Journal of Internet Services and Applications, 2023, 14:1, doi: 10.5753/jisa.2023.3076

Analysis of Electrical Signals by Machine Learning for Classification of Individualized Electronics on the Internet of Smart Grid Things (IoSGT) architecture

Leonardo Marques [**Federal University of Rio Grande do Norte** | ***leonardo.augusto.103@ufrn.edu.br***]
Paulo Eugênio [**Federal University of Rio Grande do Norte** | ***paulo.filho.071@ufrn.edu.br***]
Lucas Bastos [**Federal University of Pará** | ***lucas.bastos@itec.ufpa.br***]
Hugo Santos [**Federal University of Pará** | ***hugosantos@ufpa.br***]
Denis Rosário [**Federal University of Pará** | ***denis@ufpa.br***]
Eduardo Nogueira [**Federal University of Rio Grande do Norte** | ***eduardo.nogueira@ufrn.br***]
Eduardo Cerqueira [**Federal University of Pará** | ***cerqueira@ufpa.br***]
Márcio Kreutz [**Federal University of Rio Grande do Norte** | ***marcio.kreutz@ufrn.br***]
Augusto Neto [**Federal University of Rio Grande do Norte** | ***augusto@dimap.ufrn.br***]

DIMAp - Department of Informatics and Applied Mathematics, Federal University of Rio Grande do Norte, Campus Universitário - Lagoa Nova, Natal, RN, 59078-970, Brazil.

Received: 26 December 2022 • **Accepted:** 04 May 2023 • **Published:** 23 August 2023

Abstract This paper presents a study on the performance of Machine Learning techniques for the task of determining each of the electronics systems connected within the same electrical network, in the Internet of Smart Grid Things (IoSGT) architecture. This is regarded as an ecosystem which has cutting-edge technologies that work together to enable advanced SG applications, and run in core/edge cloud datacenters connected to an underlying IoT network infrastructure. This identification was carried out by analyzing traditional energy measures (i.e., voltage, current and power) through ML-based techniques. The analysis of regular energy measurements is required as a means of ensuring compatibility with legacy /smart meters, without the need to exchange them for new ones or make updates. The data was obtained by using a smart meter built by our group, and was processed and validated in the IoSGT edge-cloud.

Keywords: Smart Grid; Machine Learning; Power Data Analysis; Electric Equipment Identification

1 Introduction

The increase in population density on the planet has led to an exponential growth in energy consumption. However, as this was based on large (preponderant) generating plants often far from the main urban centers, the electrical system modeling software has not accompanied this exponential growth at the same rate. This has caused a number of challenges that need to be settled, especially those related to energy distribution for high-density environments, such as outages, interruptions, transmission losses, deterioration of legacy systems, and low energy efficiency [Barja-Martinez *et al.*, 2021]. Designing solutions that can fill the gaps in these areas rests on the premise of generating distributed energy and, at the same time, handling control mechanisms that are capable of supporting a systematic interaction between energy generators, energy distributors, and end consumers. In general terms, Samart Grid (SG) [Gopstein *et al.*, 2021] allows a consolidated approach to be adopted that can lead to a paradigm shift in the existing power grid systems [Wang *et al.*, 2019; Alonso *et al.*, 2022]. SG allows the monitoring and management of electric power transmission in real time.

The intelligence of SG is acquired by adopting specific sensors/actuators (called IEDs, Intelligent Electronic Devices) that are capable of operating telemetry and ensuring the supply and consumption of energy. According to the National Institute of Standards and Technology (NIST) [Keith Stouffer and Hahn, 2015], an IED stands for ”any device incorporating one or more processors with the capability to receive or send data/control from or to an external source (e.g., electronic multifunction meters, digital relays, controllers).” The microprocessor-based power system approach that IEDs adopt, makes it possible to harness both the networking (sending/receiving data/control from or to external systems) and to handle computerized control and automated functions.

The bidirectional data and energy flow that SG provisions in all sectors of the grid system, mean that both IEDs and Supervisory Control and Data Acquisition (SCADA) have essential architectural components that are interlinked to assist utilities in achieving reliability and efficiency [Hajam and Sofi, 2021]. Such an interlinking approach provides advanced power automation for the whole electricity grid in terms of detection, measurement and control, among other capabilities [Cabrini *et al.*, 2022a]. The next SG generation is predicted to advance at unprecedented levels, through an ecosystem of cutting-edge Information and Communication Technologies (ICTs) led by the Internet of Things (IoT) and Cloud Computing paradigms. This will transform the power grid into a full automated, intelligent, and widely distributed energy network [Cabrini *et al.*, 2021; Saleem *et al.*, 2019; Hashmi *et al.*, 2021].

