Bramer M. (2010). "Principles of Data Mining," University of Portsmouth, UK; Springer.

Chaudhary K., Papapanagiotou I. and Devetsikiotis M. (2010). "Flow Classification Using Clustering And Association Rule Mining," 2010 15th IEEE International Workshop on Computer Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD) (2010). pp. 76-80.

Communicable Disease Prevention and Control (CDC) (2010). "Communicable Disease Prevention and Control (Focus Area Profile) plan," Wisconsin Department of Health.

Fayyad U., Piatetsky-Shapiro G, and Smyth P. (1996). "From Data Mining to Knowledge Discovery in Databases," American Association for Artificial Intelligence, (AI Magazine Vol. 17, No. 3.

Federal Ministry of Health (FMoH) (2009). "Public Health Emergency Management Guideline," Ethiopian Public Health Institute, Addis Ababa, Ethiopia.

Grant R., and Spring S. (2011). "A multidisciplinary approach for the early detection and response to disease outbreaks," USA, Armed Forces Health Surveillance Center.

Han J. and Kamber M. (2006). "Data Mining: Concepts and Techniques," 2nd Edition, The University of Illinois at Urbana-Champaign, Morgan Kaufmann.

Institute of Environmental Science & Research (ESR) (2002). "Disease Outbreak Manual," Porirua, New Zealand.

Institute of Environmental Science & Research (2002). "Disease Outbreak Manual," Porirua, New Zealand.

Kamber M., and Han J. (2006). "Data Mining: Concepts and Techniques", 2nd Ed., University of Illinois at Urbana-Champaign.

Koh H. C. and Tan G. (2005). "Data Mining Applications in Healthcare," Journal of Healthcare Information Management, Vol. 19, No. 2.

Krusche and OtterWasse (2007). "Introduction of Ecological Sanitation for Large Scale Housing Programs in Ethiopia," GTZ.

Mulugeta A. (2004). "Communicable Disease Control, Ethiopian public health training initiatives," Hawassa University, Ethiopia.

Ranjan J., Nagar R. and Ghaziabad (2007). "Applications of Data Mining Techniques In Pharmaceutical Industry," Journal of Theoretical and Applied Information Technology, Uttar Pradesh, India.

Shillabeer A., and Roddick J. F. (2007). "Establishing a Lineage for Medical Knowledge Discovery," Carnegie Mellon University, Australia.

Soni J., Ansari U., and Sharma D. (2011). "Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction," International Journal of Computer Applications, Vol. 17, No.8, pp.0975 – 8887.

Srinivas K., Kavihta B. R. and Govrdhan A. (2010). "Applications of Data Mining Techniques in Healthcare and Prediction of Heart Attacks," International Journal on Computer Science and Engineering, Vol. 2, No. 2, pp. 250-255).

US Department Of Health and Human Services (2005). "Principles of Epidemiology: An Introduction to Applied Epidemiology and Biostatistics," 2nd Ed, Centers for Disease Control and Prevention (CDC), Atlanta, Georgia 30333.

WHO (2010). "Asian pacific strategy for Emerging Diseases," Library Cataloguing in Publication Data.

Who (2012). "Outbreak Surveillance And Response In Humanitarian Emergencies," Geneva, Switzerland.

On-Demand Service in Mass Transit: Addis Ababa Smart City Initiative

Eyobed Tilaye, Eyuel Tibebu, Ezedin Ali, Milkessa Oljira, Sifan Dereje, Ramasamy S.

Core members of Addis Ababa Science and Technology University's AI and Robotics Center of Excellence
E-mail: rams@aastu.edu.et

ABSTRACT

Digitization, increasing automation and new business models like shared mobility has revolutionized transportation and mobility in developed countries. There are ridesharing companies like Uber and Lyft provide technological platforms and support to connect drivers and riders on the basis of demand-response services in western world. However, in Ethiopia ridesharing services were at nascent stage. Although the most improvements in on-demand applications have been experimented in private transit services, there is no any implementation in public transportation to connect public transit services and passengers each other especially during peak commute hours. The purpose of this paper is to introduce a tool that attempts to remove unnecessary waiting time in bus depot that impedes the productivity of the workforce and work ethics. This paper presents the concept of mobility on-demand service and its application in public transit services with a technological innovation using YOLO and XG Boost algorithm. This tool is made up of two parts: Detection and Prediction. Detection includes the method and algorithm that we have implemented to observe and extract raw data: that is, an object detection algorithm that detects the number of people in a given place, YOLO is used for density estimation. The Prediction algorithm then utilizes the output of the Detection to finally predict the number of people at a specific time in a given area. To do this, we have implemented the XGboost algorithm with GirdSearchCV on the dataset. A similar predictive algorithm was made by other scholars from India [1], and had an accuracy rating of 80.05%. Our algorithm has an accuracy of 81.12%. It is important to note that our predictive algorithm was tested on the Telcom Dataset and Kaggle dataset, since pending permission on accessing real-time camera feed information. Obtaining the results on above datasets, we can transfer its attributes to our own custom data as needed.

1. INTRODUCTION

A significant amount of time and energy is wasted by the working class due to the inefficiency of the public transportation system. Employers and employees alike stand in long queues, and in even more common scenarios, hustling - rushing and pushing to get to a particular destination is observed. People suffer such nuances to get to work, and to get home from work. Most of the public in Ethiopia utilizes public transportation as a means to get by. Noting this, it then is accurate to infer that the transportation sector contributes a significant amount to the annual expense of the country. In an analysis of the Ethiopian Budget in fiscal year 2019/2020, it was estimated that 2.3 Billion ETB was spent on Transport and Communication [2]. Hence, it is knowledgeable to imply that optimization in the service of the transportation sector is of paramount importance.

This paper employs the use of Artificial Intelligence to provide real-time data and the use technology attempts to utilize the object tracking feature of AI to recognize the number of people in a particular place and alert the authorities if an agreed threshold is to be surpassed.

It will contribute a great deal to the beneficiaries of the Addis Ababa city public transport. This paper also introduces a data export feature of real-time events. This data can be utilized; this paper proposes two ways to effectively utilize the generated data:

1. Prediction

Given that we will have collected a satisfying dataset to train a model, and after we have gone through the rigorous data preprocessing measures, we could use a prediction algorithm to predict the number of people at a given date at a given location at a given time. This could be used by the authorities so as to prepare for predicted incoming traffic aforementioned date. This could offer a huge support as the concerned authorities have the ability to prepare - in the number of public transportation vehicles to put to actions, new road plans, and budgeting - ahead of time, and make an educated and likely estimation based on this technology.

2. Selling Data

Advertising companies' success depends on how many people see the creative work the company puts in. In this instance, the technology we propose generates data that contains the location, the number of people in a particular area at a particular date at a particular time of day. If the exported data is sold to advertising companies, along with the prediction for a desired date - it could be utilized to identify the estimated population density of a certain location so that the advertising companies can display their work only on desired time schedules.

3. FRAMEWORK SELECTION

Pytorch vs. TensorFlow

Two popular frameworks exist when we consider the topic of object detection and tracking. One is the Pytorch Framework developed by Facebook [3]. The aim of this development was to bring the usability and speed of a deep learning algorithm into one package.

On the other hand we have the Google-made Tensorflow framework [4]. Tensorflow is a machine learning system that works on heterogeneous environments; it implements multi-core CPUs, General Purpose GPUs, and the even more impressive custom-designed ASICs known as Tensor Processing Units (TPUs).Pertaining to this project, which of these frameworks would be more suitable in the efficient implementation of the project? To answer this, we refer to a detailed study conducted on the subject of Pytorch Vs Tensorflow conducted by Kirill Dubovikov, the CTO of Cinimex DataLab [5]. In this analysis, multiple differences between the two frameworks are provided. Moreover, the author attempts to show the difference in accuracy and loss between the frameworks based on different loss metrics. In yet another article in which an analysis performed by Visio.ai, a comprehensive guide in the difference between the two frameworks can be observed [6]. This article compares the two in regards to Performance, Accuracy, Training Time and Memory Usage, and Ease of Use. This article summarizes that both are functionally equivalent although Pytorch slightly outperforms Performance and Ease of Use: the latter is due to the Object Oriented structure of the Pytorch algorithm.

Thus, we have decided to use the Pytorch implementation of object detection and tracking. Yet another problem poses in this decision - Which Pytorch-based algorithm do we use? In the sections below, the selected algorithm as well as the rationale behind this choice is provided.

YOLO- Key Enabler:

Object detection algorithms come in many forms and sizes. Some use the Convolutional Neural Networks (CNN) others use the Recursive Neural Networks (RNN) and some use even the more advanced fast RNN and faster RNN. Of these Object detection families and more, two are prevalent in today's application: RCNN and YOLO.

Yolo - You only Look Once - is an object detection and object classification algorithm developed in 2015. It has since then been widely used and developed. In its initial paper [7], the authors stated that in YOLO "*A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation.*" Hence its name, YOLO.

The authors continue to compare the performance of YOLO with other real time object detection methods in terms of accuracy and speed - the two paramount factors in Computer Vision and Object detection.

In doing so, this paper asserts that YOLO is superior in speed that any other real-time detection algorithm. The proposed alow chart is shown in Fig 1, the speed of the algorithm is fast. '*Our unified architecture is extremely fast. Our base YOLO model processes images in real-time at 45 frames per second.'*

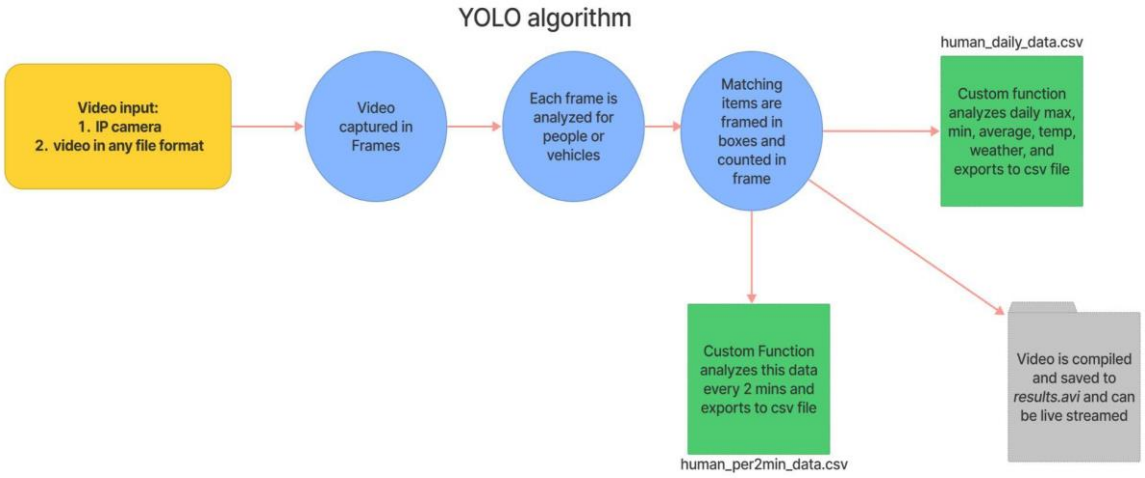


Figure 1: Flowchart for our algorithm.

Why YOLO?

A paper was published comparing the various existing real-time objects [8]. In this paper, the author tabulates the different accuracies as well as speeds pertaining to different object detection algorithms as follows:

Table 1: Analysis of Different Object detection algorithms

Method	mAP	FPS	batch size	#Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	6000	1000X600
Fast YOLO	52.7	155	1	98	448X448
YOLO (VGG16)	66.4	21	1	98	448X448
SSD300	74.3	46	1	8732	300X300
SSD512	76.8	19	1	24564	512X512
SSD300	74.3	59	8	8732	300X300
SSD512	**76.8**	**22**	**8**	**24564**	**512X512**

Here, we can observe that the speed of YOLO is superior to its peers by a great margin. It is also noticed that the accuracy criterion **mAP (mean Average Precision)** is varying in favor of Faster R-CNN.

To this, the authors of YOLO have embarked to better the algorithm in terms of both speed and accuracy. In this same publication, a comparison between YOLOv2 and its peers has been provided as follows:

Table 2: Analysis between the Yolo Family and R-CNN in terms of speed and accuracy

Detection Frameworks	Train	mAP	FPS
Fast R-CNN	2007+2012	70.0	0.5
Faster R-CNN VGG-16	2007+2012	73.2	7
Faster R-CNN ResNet	2007+2012	76.4	5
YOLO	2007+2012	63.4	45
SSD300	2007+2012	74.3	46
SSD500	2007+2012	76.8	19
YOLO v2 288 x 288	2007+2012	69.0	91
YOLO v2 352 x 352	2007+2012	73.7	81
YOLO v2 416 x 416	2007+2012	76.8	67
YOLO v2 480 x 480	2007+2012	77.8	59
YOLO v2 544 x 544	2007+2012	78.6	40

It can be well said that the trade-off between speed and accuracy in the improved YOLO model is productive. At its slowest - at 40 frames per second, YOLOv2 can provide a mean accuracy of 78.6 with an image resolution of 544 X 544 while the Faster R-CNN ResNet model provides an accuracy of 76.4 with an inferior speed of only 5 frames per second. Moreover, YOLO continues its upgrade into YOLOv3, then onto YOLOv4 and even YOLOv5, meaning only a steady, if not an exponential, growth in speed and accuracy.As described above, our project relies on real-time, fast and accurate prediction of objects given within a frame. By means of the above comparison, it is clear that YOLO will suit our project best and give us the best possible results as compared to existing models.

Predictive algorithm

Many Predictive algorithms exist in the Machine learning and Deep Learning domain that range from the simplistic logistic regression to the more complex boosting algorithms.Boosting is an ensemble learning method that combines a set of weak learners into a strong learner to minimize training errors. Multiple forms of boosting exist; the main categories of which are AdaBoosting - Adaptive boosting, Gradient

boosting and Extreme Gradient boosting - XGBoost. Our project utilizes XGboost as its primary prediction tool. In a study [9], that compares and contrasts the main machine learning algorithms: XGboost, Gradient Boost, and Random Forest, it can be seen that XGboost is superior it's competitors. This study was tested on a wide range of datasets - in order to ensure the algorithms' robustness. In addition to this, this study shows that there is a significant increase in accuracy when hyperparameters are tuned. In our algorithm, as in the aforementioned study, we use GridSearchCV as a means to find the optimal hyperparameters.

GridSearchCV is a brute-force algorithm; it goes through all possible combinations to find the optimal hyperparameters.

It is for this reason that we have picked XGboost with GridSearchCV hyperparameter tuning as the predictive tool for our algorithm. Although GridSearchCV is robust, it takes a lot of computational resources to operate. So, as an alternative to GridSearchCV, we have implemented the RandomizedSearch algorithm. On the same dataset - Telcom Dataset - the GridSearchCV and RandomizedSearch algorithms results are as follows:

Algorithm	**GridSearchCV**
Input Hyperparameters	'gamma' : [0.2,0.8,1,1.5,2] 'learning_rate': [0.1,0.15,.195,0.2,.215] 'max_depth': [4,5,8,10,12] 'n_estimators': [50,75,100,150] 'reg_lambda': [30,33,35,40]
Optimal Hyperparameters	'colsample_bytree': 0.6 'gamma': 1 'learning_rate': 0.215 'max_depth': 8 'n_estimators': 100 'reg_lambda': 80 'subsample': 0.7
Analysis	1. Test data set score is: 81.12136266855926% 2. Train data set score is: 81.5406460773873%

Table 3: Experimental results for GridSearch CV

Algorithm	**RandomizedSearch**
Input Hyperparameters	'gamma' : [0.2,0.8,1,1.5,2] 'learning_rate': [0.1,0.15,.195,0.2,.215] 'max_depth': [4,5,8,10,12] 'n_estimators': [50,75,100,150] 'reg_lambda': [30,33,35,40]
Optimal Hyperparameters	'subsample': 0.7 'reg_lambda': 90 'n_estimators': 150 'max_depth': 8 'learning_rate': 0.215 'gamma': 1 'colsample_bytree': 0.6
Analysis	1. Test data set score is: 81.618168914123 49% 2. Train data set score is: 81.36315228966986%

Table 4: Experimental results for Randomized Search

Here, we can see there isn't much difference as pertaining to the optimal parameters. But if we look at the analysis, we can see that the RandomizedSearch has under-fitted our data because the training data set accuracy is less than that of the testing data set. Pursuant to this, we have implemented XGboost with GridSearchCV. The flowchart of the same in shown in Figure 2.

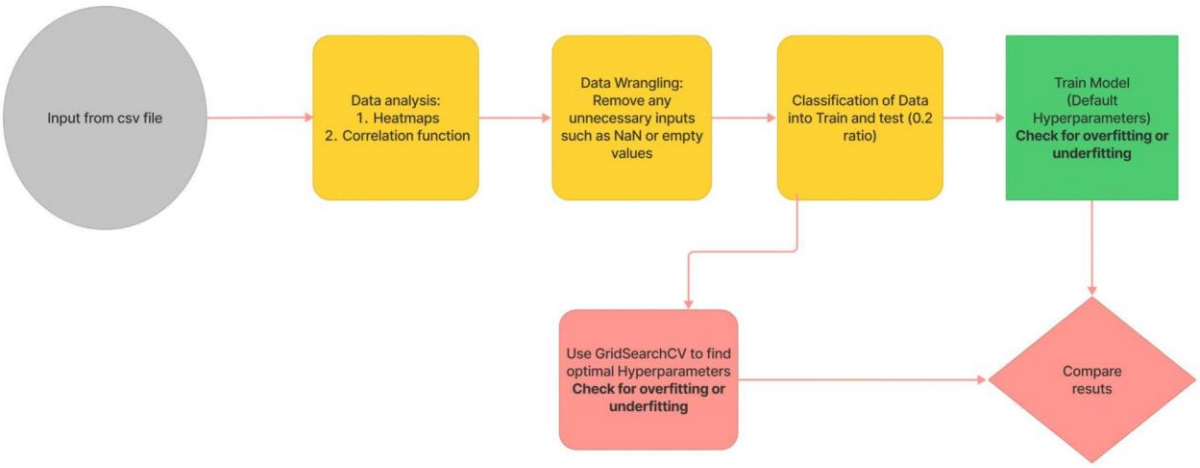


Figure 2: Flowchart for XGboost algorithm with and without GridSearchCV

4. RESULTS AND DISCUSSIONS

Data Generation:

We've discussed that our Yolo Object Detection Algorithm (Yolov4) counts the number of objects. For the purposes of this project, we've modified the classes of objects to be captured and analyzed in two - person and vehicle. We've implemented this as it is parallel to our goal and we needn't waste additional computational power by detecting other objects - although, it is possible for us to add objects for future works such as license plates.

It would be rather short-handed for our project to just count the number of objects in a given scenario. Being able to count the number of objects only goes so far as far as the utility of the algorithm is concerned. We've designed our project so that the collected data is stored in a meaningful manner that could be used in other endeavors in the future.

The grandiosity of this can be more emphasized when viewed in context of the lack of data mentioned in the introduction of this paper. Africa is lacking in data - our project aims at tackling this project from one angle. The data exported is stored in a csv file so that it can be organized and usable in other areas - one of which we have taken the liberty of putting to practice. The data exported is based on the classes predicted: meaning that a separate csv file is exported for each class detected. For instance, if we are detecting the number of people in a given scenario, then a file *human_per2min_data.csv* and *human_daily_data.csv* will be separately exported.

The file *human_per2min_data.csv* records and exports the data of people recorded by a camera. We could export the data based on each frame captured by our algorithm, but this makes for an unnecessarily large and useless dataset as it would be inconsistent and inaccurate. To elaborate, a person walking across the visual range of the camera could be captured in one of the frames and left out in the next. This would result in poor quality of the algorithm and reliability of the dataset. Hence we have opted to average the amount of people captured in the number of frames present in a span of two minutes. We've found that this span of time would make for a reliable dataset: for a 16 hour active time of the camera, we would get

(16*60)/2 number entries of data per day. When necessary, we can alter this number as seen fit in future endeavors.

This dataset also contains the location and weather information of the place where the camera is located. The figure below shows a sample data of the described file.

	A	B	C	D	E	F	G
1	Date	Day of Week	Location	Weather	Average People per 2 mins		
2	10/8/21 5:08 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	3		
3	10/8/21 5:08 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	3		
4	10/8/21 5:08 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	3		
5	10/8/21 5:08 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	3		
6	10/8/21 5:08 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	3		
7	10/8/21 5:08 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	3		
8	10/8/21 5:08 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	3		
9	10/8/21 5:09 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	2		
10	10/8/21 5:09 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	3		
11	10/8/21 5:09 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	2		
12	10/8/21 5:09 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	2		
13	10/8/21 5:09 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	1		
14	10/8/21 5:09 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	1		
15	10/8/21 5:09 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	1		
16	10/8/21 5:09 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	2		
17	10/8/21 5:09 PM	Fri	['9.025', '38.7469']	['Rain', '16.75']	1		

Figure 3: Snippet of human_per2min_data.csv

The file *human_daily_data.csv* consists of the summarized daily account of the algorithm. When 24 hours have elapsed - at 23:59:59 - the daily maximum number of people, the time at which this daily maximum occurred, the daily minimum number of people, the time at which this daily minimum occurred, the location, the weather will all be exported to said file.

It is important to note that having the *per2min* and *daily* statistics offers significant information in unison. The *per2min* could be used to alert concerned bodies to act when an agreed number of people stand in a crowd for more than an agreed amount of time; the *daily* data could be used as a guide and means of prediction that will be discussed in the topics below.

It is also important to note that the export nature is not only limited to people, but to every class used. The vehicle class will also have these attributes: vehicle*_per2min_data.csv vehicle_daily_data.csv.*

	Date	Day Of We	Location	Weather	Daily_Average	Daily_Max	Daily_Max_time	Daily_min	Daily_min_time
2	10/30/2021 21:34	Sat	['9.025', '38.7469']	['Clouds', '10.46']	nan				
3	10/30/2021 21:38	Sat	['9.025', '38.7469']	['Clouds', '10.46']	nan				
4	10/30/2021 21:46	Sat	['9.025', '38.7469']	['Clouds', '10.46']	nan				
5	10/30/2021 21:46	Sat	['9.025', '38.7469']	['Clouds', '10.46']	1	1	21:46:16	1	21:46:16
6	10/30/2021 21:46	Sat	['9.025', '38.7469']	['Clouds', '10.46']	2	3	21:46:16	1	21:46:26
7	10/30/2021	Sat	['9.025', '38.7469']	['Clouds', '10.46']	2	3	21:46:16	1	21:46:26
8	10/30/2021 21:46	Sat	['9.025', '38.7469']	['Clouds', '10.46']	2	3	21:46:16	1	21:46:26
9	10/30/2021 21:46	Sat	['9.025', '38.7469']	['Clouds', '10.46']	3	5	21:46:16	1	21:46:42
10	10/30/2021 21:46	Sat	['9.025', '38.7469']	['Clouds', '10.46']	3	5	21:46:16	1	21:46:42
11	10/30/2021 21:46	Sat	['9.025', '38.7469']	['Clouds', '10.46']	3	5	21:46:16	1	21:46:42
12	10/30/2021 21:46	Sat	['9.025', '38.7469']	['Clouds', '10.46']	3	5	21:46:16	1	21:46:42
13	10/30/2021 21:47	Sat	['9.025', '38.7469']	['Clouds', '10.46']	3	5	21:46:16	1	21:46:42
14	10/30/2021 21:47	Sat	['9.025', '38.7469']	['Clouds', '10.46']	3.75	6	21:46:16	1	21:47:09
15	10/30/2021 21:47	Sat	['9.025', '38.7469']	['Clouds', '10.46']	4.2	6	21:46:16	1	21:47:09
16	10/30/2021 21:47	Sat	['9.025', '38.7469']	['Clouds', '10.46']	4.2	6	21:46:16	1	21:47:09
17	10/30/2021 21:47	Sat	['9.025', '38.7469']	['Clouds', '10.46']	4.5	6	21:46:16	1	21:47:09
18	10/30/2021 21:47	Sat	['9.025', '38.7469']	['Clouds', '10.46']	4.5	6	21:46:16	1	21:47:09
19	10/30/2021 21:47	Sat	['9.025', '38.7469']	['Clouds', '10.46']	4.5	6	21:46:16	1	21:47:09
20	10/30/2021 21:47	Sat	['9.025', '38.7469']	['Clouds', '10.46']	4.5	6	21:46:16	1	21:47:09
21	10/30/2021 21:47	Sat	['9.025', '38.7469']	['Clouds', '10.46']	4.857142857	7	21:46:16	1	21:47:56
22	10/30/2021 21:48	Sat	['9.025', '38.7469']	['Clouds', '10.46']	4.857142857	7	21:46:16	1	21:47:56

Figure 4: Snippet of human_daily_data.csv

We've tested our algorithm on the Kaggle Veri Seti Dataset [10]. Our Algorithm works extremely well on this dataset. A sample output can be shown below for each of the detection items used in the algorithm:

Figure 5: Detection and Annotation of human class from Veri Seti Dataset

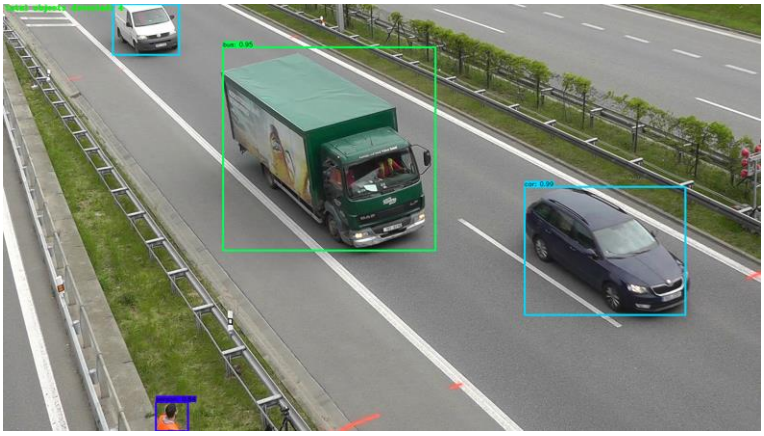


Figure 6: Detection and Annotation of the vehicle class from the Veri Seti dataset

We wanted to further test the robustness of the algorithm. So we implemented it on a more challenging picture as shown below:

Figure 7: Overhead shot of pedestrians [Not in local context]

We've found that there are 46 identifiable people in this picture. Our algorithm was ran on the following picture and the results are shown below: 42 people and 1 handbag have been detected

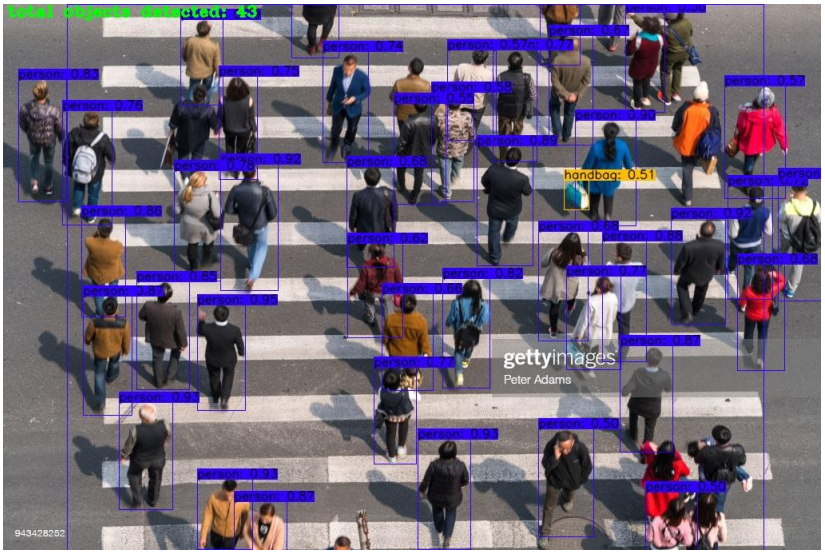


Figure 8: Result of output of our algorithm on figure 7

The prediction Algorithm uses export data as input. Since we weren't able to generate data by ourselves – we couldn't use our own camera to put up on the streets and record - we've implemented our prediction model on the Telcom Dataset [11]. We've done this because it is easy to transfer the methods and steps used here once we get the data we need.