In the context of SG, power meters measure the energy consumed by eletroelectronic systems within a power grid. For instance, an electro-mechanical power meter causes problems regarding metering precision owing to variations in ambient temperature, primary voltage, frequency, harmonics signals, and external magnetic fields. Electronic meters have emerged to overcome the problems raised by electro-mechanical meters, as they are based on a set of A/D converters, microcontrollers, and microprocessors that make power measuring more accurate. Currently, power meters have evolved for the smart era¨ in the form of Intelligent Electronic Devices (IEDs) that embed functions capable of sharing power metrics in real-time with remote SCADA systems, which are transmitted across networks in the form of telemetry data [Santos *et al.*, 2022].

The Internet of Smart Grid Things (IoSGT) [Santos *et al.*, 2022] has emerged as a promising ecosystem with a wide range of technologies that are orchestrated to pave the way for new SG applications and services. Technically speaking, the IoSGT cloud-native approach advances across servers that are located at both core (*cloud computing*) and edge (*edge computing*) facilities, thus, establishing an *IoT-to-edge-to-cloud* continuum. Hence, the IoSGT can accelerate SG automation and control by enabling functions to be executed at edge data center facilities that are near IEDs, rather than in cloud computing where the data must be sent to faraway centralized data centers (which are likely to be situated in a different country). In our previous work, Modesto *et al.* [2022], the findings from IoSGT validation outcomes suggested that SG functions running at edge servers proved to have a better performance than in central clouds (among other benefits).

The literature displays a large number of footprints regarding the application of Artificial Intelligence (AI) techniques to SG use cases. In the research of Singh and Gill [2023] and Omitaomu and Niu [2021], for instance, the authors undertook comprehensive studies to show that AI-supported SG is becoming more and more prominent, in the sense that it is able to overcome the limitations of traditional modeling, optimization, and control technologies. In addition, the introduction of AI capabilities into Edge computing facilities ensures highly accurate decision-making at on-site premises, since the high proximity between IEDs and servers, entail computing, data storage, and AI-enforced SG control applications [Ramalho and Neto, 2016]. Moreover, Edge Computing enhances SG by avoiding the need to constantly interact with Cloud server facilities to enable SG data processing and analysis to be carried out, as well as to respond back with the actions needed [Cabrini *et al.*, 2022b].

More recently, devices outside the Edge Computing server premises have been deployed to offload upper-layer computing tasks, also called Extreme Edge. Extreme Edge attempts to speed up processing, as well as to achieve energy sustainability by preventing the exchange of useless data with external servers [Portilla *et al.*, 2019]. The Extreme Edge computing scheme refers to the outer-most layers within a cloud computing architecture, i.e., those nearest either the end users or things under monitoring/control. On the basis of the Extreme Edge use cases, the classical approach provided by the IoT-supported Advanced Metering Infrastructure (AMI) [Kaur, 2021] has attracted particular attention in the area of our research. In AMI-IoT, the smart meters transmit through an Internet connection, (time-to-time), and all the electrical readings are delivered to a known remote utility in the SCADA system which is generally running in a central cloud server. This in turn is responsible for data processing and offering analytical perspectives.

In this research endeavor, we hypothesize that introducing extreme edge capabilities into IEDs offers a prospect of ensuring a higher AMI-IoT use-case performance, by enabling metering, analytics, automated control, and other SG functions to be carried out on-site at IEDs. Thus, it allows a more secure and scalable solution to be found. In this scenario, Edge Computing adds vital support, by provisioning AI training functions which are carried out in the environment nearest to the extreme edge IEDs.

1.1 Problem Statement

As a use case of IoSGT for future SG applications and services, energy consciousness, along with the quality of energy offered to customers, are key factors to achieve sustainability and power efficiency, given the great scarcity of resources and energy, and in compliance with Goals 7, 11, and 13 set out in the guidelines of the 2030 Agenda for Sustainable Development of the United Nations (UN) [Assembly, 2015]. To achieve this, power consumption monitoring must be put into effect in a per eletroelectronic granularity within the same power grid. Energy sensing of off-the-shelf smart meter IEDs focuses on the cumulative records of the total amount of energy consumption (*e.g.*, for monthly billing purposes), regardless of the eletroelectronics granularity within the same power grid. However, identifying eletroelectronics connected inside the same power grid is not a trivial task, since it requires specific classification techniques and methods to provide an accurate identification.

Individual eletroelectronic identification can be obtained by studying and analyzing the power signals injected into the grid. Analytical studies of classic power signals (namely, voltage, current, and power) have been conducted since the 1990s [Hart, 1992]. Our starting-point is based on the hypothesis that the emergence of the Edge Computing facilities at IoSGT has turned Artificial Intelligence (AI) techniques for classic energy power metrics into a cost-effective method for classifying eletroelectronics with a very high degree of accuracy, as shown in Singh and Gill [2023] and Santos *et al.* [2022]. Moreover, most of the smart meter IEDs can sense and share traditional power metrics, by adopting a fully compatible approach without the need for upgrades.