Even though finding a well-organized local context dataset is difficult, we were able to procure a video that shows people standing in queues whilst waiting for public transportation. Here, the location is at Megenagna, Addis Ababa. The image below shows our algorithm tracking people in this footage.

Figure 9: Result of our algorithm on Localized snapshot around Megenagna

In this snapshot, we can find 17 visible people. Of the present 17, 15 people were captured and recognized by our algorithm. Provided that this video [12] wasn't taken at the right angle and the right resolution, we can assume that an 88.2% accuracy is sustainable enough accuracy for its intended purpose.

5. CONCLUSION & FUTURE WORKS

The YOLO and XG-Boost algorithm with GridSearch are used for object detection and prediction. This algorithm is dynamic; it's not limited to a single objective. Currently, we are working on modifying the algorithm to include speed detection and fuel wastage estimation for cars in traffic jams. With Speed detection, a camera will detect the speed of a moving car and determine - to a reasonable degree - if the car is above the speeding limit. This would be very useful on roads where the speed limit is low (below 30 km/h) or on high speed roads (50-60 km/h). In case of violation of any of these speeding limits, an alert can be sent to authorities, and the license plate can be recognized and be sent to the authorities for action. In regards to fuel wastage, our algorithm can count the number of vehicles in a given frame. If we can estimate the given fuel wastage by a single vehicle while waiting in traffic per day, and then give detailed statistical information to the government, then valuable information will be input to the decision to be made regarding pollution policies, road expansion, and the likes.

On the user side, we are now developing a mobile application that enables people to see where there are traffic jams and where there are large queues for public transportation. Hence, a user, before leaving his/her house, can know the congestion and traffic jam status of his/her destination.

REFERENCES

1) Telecom Churn Prediction Model using XGboost Classifier and Logistic Regression Algorithm:https://www.academia.edu/53283510/IRJET_Telecom_Churn_Prediction_Model_using_XgBoost_Classifier_and_Logistic_Regression_Algorithm
2) Analysis of the 2019/20 Federal Budget Proclamation by UNICEF www.unicef.org/ethiopia/reports/analysis-201920-federal-budget-proclamations
3) PyTorch: An Imperative Style, High-Performance Deep Learning Library: arXiv:1912.01703
4) TensorFlow: A System for Large-Scale Machine Learning: ISBN 978-1-931971-33-1
5) PyTorch vs TensorFlow — spotting the difference: https://github.com/kdubovikov/tf-vs-pytorch.git
6) Pytorch vs Tensorflow: A Head-to-Head Comparison
7) You Only Look Once: Unified Real-Time Object Detection - arXiv:1506.02640
8) A review: Comparison of performance metrics of pretrained models forobject detection using the TensorFlow framework - S A Sanchez et al 2020 IOP Conf. Ser.: Mater. Sci. Eng. 844 012024
9) A Comparative Analysis of XGboost: https://arxiv.org/abs/1911.01914
10) https://www.kaggle.com/enesbayturk/vehicle-and-pedestrian-detection-dataset
11) https://www.kaggle.com/blastchar/telco-customer-churn
12) https://www.youtube.com/watch?v=lH8TB5IVVPc

Modeling & Designing of a Multilevel SVPWM & Fuzzy (AI) Based Dynamic Voltage Restorer with Sag & Swell Limiting Function

G. Madhusudhana Rao[1], Y. Prasanna Kumar[1], P. Janaki Ram[2], T. Gopi Krishna[3]

[1]*Department of Electrical and Computer Engineering, Scool of Engineering and Technology, Bule Hora University, Ethiopia*
[2]*Department of Mechanical Engineering, Adama Science & Technology University. Adama, Ethiopia*
[3]*Department of Computer Science and Engineering, SOEEC, Adama, Ethiopia*
E-mail: gmrgurrala@gmail.com, prasannaky@gmail.com, srirama309@gmail.com, gktiruveedula@gmail.com

ABSTRACT

In this manuscript, DVR operation is introduced and control strategy utilized for VSI is Space vector PWM (SVPWM). The SVPWM strategies could use the best DC voltage and produces less harmonic in "inverter output voltage". The phase jump compensation is accomplished by utilizing "Phase Locked Loop". As current vitality circumstance manages one difficult problem is Power quality (PQ). The PQ is logically relevant, concentrated, with extension of sensible equipment, where its conduct is particularly critical to PQ input supply. The problem because of PQ is a miracle as a phenomenal standard current, voltage repeat that achieves a mistake of progressive devices. The essential matter centers at voltage dips & extension. One of them, the DVR is the premier as similar advanced redid control equipment used in power dispersion systems. The customary type controller such as FLDVR Controller and Proportional-Integral one are used here for assessment. In suggested procedure, FLDVR controllers realized are displaced by regular PI controller to develop display of composite gird system. This manuscript depicts the DVR based on SVPWM including PLL gives voltage helps to sensitive loads and is reproduced by utilizing SIMULINK/MATLAB.

Keywords: *Dynamic Voltage Restorer (DVR), voltage source inverter (VSI), Fuzzy logic based DVR (FLDVR) controllers, steady-state error (SSE), Sinusoidal Pulse Width Modulation technique (SPWM), fuzzy-logic adjustor (FLA), Power quality (PQ), Membership function (MF), Space Vector Pulse Width Modulation technique (SVPWM), adaptive fuzzy dividing frequency controller (AFDFC), generalized integrator control (GIC)*

1. INTRODUCTION

The power quality (PQ) is identified with the capacity of utilities to give electric power with no interference. The significant worry in electric industry is PQ issues to sensitive burdens. The PQ issues like harmonic distortion, sag, transient, swell, unbalance, and flicker might affect client gadgets, cause malfunctions and price on production loss. The high expense related with these troubles clarifies the expanding interest towards "voltage sag mitigation" procedures. The voltage sag is broadly identified as very significant PQ troubles. The voltage sags might just happen regularly than some other PQ issue does. In this manner, loss resulted because of voltage sag issue for client at end of load is enormous. This is particularly accurate in floating ground or ungrounded delta systems, whereas the unexpected modification in ground reference outcome in rise of "voltage on ungrounded phases". The swells might also be produced by unexpected load diminishes and switching on huge capacitor bank regularly causes an oscillatory transient. To deal this issue, custom power gadgets have been utilized. The DVR is the very productive and viable current custom power gadget utilized in "power distribution networks". Its appeal incorporates small

size, less price, and fast dynamic reaction to disturbance. The DVR is power electronic gadget, which will be utilized to infuse "3-phase voltage" in synchronism and in series with dissemination feeder voltages [4] and comparably it responds rapidly to infuse the proper voltage segment. In this paper, voltage swell and sag is remunerated utilizing DVR dependent on SVPWM. It is discovered that DVR dependent on SVPWM repays voltage swells and sags adequately contrast with SPWM. Here, FLDVR is suggested to update the power behavior of system and to provide moved "power stream control" under various functioning conditions of systems reliant upon power system, for discard the drawbacks of customary kind DVR [5]. For dealing with complex structure problems effectively, FL controllers have best plan in such way. The IEEE – 14 transport systems is deliberated for test survey, to explain the lead of FLDVR controller for updating power behaviors of transmission. The programming of MATLAB is acquired the proliferation, as it has failure of getting effectively incredible and quick assessments for requirements engaged with composite power system and tested with Energy analyzer and PQ. The "FL based controllers" compared with customary PI controller that will lift the structure behavior by using DVR. To update power behavior of system, under huge situations, suggested DVR has been used under significant kind of flaws.

2. DYNAMIC VOLTAGE RESTORER

The DVR be a series associated custom power gadget. Its principle work be the assurance of sensitive loads from any voltage troubles expects voltage output [2] [1]. It has been fundamentally comprises of control and power circuit. The DVR power circuit has 4 principle parts; voltage injection transformer, VSI, low pass filter, and DC energy storage gadget [4] [5] as displayed in Figure 1.

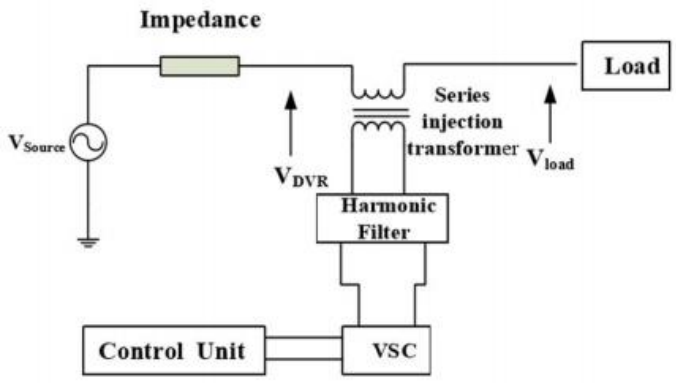


Figure 1: DVR Operation

2.1. Maintaining the Specifications Integrity

The VSI has been utilized to modify DC voltage given by storage gadget of energy to AC voltage. This voltage has been helped by infusion transformer to primary system. Typically, VSI rating will be low voltage and high current by reason of step-up infusion transformers utilization.

2.2. Voltage Injection Transformer

Its fundamental capacity is to step up AC low voltage provided by VSI to necessary voltage. In instance of 3-phase DVR type, 3 single phase infusion transformers have been normally utilized. The greatest voltage sag the DVR might compensate relies essentially upon infusion transformer and inverter rating.

2.3. DC energy storage gadget

It gives real power necessity of DVR through compensation. The "Super conducting Magnetic Energy Storage (SMES)", Lead- acid batteries, Super capacitors, and Flywheels might be utilized as storage gadgets [18] – [21]. For SMES & Batteries, "DC to AC inverters" has been essential, whereas for flywheels AC to AC transformation has been essential [5].

2.4. Passive Filter (PF)

A PF comprises of capacitor and inductor, it might be placed either at inverter side or high voltage side of injection transformer. It has been utilized to filter out changing harmonic parts from infused voltage. By assigning higher order harmonics have been kept from entering into transformer, the filter at inverter side, subsequently it lessen voltage stress on infusion transformer [5] [4].

2.5. DVR Operating Principle

A "power electronic converter based series compensator", which might protect loads all supply side problems other than outages is named as DVR. This gadget utilizes "IGBT solid state power electronic switches" in inverter structure of PWM. The DVR is able to absorbing or producing autonomously controllable reactive or real power at its "AC output terminal". The DVR has been completed of solid state DC to AC exchanging power converter, which infuses bunch of "3-phase AC output voltages" in synchronism and series with dispersion feeder voltages. The phase angle and amplitude of infused voltages are adaptable thus permitting control of reactive and real power interchange among DVR and dispersion system. The DC input terminal of DVR has been associated with energy storage gadget or energy source of appropriate capacity [17], [12]. The receptive force traded among dissemination system and DVR is inside produced by DVR without AC passive responsive segments. A distinctive DVR association is displayed in Figure 2. It has been associated in series with distribution feeder, which provisions a sensitive load.

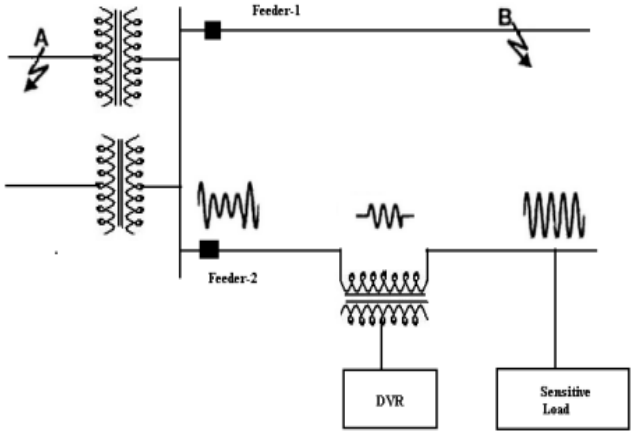


Figure 2: Connection of DVR for sensitive load's voltage sag correction

3. AFDFC TECHNIQUE

The conventional linear feedback controller have been used to enhance dynamic response or/and to enhance closed loop system's stability margin. Nevertheless, these controllers might introduce a poor SSE for harmonic reference signal. The AFDFC strategy is introduced in below Figure that comprises of 2 control

units: 1) FAU, 2) GIC unit. The GIC that might ignore impact of phase and magnitude will be utilized for frequency integral control, whereas fuzzy arithmetic has been utilized to timely change coefficients of PI.

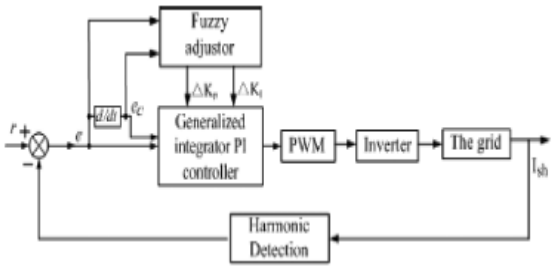


Figure 3: AFDFC Configuration

Since the reason for control plan is to get minimum SSE, r is the "harmonic reference signal" has been set to 0. Initially, supply harmonic current has been recognized. Then, at that point, the expectation inverter's control signal is uncovered by AFDFC. The system stability is accomplished by proportional controller, and perfect dynamic state has been gotten by GIC. The FA is set to change the boundaries of corresponding control and GIC. Hence, the suggested "harmonic current tracking controller" might diminish the "tracking error of harmonic compensation current", and have best dynamic robustness and response.

4. FUZZY ADJUSTOR (FA)

The FA will be utilized to modify parameters of integral control gain KI and proportional control gain K*P built on change of error ee and error e.

KP = K*P + ΔKp

KI + K*I + ΔKI

Here, K*P and K*I are the PI controller's reference values. In this manuscript, K*P and K*I are estimated offline rely on Ziegler–Nichols model. The improvement of rules needs a full knowledge of procedure to be managed; however, it does not need a calculated method of system. A flow-diagram representation of FLA is displayed in below figure.

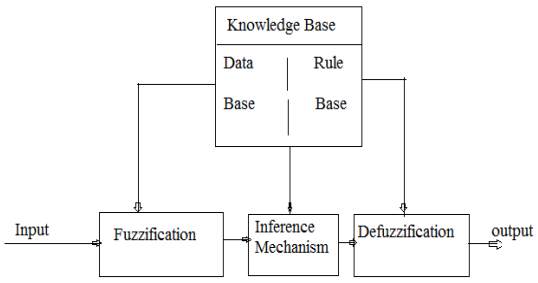


Figure 4a: Flow diagram of the fuzzy adjustor unit

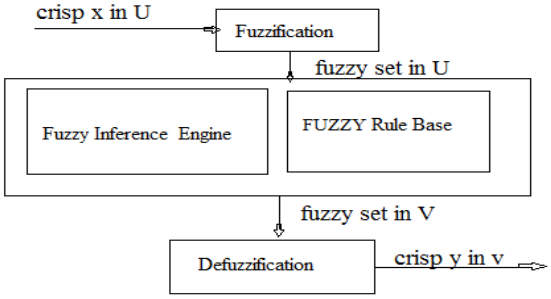


Figure 4b: Flow diagram of the fuzzy adjustor unit

In this way, quick dynamic reaction and stability of system with little overshoot might be accomplished with appropriate handling of FLA. The Fuzzification changes crisp information into fuzzy sets, creating it contented with fuzzy set depiction of state variable in standard. In fuzzification procedure, standardization by changing a scale transformation has been required at initially.

The "e" is the error and ee be the error changes have been utilized as numerical variables from real system. To change these numerical into linguistic variables, the subsequent 7 fuzzy sets or levels are selected as [17]: positive big (PB), positive medium (PM), positive small (PS), zeros (ZE), and negative small (NS), negative medium (NM), and negative big (NB). To ensure robustness and sensitivity of controller, the MF of fuzzy sets for e(k), ee(k), ΔKp and ΔKI in this manuscript are developed from ranges of e, ee ΔKp, and ΔKπ, those have been attained from experience and project. And MFs have been displayed in below figure.

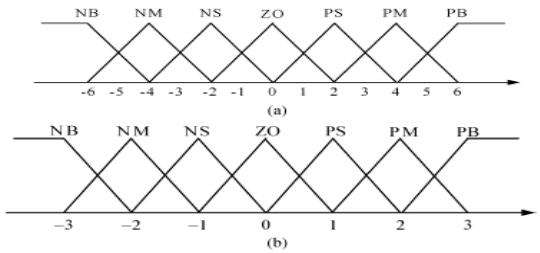


Figure 5: MFs of fuzzy variable. (a) MF of e(k)and e_e(k) (b) MF of ΔKp and ΔK_I

The fuzzy control rule plan includes describing rules, which associate input variables to output method. For planning the "control rule base for tuning" ΔK_P and ΔK_π, the subsequent significant parameters are taken into account.

1. For smaller values of e| |, a small ΔK_P is essential and for larger values of ln(e), a large ΔK_P is essential.
2. For $e.e_e < 0$, a smaller ΔK_P has been essential, and for, $e.e_e > 0$, a large ΔK_P is essential.
3. For larger values of $|e \text{ and } e_e|$, ΔK_I has been set to 0 that might evade control saturation.
4. For smaller values of |e, ΔK_I is effective, and ΔK_I is large while e| is smaller that will be best to diminish the SSE. So the tuning rules of ΔK_P and ΔK_I might be attained as Tables.

Table 1: Regulating Rule of ΔK_P Parameter

ΔKp		e_e						
		NB	NM	NS	0	PS	PM	PB
e	NB	PB	PB	NB	PM	PS	PS	0
	NM	PB	PB	NM	PM	PS	0	0
	NS	PM	PM	NS	PS	0	NS	NM
	0	PM	PS	0	0	NS	NM	NM
	PS	PS	PS	0	NS	NS	NM	NM
	PM	0	0	NS	NM	NM	NM	NB
	PB	0	NS	NS	NM	NM	NB	NB

Table 2: Regulating Rule of ΔK_I Parameter

ΔK_I		e_e						
		NB	NM	NS	0	PS	PM	PB
e	NB	0	0	NB	NM	NM	0	0
	NM	0	0	NM	NM	NS	0	0
	NS	0	0	NS	NS	0	0	0
	0	0	0	NS	NM	PS	0	0
	PS	0	0	0	PS	PS	0	0
	PM	0	0	PS	PM	PM	0	0
	PB	0	0	NS	PM	PB	0	0

The inference technique works the MAX-MIN technique. The "imprecise fuzzy control activity" produced from inference should be transformed to exact control activity in real applications.

$$K_P = P_P^* + \frac{\sum_{j=1}^{n} \mu_j(e, e_e)\Delta K_{Pj}}{\sum_{j=1}^{n} \mu_j(e, e_e)}$$

$$K_I = P_I^* + \frac{\sum_{j=1}^{n} \mu_j(e, e_e)\Delta K_{Ij}}{\sum_{j=1}^{n} \mu_j(e, e_e)}$$

The SVPWM strategy is a high level, computation–intensive PWM technique and is potentially the superior between whole PWM strategies for variable recurrence drive applications [13]. Due to its

predominant qualities and simple execution with advanced signal processors, it is discovering extensive application in present era. The "space vector modulation" is a profoundly effective approach to produce the 6 PWM pulses essential at "power phase for 2-level inverter". The circuit method of ordinary "3-phase voltage source PWM inverter" is displayed in figure 7. The 6 power switches are S_1 to S6, which shape the yield that are constrained by exchanging factors a, b, c, a′, b′, and c′. While upper semiconductor is turned on, i.e., while a, b or c is 1, the relating lower semiconductor is turned off, i.e., the comparing a′, b′ or c′ is 0. Accordingly, on & off conditions of upper semiconductors S_1, S_3 & S_5 might be utilized to define output voltage.

SPWM

In SPWM sinusoidal wave of fundamental frequency is the reference or modulating input and the carrier is a triangular wave form of very high frequency. The carrier frequency is constrained by the maximum operating frequency of the switching devices of the converter. Due to its several advantages over other switching techniques, such as easy implementation, lower THD and low switching losses, it is widely used.

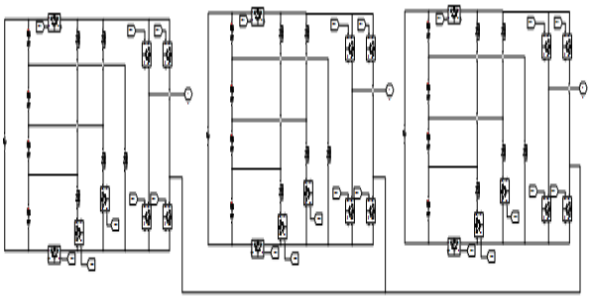


Figure 6: 3-Phase Voltage Source SVPWM Inverter DVR

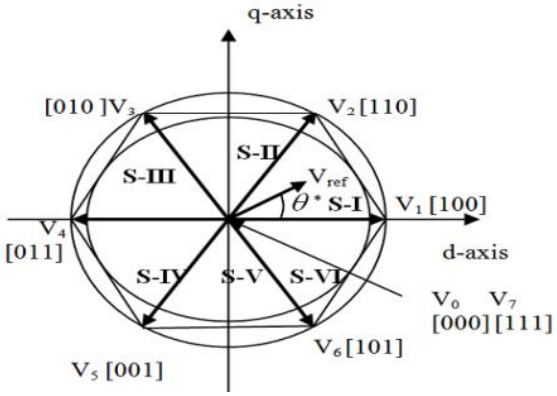


Figure 7: Switching Gates Vector Representations

A. Determination of V_q, V_d, V_{ref}, and angle (α)

From fig.7, the V_q, V_d, V_{ref}, and angle (α) might be defined as follows:

$$V_d = V_{an} - \frac{1}{2} V_{bn} - \frac{1}{2} V_{cn}$$

$$V_q = V_{an} + \frac{\sqrt{3}}{2} V_{bn} - \frac{\sqrt{3}}{2} V_{cn}$$

$$\alpha = tan^{-1}\frac{V_d}{V_q}_{an}$$

$$V = \sqrt{V_d^2 + V_q^2}$$

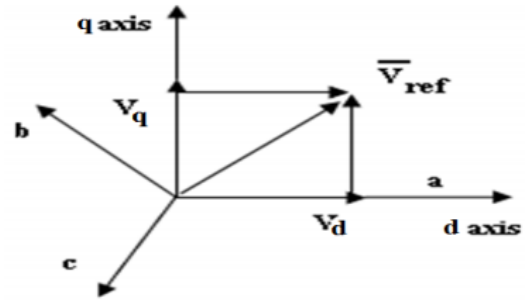


Figure 8: The "Voltage Space Vector" and its Modules in (α, β)

B. Time Periods T0, T1, T2, determination in figure 7. The time period of switching might be estimated below:

$$T_1 = \frac{\sqrt{3}}{V_{dc}} T_Z |V_{ref}| \left(sin\frac{n\pi}{3} cos\theta - cos\frac{n\pi}{3} sin\vartheta \right)$$

$$T_2 = \frac{\sqrt{3}}{V_{dc}} T_Z |V_{ref}| \left(sin\theta\ cos\frac{(\pi - 1)\pi}{3} - cos\theta\ sin\frac{(\pi - 1)\pi}{3} sin\vartheta \right)$$

$$T_2 = T_z - (T_1 + T_2)$$

$$T_0 = \frac{1}{f_z}$$

Here n=1 through 6, i.e. sector 1 to 6 and $0 \leq \theta \leq 60$

5. SIMULATION OUTCOMES AND DISCUSSION

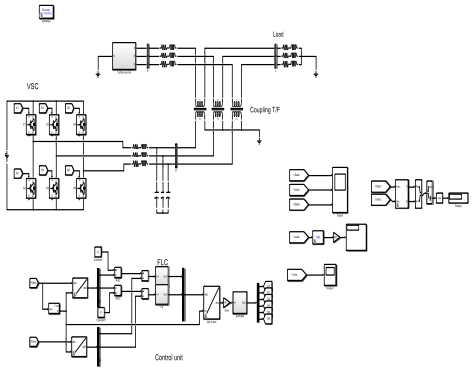


Figure 9: Simulation Main diagram of SVPWM & FLC based DVR

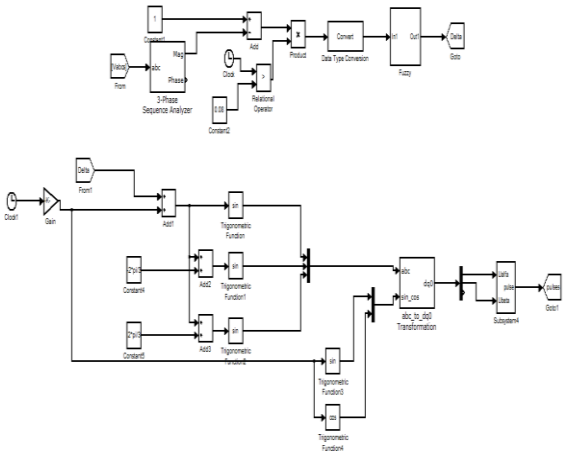


Figure 10: Fuzzy based SVPWM control structure

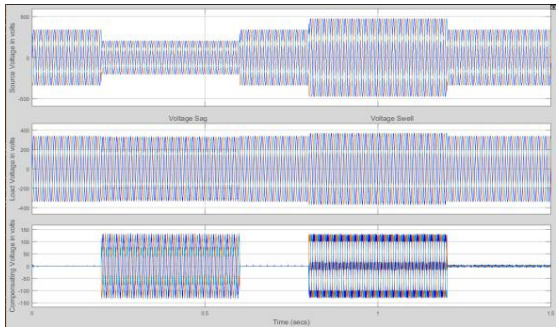


Figure 11: Voltage sag & swell & compensation voltage at load PWM control structure

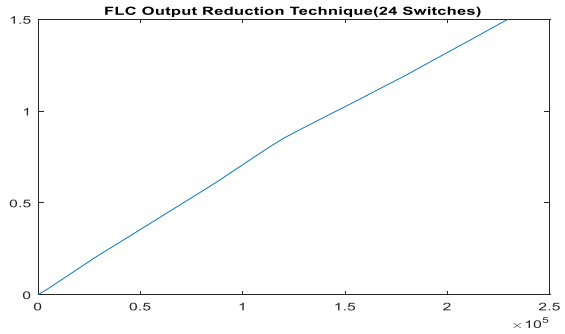


Figure 12: Output of the analysis of Fuzzy SVPWM based DVR

Table 3: Simulation Parameters of DVR System

S.NO.	Block Name	Parameter	Value
1	3 phase Programmable Voltage	Vrms	415Volts
		Frequency	50HZ
2	Transmission line	Resistance	0.1 Ω
		Inductance	0.001mH
3	3 phase VI Measurement	Line Voltage	------
4	3 Phase Two Winding Transformer	Y-Y	415V
		Power	5 KVA
		Frequency	50HZ
5	Linear Load	Resistance	8Ω
		Inductance	0.05mH
7	Load voltage		Vrms = 1000Volts
8	Load power		10KW
9	Switching Devices in MLI (IGBT)	Internal resistance	0.001Ω
		Snubber resistance	100KΩ
		Snubber Capacitance	Infinity or œ
10	3 phase Injection Transformer (Three phase linear T/F, 12 terminals))	Power Rating Winding1	5KVA Phase voltage Vrms = 415V R(pu) = 0.002Ω X(pu)=0.05Ω
		Winding 2 (phase voltages)	Vrms = 415V R(pu) = 0.002Ω X (pu) = 0.08Ω
		Frequency Rating	50HZ
		Magnetizing Branch	Rm(pu)=1100
			Xm(pu) = 1100
14	Low pass Filter	Resistance Inductance Capacitive	0.001Ohm 15mh 100Micro Farad
15	PI Controller	Kp	0.0523
		Ki	0.485
		output limits (Upper lower)	[1e6 -1e6]
16	Demux	No.of outputs	2
17	Mux	No.of inputs	1
18	abc to dqo , dqo to abc	-----	----

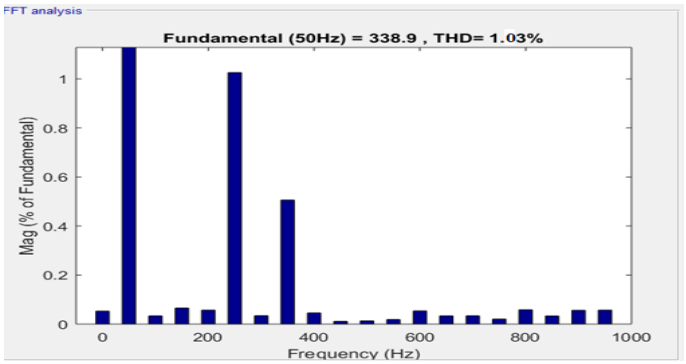


Figure 13: THD analysis of Fuzzy SVPWM based DVR (THD=1.03%)

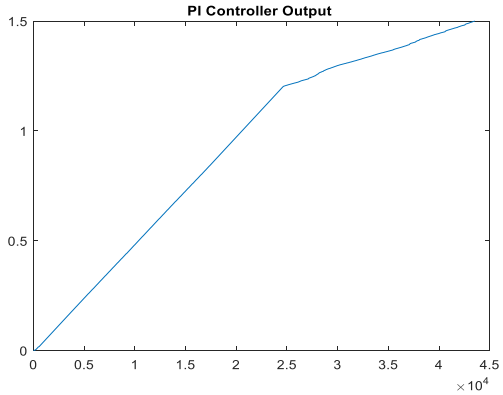


Figure 14: Wave forms of with PI Controller (SVPWM)

The Figure 14 shows the comparative study of the voltage with sag and with SVPWM technique during the fault as well as the compensated voltage by the SVPWM technique during the faulty conditions. The work basically focuses on the analysis of fault and implementation of DVR for mitigating voltage sag. DVR is being developed and designed so that it can meet various different load demands for power quality improvement. The study of this device is of importance because to overcome the newer problems. The performance of the DVR with an unbalanced voltage swell is shown in Figure 14. The injected voltage is generated by DVR to accurate the load voltage and the load voltage.

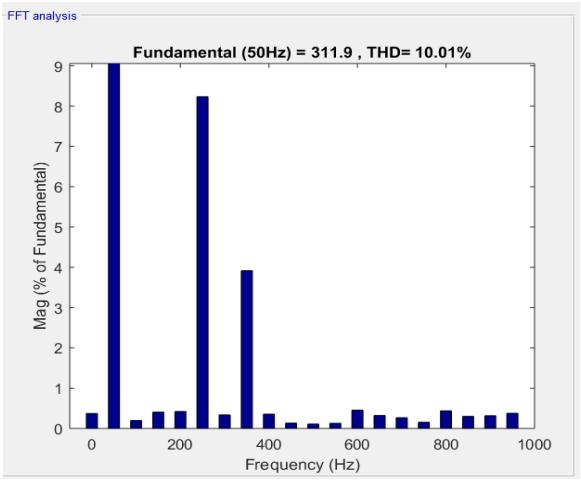


Figure 15: Fig 15: THD % of with PI Controller (SVPWM)

The THD of the proposed controller using SVPWM technique with PI controllers is shown in Fig 15. Here the compensation is analyzed in addition to voltage harmonics (5th 7th 11th and 13th). In this the THD is tremendously reduced from 11.00% to 10.01% by using the PI controller. At this harmonics of the DVR is works very effectively by using the SVPWM technique. The case studies verify that the proposed method performs very effectively and compensates voltages by using the PI controllers and it can be seen the clear difference between the PI and without PI controllers.

Table 4: Compensated Sag and Swell

	SPWM with FLC		SVPWM with FLC	
No. of Levels of DVR	%THD	PF	%THD	PF
3	7.63	0.847	7.37	0.85
5	6.40	0.88	6.00	0.89
7	4.97	0.973	3.78	0.9816
9	1.72	0.994	1.60	0.997

Table 5: Compensated Sag and Swell

	SVPWM with PI Controller		SVPWM with FLC	
No. of Levels of DVR	%THD	PF	%THD	PF
3	7.8	0.828	7.27	0.843
5	6.77	0.868	5.90	0.899
7	5.02	0.924	3.78	0.9516
9	1.92	0.9603	1.78	0.962

6. CONCLUSION

The DVR has been suggested cascaded solid state hardware that infuses possible contrasts into structure, to control the output side voltage at reliable. At reason for fundamental coupling, DVR has been usually united with distribution structure among critical load and input. For level of THD decrease in case of networks, which have been fixed to DVR, harmonic produced load has been required. The DVR voltage outputs diagrams using "FL type Controller" with SVPWM with voltage dip and amplification during 3 phase fault have been applied. The FL-based SVPWM DVR executes best between DVR with different sorts of controllers. Therefore, the recommended FL-based SVPWM DVR has high level accomplishments contrasted with other kind of controllers with respect of improvement in reactive and active flow of power through "transmission network lines".

REFERENCES

[1]. P. F. Comesana, D. F. Freijedo, J. D. Gandoy, O. Lopez, A. G. Yepes, and J. Malvar, "Mitigation of voltage sags, imbalances and harmonics in sensitive industrial loads by means of a series power line conditioner," Elect. Power Syst. Res., vol. 84, pp. 20–30, 2012.

[2]. A. Felce, S. A. C. A. Inelectra, G. Matas, and Y. Da Silva, "Voltage sag analysis and solution for an industrial plant with embedded induction motors," in Proc. IEEE Ind. Appl. Soc. Conf. Annu. Meeting., 2004, vol. 4, pp. 2573–2578.

[3]. A. Sannino, M. G. Miller, and M. H. J. Bollen, "Overview of voltage sag mitigation," in Proc. IEEE Power Eng. Soc. Winter Meeting, 2000, vol. 4, pp. 2872–2878.

[4]. E. Babaei, M. F. Kangarlu, and M. Sabahi, "Mitigation of voltage disturbances using dynamic voltage restorer based on direct converters," IEEE Trans. Power Del., vol. 25, no. 4, pp. 2676–2683, Oct. 2010.

[5]. G. MadhusudhanaRao, V Anwesha Kumar, B V Shankar Ram, "Damping control of DPFC for improving the transient stability using Fuzzy logic controller", IJECIERD, Vol. 6, Issue 5, 2016, pp.1-8

[6]. H. K. Al-Hadidi, A. M. Gole, and D. A. Jacobson, "A novel configuration for a cascade inverter-based dynamic voltage restorer with reduced energy storage requirements," IEEE Trans. Power Del., vol. 23, no. 2, pp. 881–888, Apr. 2008.

[7]. D. M. Vilathgamuwa, A. A. D. R. Perera, and S. S. Choi, "Voltage sag compensation with energy optimized dynamic voltage restorer," IEEE Trans. Power Del., vol. 18, no. 3, pp. 928–936, Jul. 2003.

[8]. N. A. Samra, C. Neft, A. Sundaram, and W. Malcolm, "The distribution system dynamic voltage restorer and its applications at industrial facilities with sensitive loads," presented at the Power Convers. Intell. Motion Power Qual., Long Beach, CA, USA, Sep. 1995.

[9]. Rosli Omarand and Nasrudin Abd Rahim, "Mitigation of Voltage Sags/Swells Using Dynamic Voltage Restorer (DVR)", VOL. 4, NO. 4, JUNE 2009, ISSN 1819-6608.

[10]. Pham D. T. and Liu X., "Neural Networks for Identification, Prediction and Control," Springer, Berlin, 1995.

[11]. Madhusudhana Rao G, and Sanker Ram B.V."A neural network based speed control for DC motor", International Journal of Recent Trends in Engineering (IJRTE), 2009, Vol 2, No 6, pp.121-124.

[12]. Jianjun Y. , Liquan W. , Caidong W., Zhonglin Z. , and Peng J., "ANN-based PID Controller for an Electro-hydraulic Servo System", in Proceedings of the IEEE International Conference on Automation and Logistics, 18-22. Qingdao, China, 2008.

[13]. Liu Luoren, and Luo Jinling, "Research of PID Control Algorithm Based on Neural Network", Energy Procedia 13, 2011, Science Direct, pp.6988- 6993.

[14]. G. MadhusudhanaRao, "TCSC Designed Optimal Power flow using Genetic Algorithm", International Journal of Engineering Science and Technology Vol. 2(9), 2010, 4342-4349

[15]. Thaha H S, Ruben Deva Prakash, "Reduction of Power Quality Issues in Micro-Grid using GA Tuned PI Controller Based DVR", International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN 2278-3075, Vol- 8, Issue-10 (August 2019) pp. 4166-4172.

[16]. Hossein Shahinzadeh, Majid Moazzami, S. Hamid Fathi, Gevork B. Gharehpetian, "Optimal Sizing and Energy Management of a Grid Connected Microgrid using HOMER Software", 2016 Smart Grids Conference (SGC), 20-21 Dec. 2016, Graduate University of Advanced Technology, Kerman, Iran.

[17]. Nagaraj B., Subha S., and Rampriya B "Tuning Algorithms for PID Controller Using Soft Computing Techniques", IJCSNS International Journal of Computer Science and Network Security, Vol.8 No.4, pp.278-281, 2008.

[18]. Shairul Wizmar Wahab and Alias Mohd Yusof, "Voltage Sag and Mitigation Using Dynamic Voltage Restorer (DVR) System", electrika, vol.8, No.2, pp. 32-37, 2006.

[19]. D. Ranjith Perera and S. S. Choi, "Performance Improvement of the Dynamic Voltage Restorer with Closed-Loop Load Voltage and Current-Mode Control, "IEEE Transactions on Power Electronics, vol. 17, no. 5, Sept. 2002.

[20]. N. H. Woodley, L. Morgan, and A. Sundaram, "Experience with an inverter-based dynamic voltage restorer," IEEE Transactions Power Delivery, vol. 14, pp. 1181–1186, Jul. 1999.

[21]. Hernandez, K.E. Chong, G. Gallegos, and E. Acha, "The implementation of a solid state voltage source in PSCAD/EMTDC,"IEEE power eng. Rev., pp. 61-62, dec. 1998.

Weiner filtered FRFCM image segmentation and CNN-SCA model and for Detection and Classification of Lungs related tissues

Satyasis Mishra*, Tadesse Hailu Ayane, Harish Kalla, Demissie Jobir Gelmecha, Dereje Tekilu, Davinder Singh Rathee

Dept.of ECE, SoEEC, Signal and Image Processing SIG, Adama Science and Technology Uni-versity, Adama, Ethiopia
**Corresponding author, e-mail: satyasismishra@gmail.com*

ABSTRACT

This research work proposes a novel Weiner filter-based fast and robust Fuzzy C Means (FRFCM) segmentation technique and Deep CNN-SCA model for detection and classification of covid-19 lungs affected tissues from chest images. As the chest images are X-Ray images, from which it is difficult to extract the diseased tissues, to avoid such a situation we are motivat-ed to apply the proposed FRFCM segmentation technique and achieved 99.21% of segmentation accuracy. The segmented images are applied to the proposed Deep CNN-SCA (Convolutional neural network with sine cosine algorithm) for classification of the type of diseased tissues for visual localization by the radiologists. In this research work, we have considered Xception, In-ceptionV3 for the purpose of comparison to the CNN-SCA proposed model. The chest image dataset has been collected from the Kaggle data repository. The proposed SCA-CNN model achieved an accuracy of 99.7252% and a computational time of 3127 seconds. Further, the pro-posed Deep CNN-SCA model will treat as a monitoring system to serve the patients affected by COVID-19. The proposed model will help doctors to identify and classify the covid-19 diseases with an automated system.

Keywords: *Fuzzy C Means (FCM); Convolutional Neural Network (CNN); Sine Cosine Algorithm (SCA)*

1. INTRODUCTION

According to the report, all countries followed lockdown to save their people from the virus ef-fect. COVID-19 affects drastically in the countries such as Italy, Spain, and Iran, US, Germany [1-4] directly. Ethiopia is also affected by corona, but the cases registered as per the source are much less as compared to other country cases. The patients are admitted to hospitals have to go through the process of CT-Scan image of the chest to identify the virus, but it is difficult for the doctors to get information from the images of the chest. The medical imaging method and deep convolutional neural networks (CNNs) are to predict the tissues in an automated fashion due to its fast computational time. The research will help the radiologists to handle the panic situation in a proper order. The clustering image segmentation is based on FCM (Fuzzy c means) which is efficient for images with simple texture and background [5]. Cai et al. [6] proposed the fast gen-eralized FCM algorithm (FGFCM) , Gong et al. [7] utilized a variable local coefficient Fuzzy Lo-cal Information C Means (FLICM) segmentation, Guo et al. [8] proposed an adaptive FCM algo-rithm based on noise detection (NDFCM), but more parameter involvement leads to higher itera-tions. Mishra et al.[9] employed Fast and robust FCM and LLRBFNN(Local Linear Radial Ba-sis function neural network) to reduce rician noise from the brain tumor MRI images and classi-fication. Motivated by this, in this research work, the Weiner- FRFCM segmentation technique has been proposed which employs morphological reconstruction (MR) to smoothen images and improve the noise-immunity.

The classifiers such as SVM, PNN, RBFNN, LLRBFNN, etc., have been proposed for the cancer-ous and noncancerous brain tumors. Ribli et al [10] used transfer learning to implement the Faster R-CNN model to classify these lesions into benign and malignant utilizing the MR images. Shen et al [11] presented a deep architecture to pledge weights of the full image in an end-to-end fash-ion. Huynh et al. [12] proposed a hybrid method that used both CNN and features of an SVM classifier with 5-fold cross-validations for classification. Pedro S. et al. (2020) [13] proposed EffiecintCovidNet along with a voting-based approach and a cross-dataset analysis with accura-cy of 87.6%. Athanasios V. et al. [14] presented a deep learning-based approach for se-mantic segmentation in CT images for the detection of COVID-19 induced symptoms in the lung area. Xiang Y. et al. (2020) [15] proposed an automatic detection system of COVID-19 by their proposed GoogLeNet-COD. The size of the dataset, which would significantly affect the per-formance of networks, is still quite limited. Dong Z. et al. (2020) [16] proposed a seven-layer standard convolutional neural network as background and integrating data augmentation and stochastic pooling methods. Matthew Z. et al. [17] created lung diseases classification pipeline based on transfer learning that was applied to small datasets of lung images. U-net seg-mentation network and InceptionV3 deep model classifier proposed for performance evaluation. Shayan H. et al. [18] presented three types of deep learning methods for the classification and segmentation of X-Ray images of patients' lungs infected by the COVID-19 virus. For the diagnosis of patients, they provided two methods of deep neural network (DNN) method on the fractal feature of input images and convolutional neural network (CNN) with direct use of CT scan images. Results classification shows that the presented CNN architecture with higher accu-racy (93.2%) and sensitivity (96.1%) is outperforming than the DNN method with an accuracy of 83.4% and sensitivity of 86%. The above CNN-based classification involves complex mathemat-ical calculations. Further, it is found from the literature that CNN consumes larger computational time for classification for magnetic resonance images. To get rid of such a situation, we are moti-vated to propose a novel Deep CNN-SCA model for the classification of COVID-19 diseases to improve the performance of conventional CNN classification. We considered Kaggle and Github [19-21] for data sets containing Chest X-rays. We divide the data sets into two main cate-gories, i.e., (a) Diseased Lungs (Covid-19) (b) Non diseased Lungs (Non-covid).

The rest of the article is organized as follows. Section 2 presents the implementation of the re-search flow diagram, brief details of covid related images, details of proposed FRFCM segmen-tation, and proposed Deep CNN-SCA model. Section 3 presents the details of segmentation and classification results. In section 4, the discussions of the research are presented, Section 5 pro-vides the concluding remarks for the article followed by references.

2. MATERIALS AND METHODS

2.1. Implementation

The research flow diagram shown in Figure 1 indicates the step by step accomplishment of the research work. Further the block diagram shows the flow of algorithm application for detection and classification of brain tumor.

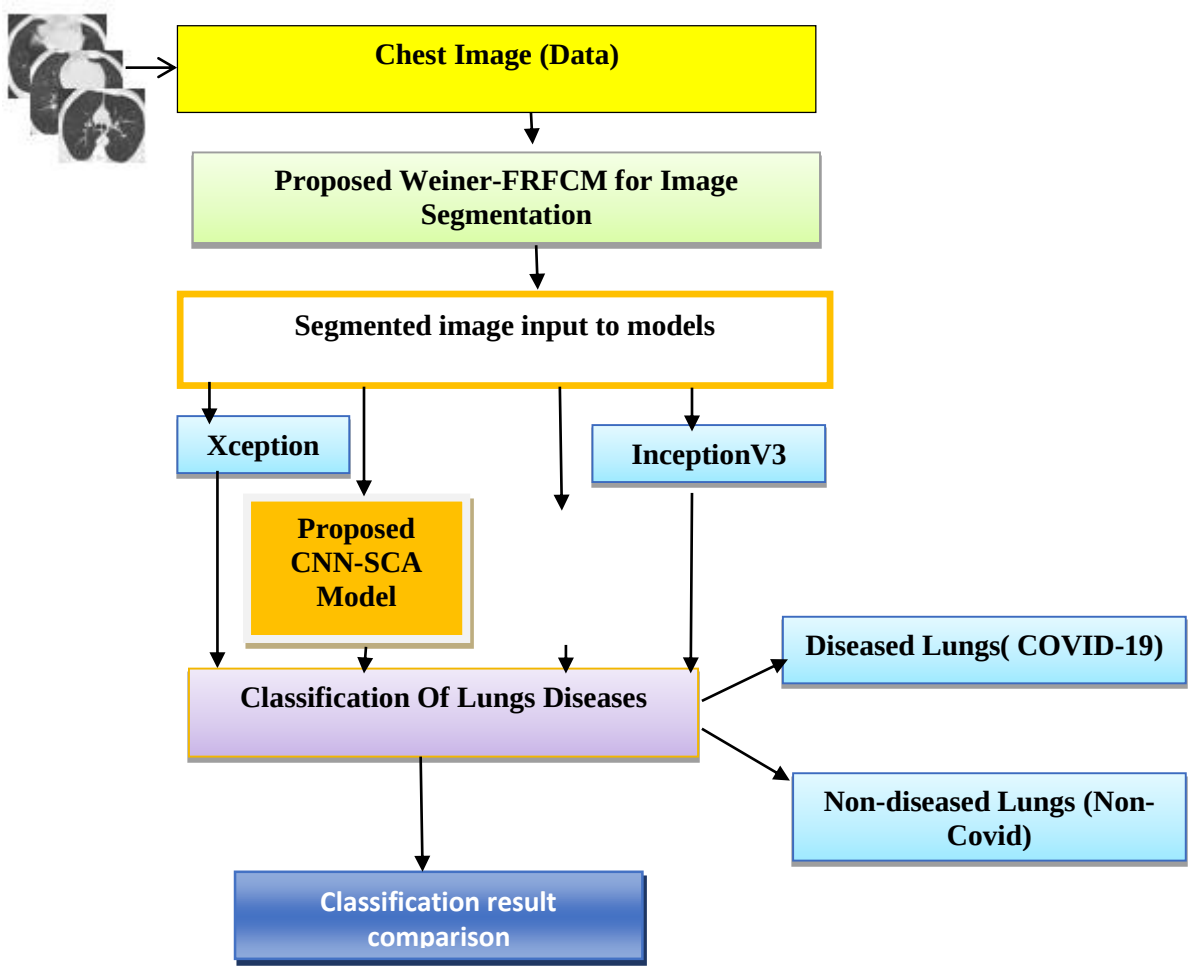


Figure 1: Research flow diagram

2.2. Details of Lungs disease related to Coronavirus

The coronaviruses symptoms are common cold to more Severe Acute Respiratory Syndrome (SARS) with infection can cause a severe viral respiratory illness. The COVID-19 dataset [19-21] was collected from the Kaggle website. We emphasized the statistical significance of these measures for the purpose of detection and classification of COVID-19 diseases also. Fig.2 and Fig.3 show the images of covid (Pneumonia) and non –covid.

2.3. Proposed FRFCM Image Segmentation

In this research work, FRFCM algorithm distinguishes the performance of segmentation than the other FCM based segmentation. The advantage of FRFCM algorithm is to detect the tissues and at the same time the noise also removed from the COVID-19 images to improve the performance of the segmentation. The member partition matrix U of FRFCM algorithm has been modified to improve the performance of the segmentation algorithm [5]. The proposed fast and robust FCM segmentation uses a Weiner filter to the modified membership partition matrix of the objective function of FCM algorithm with local information [5,9]. The objective function of the fuzzy c means algorithm with local information is given by

$$J_s = \sum_{v=1}^{N}\sum_{k=1}^{c} \quad u_{\mathrm{kv}}^{m}\|x_v - v_k\|^2 + \sum_{v=1}^{N}\sum_{k=1}^{c} G_{\mathrm{kv}} \tag{1}$$

where, the fuzzy factor is given by

$$\sum_{v\neq r} \quad r \in N_v(1-u_{\mathrm{kr}})^m\|x_r - v_k\|^2 \tag{2}$$

where, the spatial Euclidean distance between pixels x_v and $x_r$is denoted byd_{vr}, $N_v$is the set of neighbours within a window around x_v and $x_r$represents the neighbours ofx_v and $u_{\mathrm{kr}}$is the neighbours ofu_{kv}. With respect to cluster$k$,x_v is the gray value of the k^{th}pixel, u_{kv} represents the fuzzy membership value of the v^{th}pixel and $N$is the total number of pixels in the gray scale image$f = [x_1, x_2 ..... x_N]$, x_v is the gray value of v^{th} pixel, c denotes the cluster centre and m determines the fuzziness of the consequential partition.

To reduce the computational complexity, the membership partition matrix is modified as

$$\sum_{v\neq r} \quad r \in N_v u_{\mathrm{kr}}^{m}\|x_r - v_k\|^2 \tag{3}$$

where, $u_{\mathrm{kr}}$is the neighbours ofu_{kv}, $\xi$is gray value of image and $\tau$is the smoothness parameter between 0 and 1. Further, considering the morphological reconstruction operations such as dilation and erosion, the reconstruction of the image is considered as ξ_p, which is given by

$$\xi_p = R_e^C(f) \tag{4}$$

where, R_e^Crepresents the morphological closing reconstruction which is efficient for noise removal and f denotes an original image and reconstruction operators considering morphological closing reconstruction is given by

$$R_e^C(f) = R^{\chi}_{R_f^{\beta}(\chi(f))}\left(\beta\left(R_f^{\beta}(\chi(f))\right)\right) \tag{5}$$

where, $\chi$is the erosion operation, $\beta$is the dilation operation,$c$is the closing operation and$f$ represents original image.

Further, considering convergence speed of the algorithms and the performance of the partition matrix U we employ a wiener filter**.** The new membership partition matrix is given by

$$U^l = \text{wiener}[U] \tag{6}$$

Update U^l according to equation (6) until convergence of objective function in programming.

2.4. Proposed Deep CNN-SCA Model

The deep CNN-SCA is proposed to reduce the computational time of conventional CNN. We are proposing the SCA (Sine Cosine Algorithm) [9, 21-22] to optimize the weights of CNN due to the robustness of the optimization capability as compared to accelerated particle swarm optimization, genetic algorithm, etc. The SCA algorithm with CNN is considered to improve the performance of CNN. Basically, the Deep CNN is modeled with a back propagation algorithm for weight optimization. Due to complex mathematical calculation and backward propagation from the last layer to the first layer during weight optimization consume larger time for classification [5]. The WCA-CNN [5] has already been developed and achieved good classification results, but the computational time is more. To improve further the computational time, we have employed SCA for weight optimization of the CNN fully connected layer. The CNN-SCA model for classification of lungs diseases is presented in **Fig.2**. To improve the performance of Deep CNN model, the weights of the fully connected layer are optimized with a novel SCA algorithm

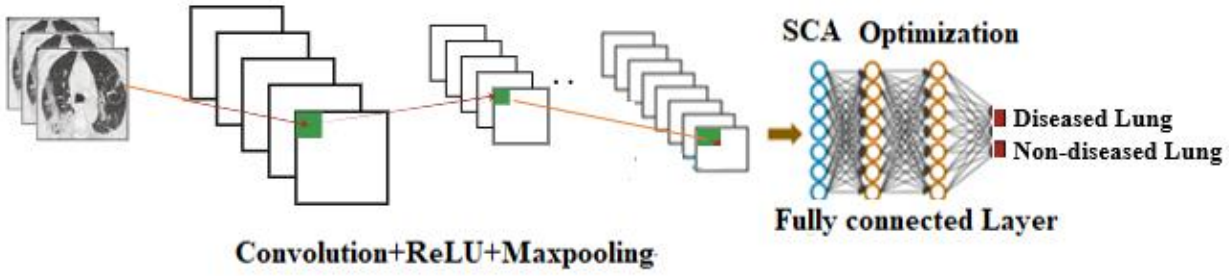


Figure 2: Deep CNN-SCA model for Lung Diseases Classification

The classification model will classify the different categories of diseases such as diseased lungs (Covid), and non-diseased lungs (Non-Covid).

2.5. Sine Cosine Algorithm (SCA)

Due the complexity involved in learning parameter, the position equation has been utilized for weight optimization at the fully connected layer. The mathematical calculations are presented for the modified SCA algorithm [22-23] to maximize the performance of the Deep CNN.

According to sine cosine algorithm, the position equation is updated as

$$X_i^{n+1} = \begin{cases} X_i^n + \alpha_1 \times \sin(\alpha_2) \times \left| \alpha_3 p^{gbest} - X_i^n \right|, & \alpha_4 < 0.5 \\ X_i^n + \alpha_1 \times \cos(\alpha_2) \times \left| \alpha_3 p^{gbest} - X_i^n \right|, & \alpha_4 \geq 0.5 \end{cases} \quad (7)$$

where, $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are the random variables and α_1 is given by

$$\alpha_1 = a\left(1 - \frac{n}{K}\right) \quad (8)$$

where, n is the current iteration, K is the maximum number of iterations.

The X_i^n represents the current position and X_i^{n+1} represents the update position. Here $|.|$ represents the absolute value. The parameter α_1 determines the next position regions of the search and explores to search the space to a higher value. The parameter α_2 represents the direction of movement of towards or away from $x_i(n)$ .The parameter α_3 controls the current movement, during each iteration $\alpha_1, \alpha_2, \alpha_3$ are updated and the parameter α_4 equally switches between the sine and cosine functions

For fast convergence of the parameter α_1 is modified as

$$\alpha_{11} = \frac{1}{1 + \exp(\alpha_1)} \quad (9)$$

And the corresponding position equation is given with modification factor presented as

$$X_{ij}^{n+1} = \begin{cases} X_{ij}^n + \alpha_{11} \times \sin(\alpha_2) \times \left| \alpha_3 y_i^n - X_i^n \right|, & \alpha_4 < 0.5 \\ X_{ij}^n + \alpha_{11} \times \cos(\alpha_2) \times \left| \alpha_3 y_i^n - X_i^n \right|, & \alpha_4 \geq 0.5 \end{cases} \quad (10)$$

Now considering the fully connected layer weights the optimization of weights $W = [W_1, W_2, \ldots\ldots\ldots W_n]$, and corresponding to the position equation are followed as

$$W_{ij}(n+1)=\begin{cases}W_{ij}(n)+\alpha_{11}r_1\sin(\alpha_2)\times\left(\alpha_3 y_i^n-W_i(n)\right)\\ W_{ij}(n)+\alpha_{11}r_1\cos(\alpha_2)\times\left(\alpha_3 y_i^n-W_i(n)\right)\end{cases} \quad (11)$$

With this update equation the algorithm continues to update the weights of the proposed CNN—SCA model till convergence achieved.

3. RESULTS

3.1. COVID-19 Medical image data-sets

Medical images in the form of Chest CT scans and X-rays [24, 25] are essential for automated COVID-19 diagnosis. This research followed the article for data [25, 26] where COVID-Net, a deep convolutional network for COVID-19 diagnosis based on Chest X-ray images are presented.

3.2. Preprocessing Results

Data augmentation is possibly stochastic transformations on the existing examples. These transformations are, slight translations or rotations, which preserve the perceptual appearance of the original images, but significantly alter the actual pixel values [5]. We performed the experiments on the Chest X-ray image data sets. We resized the 1.3 M images into 227 x 227 pixels, as a compromise between keeping a high resolution and speeding up the training. The pre-processing involves **Random Rotations,** Random Shifts and Random Flips. The details of data augmentation for training, testing and validation are presented in **Table-1** and **Table-2**.

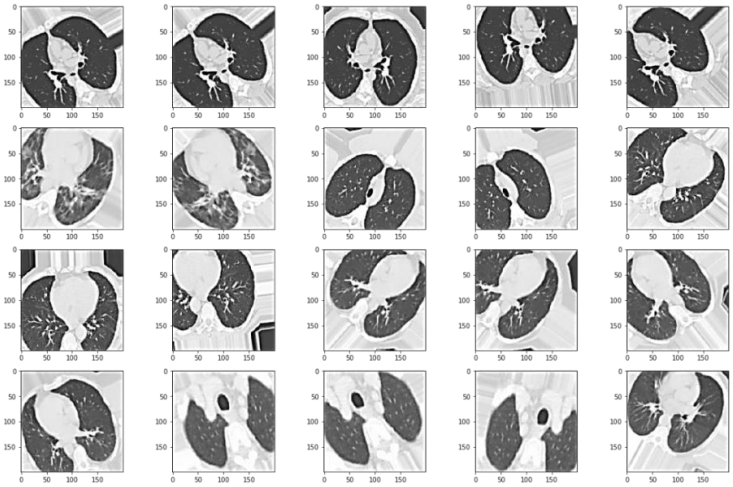


Figure 3: Output images of Image data augmentation techniques rotation, shifts and flips.

A higher number of images were needed to train a deep learning algorithm. The data augmentation technique was used to increase the number of samples in the dataset is presented in **Fig.3**. After the data augmentation, we obtained 15,216 samples [5].

Table 1: Original Images

COVID			Non-COVID		
Training	Validation	testing	Training	Validation	testing
1252	250	126	1230	250	124
Total: 1628			Total: 1604		

Table 2: Data Augmentation

Training	Validation	testing
159,472	160	79,799
	Total: 239,431	

3.3. Segmentation Results

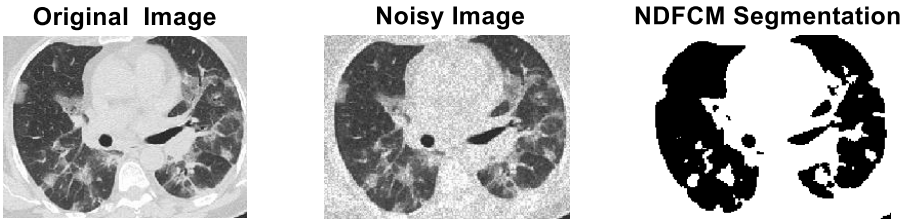


Figure 4. NDFCM segmentation results

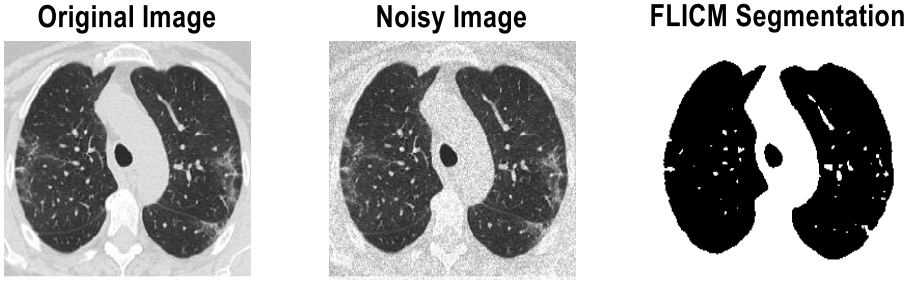


Figure 5: FLICM segmentation results

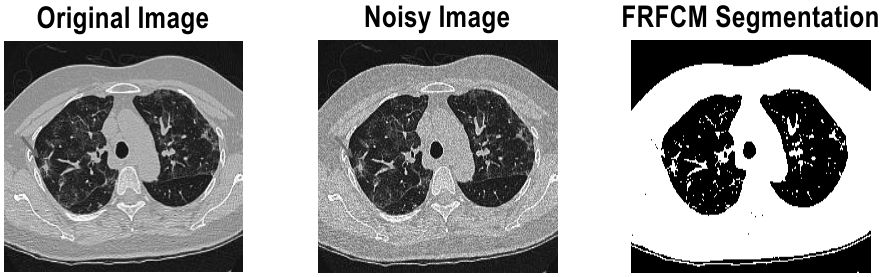


Figure 6: FRFCM segmentation results

Table 3: Segmentation Accuracy

	Noise level	
Algorithm	Rician noise(σ_n=10)	Rician noise(σ_n=20)
En FCM	91.28	88.56
FGFCM	96.85	92.11
NDFCM	97.85	95.28
FLICM	98.27	96.84
FRFCM	**99.21**	**98.56**

3.4. Classification Results

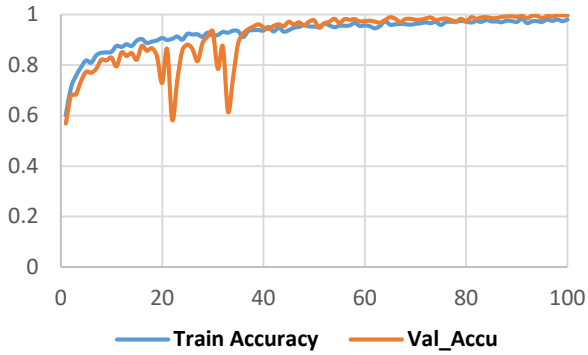


Figure 7: InceptionV3Model accuracy results

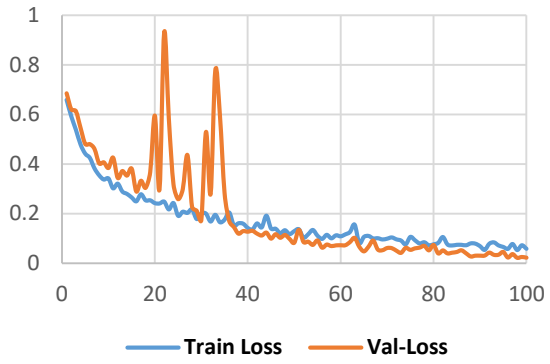


Figure 8: InceptionV3Model loss results

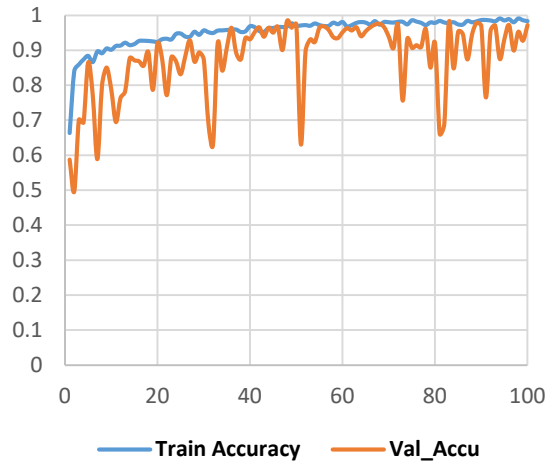


Figure 9: XceptionModel accuracy results

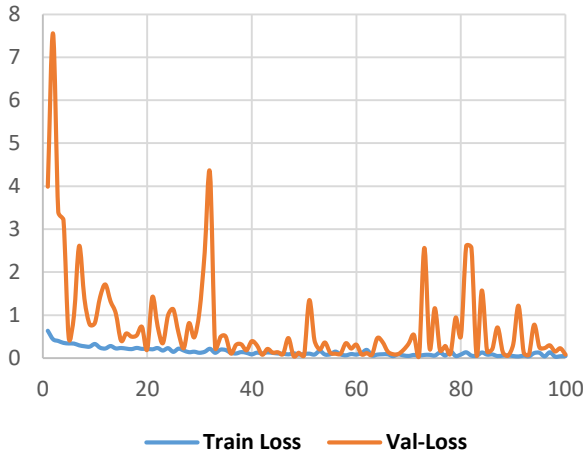


Figure 10: Xception Model loss results

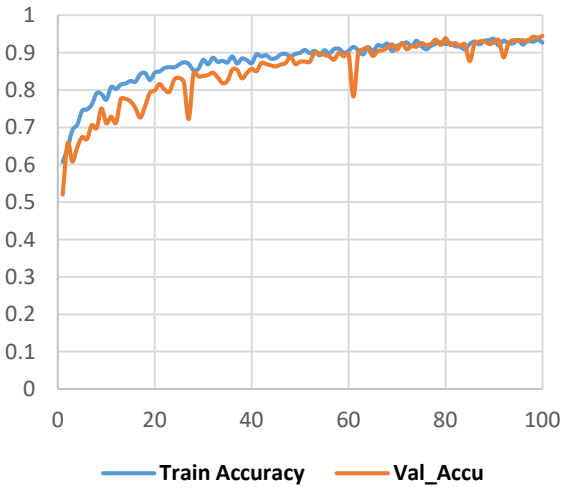


Figure 11: Deep CNN-WCA Model accuracy

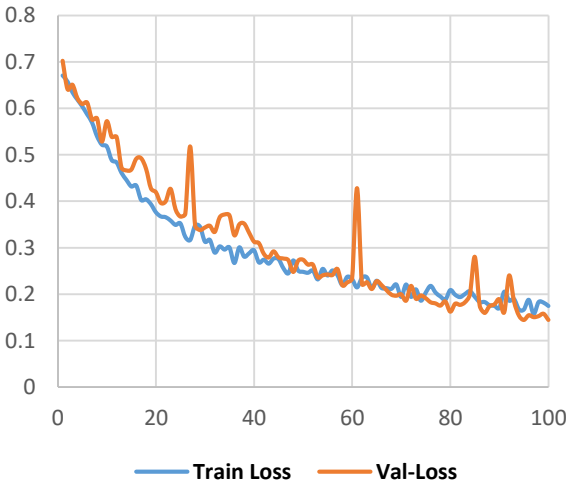


Figure 12: Deep CNN-WCA Model loss

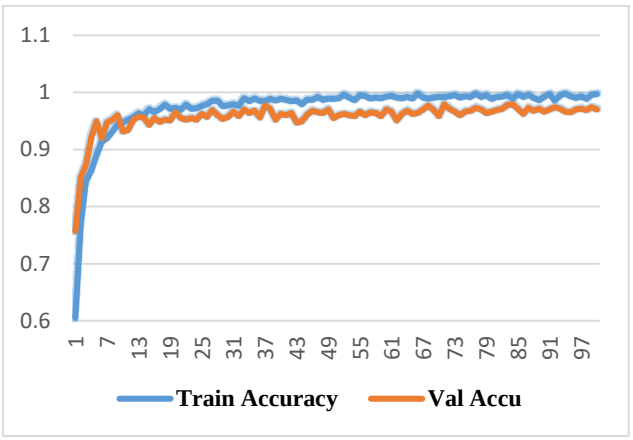


Figure 13: Deep CNN-SCA Model accuracy

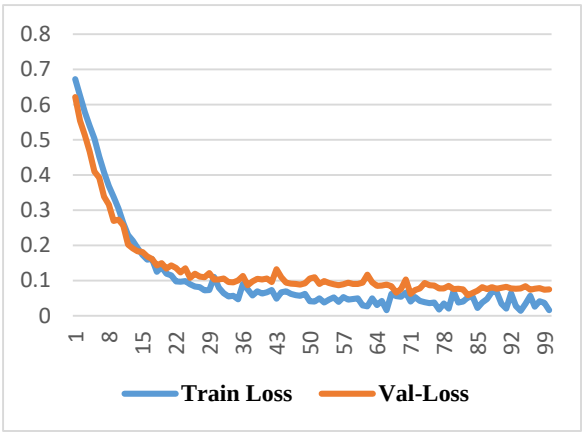


Figure 14: Deep CNN-SCA Model loss results

Table 4: Classification accuracy results

Model	Training accuracy	Validation accuracy	Training Loss	Validation Loss	Computational Time in sec
InceptionV3[5]	93.0986	91.5259	9.9014	8.4741	4033
Xception[5]	95.2347	92.3879	4.7653	7.6121	3905
Deep CNN-WCA[5]	99.6243	94.1523	0.3757	5.8477	3216
Proposed Deep CNN-SCA	99.7252	94.2551	0.3242	5.3215	3127

Note: All the values are the average values of 100 epochs

Precision:- Accuracy of positive predictions.

Precision = TP/(TP + FP)

Recall:- Fraction of positives that were correctly identified.

Recall = TP/(TP+FN)

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Where TP=True positive, TN=True Negative, FP=False Positive, FN=False Negative

Table 5: Classification Measure of Proposed Deep CNN-SCA

	Precision	Recall	F1-Score
Non-diseases Lungs	0.96	0.97	0.98
Diseased Lungs COVID	0.68	0.59	0.63

4. DISCUSSION

Fig.3 shows Image data augmentation techniques which include rotation, shifts, and flips**. Fig.4** and **Fig.5** shows the results of NDFCM and FLICM**. Fig.6** shows the image segmentation by utilizing the proposed FRFCM algorithm. The results proposed by the fast generalized FCM algorithm (FGFCM)[6] , Fuzzy Local Information C Means (FLICM)[7] segmentation, noise detection Fuzzy C Means (NDFCM) are presented in **Fig.4 to Fig.6**. The rician noise with standard deviation of 10 and 20 are considered for this experimentation. It is found that the FRFCM segmentation shows the better rician noise reduction capability in comparison to other segmentation techniques. The segmentation accuracies are presented in

Table-3.The segmentation has been utilized to remove rician noise from the images and the images are fed as input to the Deep CNN-SCA model for classification. **Fig.7, Fig.9** and **Fig.8, Fig.10,** shows the training accuracy, validation accuracy and training loss and validation loss results of the inceptionV3 model, Xception model. **Fig.11** and **Fig.12** shows the training and validation accuracy ald corresponding loss of the Deep CNN-WCA model. The Deep CNN-SCA model classifies the images into Covid and non-covid categories and the accuracies and loss of training and validations are presented in **Fig.13** and **Fig.14**. Further, the computational time and training and validation accuracy of classification models are presented in **Table-4**. **Table-5** presents the precision, recall and F1-Score of the proposed Deep CNN-SCA model.

5. CONCLUSION

This research work presents a novel CNN-SCA model for classification of lungs diseases. The FRFCM segmentation has been employed to identify the region of tissues and remove rician noise from the chest x-ray images. The preprocessed images are then fed to a novel Deep CNN-SCA model for the classification of diseased and non-diseased lungs diseases form the chest images related to covid categories. In the architecture of the Deep CNN model, the fully connected layer, in general, plays an important role in classification. The back propagation algorithm is generally utilized for weight optimization. In this research work, the meta-heuristic SCA algorithm has been employed for the optimization of weights in the fully connected layer. The results of training and validation classification accuracies are presented. It is observed that the proposed Deep CNN–SCA model outperforms than the inception V3, CNN-WCA and Xception model. Also, the computational time required is less in the proposed Deep CNN-SCA model in comparison to the other mentioned models which is presented in Table-4. Different optimization techniques such as SGD, RMSProp, and SCA are utilized in the CNN models. All these models and optimization techniques are conducted simultaneously for 100 epochs. The experiments are performed on a high-performance computer using NVIDIA RTX2060 GPU system. The simulations for the models are performed using the python platform with GPU.

REFERENCES

1. American Lung Association -Learn about lung disease symptoms, causes and treatments, as well as advice for recognizing and managing lung diseases. April 14, 2021.
2. World Health Organization - Clinical management of severe acute respiratory infection (SARI) when COVID-19 disease is suspected_ Interim guidance V1.2 (13 March 2020)-World Health Organization (2020).
3. Grant,W.B.; Lahore, H.; McDonnell, S.L.; Baggerly, C.A.; French, C.B.; Aliano, J.L.; Bhattoa, H.P. Evidence that Vitamin D Supplementation Could Reduce Risk of Influ-enza and COVID-19 Infections and Deaths. Nutrients 2020, 12, 988.
4. Yang X, Yu Y, Xu J, Shu H, Xia J, Liu H et al. Clinical course and outcomes of criti-cally ill patients with SARS-CoV-2 pneumonia in Wuhan, China: a single-centered, ret-rospective, observational study. Lancet Respir Med. 2020. Epub 2020/02/28. doi: 10.1016/S2213-2600(20)30079-5. PubMed PMID: 32105632.
5. Mishra S., Kalla H, Tekilu D., Ayane TH., Artificial intelligence CNN-WCA model and Weiner filtered FRFCM image segmentation technique for extraction and classification COVID-19 Virus, ASRIC Journal on Health Science 1 (2021) 18-32.
6. Cai W., Chen S., and Zhang D., "Fast and robust fuzzy c-means clustering algorithms in-corporating local information for image segmentation," Pattern Recognit., vol. 40, no. 3, pp. 825-838, Mar. 2007.

7. Gong M, Liang Y, Shi S, Ma J , Fuzzy c-means clustering with local information and kernel metric for image segmentation. IEEE Trans Image Process., 22:573-580,2013.
8. Guo F, Wang X, Shen J (2016) Adaptive fuzzy c-means algorithm based on local noise detecting for image segmentation. IET Image Process, 10:272-285,2016.
9. Mishra S, Sahu P, Senapati MR, MASCA- PSO based LLRBFNN Model and Improved fast and robust FCM algorithm for Detection and Classification of Brain Tumor from MR Image, Evolutionary Intelligence, 12: 647-655, 2019
10. Ribli D, Horváth A, Unger Z, Pollner P, Csabai I. Detecting and classifying lesions in mammograms with deep learning. Sci Rep 2018 Mar 15;8(1):4165 [FREE Full text] [doi: 10.1038/s41598-018-22437-z] [Medline: 29545529]
11. Shen L, Laurie LM, Joseph HR, Eugene F, Russell B, Weiva S. Cornell University. 2017. End-to-End Training for Whole Image Breast Cancer Diagnosis Using an All Convolu-tional Design URL:https://arxiv.org/pdf/1708.09427.pdf
12. Huynh BQ, Li H, Giger ML. Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. J Med Imaging (Bellingham) 2016 Jul;3(3):034501 [FREE Full text] doi: 10.1117/1.JMI.3.3.034501] [Medline: 27610399
13. Pedro S., COVID-19 detection in CT images with deep learning: A voting-based scheme and cross-datasets analysis.2020, https://doi.org/10.1016/j.imu.2020.100427
14. Athanasios V., Eftychios P., Iason K., Anastasios D., Nikolaos D. , Deep learning mod-els for COVID-19 infected area segmentation in CT images, cold spring harbor laborato-ry. 2020 https://doi.org/10.1101/2020.05.08.20094664
15. Xiang Y., Shui-Hua W., Xin Z., Yu-Dong Z.: Detection of COVID-19 by GoogLeNet-COD, 2020, computer science journal.
16. Yu-Dong Z., Suresh C. S., Li-Yao Z., Juan M. G., Shui-Hua W. (2020): A seven-layer convolutional neural network for chest CT based COVID-19 diagnosis using stochastic pooling, IEEE Sensors Journal . DOI: 10.1109/JSEN.2020.3025855
17. Matthew Z. and Adam K., Classification of Lung Diseases Using Deep Learning Models, Springer Nature Switzerland, pp. 621–634. https://doi.org/10.1007/978-3-030-50420-5_47
18. Shayan H., Mohsen A., and Abbas S., Diagnosis and detection of infected tissue of COVID-19 patients based on lung x-ray image using convolutional neural network ap-proaches, Chaos, Solitons and Fractals, Vol.140,pp:1-11, 2020,110170, ISSN 0960-0779,https://doi.org/10.1016/j.chaos.2020.110170. Elsevier Ltd.
19. Shuja, J., Alanazi, E., Alasmary, W. et al. COVID-19 open source data sets: a compre-hensive survey. Appl Intell (2020). https://doi.org/10.1007/s10489-020-01862-6.
20. F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, K. He, Y. Shi, and D. Shen. Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19. IEEE Reviews in Biomedical Engineering, pages 1–1, 2020.
21. Roman Kalkreuth and Paul Kaufmann. Covid-19: A survey on public medical imaging data resources. arXiv preprint arXiv:2004.04569, 2020.
22. Seyedali M., (2016), SCA: A Sine Cosine Algorithm for Solving Optimization Problems, Knowledge-Based Systems. doi: 10.1016/j.knosys.2015.12.022
23. Ahmed I. H.,et al. (2016): Sine Cosine Optimization Algorithm for Feature Selection, IEEE.
24. Linda Wang and Alexander Wong. Covid-net: A tailored deep convolutional neural net-work design for detection of covid-19 cases from chest radiography images. arXiv pre-print arXiv:2003.09871, 2020.
25. Joseph Paul Cohen, Paul Morrison, and Lan Dao. Covid-19 image data collection. arXiv preprint arXiv:2003.11597, 2020.