Although this system can assist in detecting and identifying the different types of eletroelectronics connected within the same power grid (*e.g.*, a computer, an air conditioner, or a televisions), the analysis of classic power measures cannot distinguish between devices of the same class (*i.e.*, the different types of televisions). However this shortcoming can be overcome by including more electrical reading features (*e.g.*, power factor, active/reactive/apparent power, and harmonic signals up to the 50th order) in the analysis. Nevertheless, this requires the adoption of new smart meter IED strategies with a capacity to handle these new reading features. Moreover, this kind of IED approach requires to entail power-expanded computing resources (*i.e.*, processing, memory, and storage), in order to become capable to achieve this goal, as well as being more expensive.

To the best of our knowledge, this kind of system lacks a standard solution that is capable of identifying every type of electro-electronic device on an individual basis. A set of ”smart” electro-electronic appliances available in the market (mostly televisions, air conditioners, and refrigerators) entail IoT-based technologies that are intertwined with smart home facilities (*e.g.*, Google Home[1], LG ThinQ[2], and Amazon Alexa Smart Home[3]). However, these kind of smart home solutions are designed to handle interactive functions aimed at customizing aspects of ambience, comfort, and well-being, rather than analyzing power consumption for the purposes of energy efficiency, conscious use, or sustainability.

AI-enabled SG envisions operational processes being enhanced to unprecedented levels, by allowing the identification of electro-electronic devices based on the classic power signals that are provided by the existing smart meter IEDs. In this context, the ability to analyze a large volume of data in a short time enables Machine Learning (ML)-supported mechanisms to adapt their functionality to dynamic changes, as well as to predict the occurrence of future events with a reasonable degree of accuracy. The literature examines a vast number of ML techniques, ranging from conventional types (based on supervised learning) to more recent techniques (based on deep neural networks, unsupervised learning, reinforced learning, and the innovative federated learning technique). The range of ML techniques has led to in-depth studies to assess which method is best-fitted for a highly accurate identification of electro-electronic devices connected to the same power grid (within the scope of this research study).

All the ML training was carried out in the edge cloud to ensure homogeneity in the results obtained, avoid possible errors and/or system incompatibilities with regard to the configuration parameters of each algorithm used, and thus achieve the best scenario that could guarantee the results would be consistently reliable.

1.2 Goals and Research Contributions

The widespread adoption of smart meters has led to a better understanding and management of electricity consumption in homes and businesses. However, there is still a significant challenge over the question of determining which electro-electronic devices should be connected to the same electrical network and thus reduce energy consumption. The identification of individual devices might be a significant step in improving energy efficiency, by making it possible to adopt intelligent consumption practices resulting in the optimization of energy use.

In this paper, we put forward a system to classify electro-mechanical devices connected to the same electricity grid, by means of the IoSGT architecture and machine learning algorithms. Data is obtained from the smart meter, which is capable of collecting information about energy consumption at regular intervals. Our aim is to provide a detailed study of the effects of advanced machine learning techniques on the identification of electro-electronic devices connected to the same electricity grid. To achieve this goal, the IoSGT architecture is employed to establish the continuum of IoT-to-edge-to-cloud and orchestrate the architectural components in Smart Grid. We hope that this study can assist in making significant advances in the current knowledge about the identification of electro-electronic devices in a smart electrical network and in the management of electricity, improving energy efficiency and reducing operational costs, which are critical factors in achieving sustainability. In addition, the main purposes of the research are: (i) to create a *dataset* in which electro-electronic devices are connected to the same power grid, with fundamental sinusoidal signals (voltage, current and power); (ii) to provide a detailed study of the effects of advanced machine learning techniques on identifying electro-electronic devices connected to the same electricity grid; (iii) to establish a bridge between IoT and Smart Grid through the IoSGT architecture, allowing IoT devices to be integrated with intelligent electrical networks; (iv) to make a contribution to enhancing research in Smart Grids, by determining challenges and opportunities related to the identification of electro-electronic devices and suggesting possible solutions

1.3 Organization

This paper is structured as follows. Section 2 provides an analysis of current related work. Section 3, gives a detailed description of the use case and how the experiments were carried out. Section 4 outlines an analysis of the outcome of the experiments. And finally, last but not least, Section 5 wraps up this article with a conclusion and makes suggestions for future work.

2 Related Work

The power and energy measurement instruments available on the market are based on either, devices embedding the electro-electronic systems, or attached to them (*e.g.*, smart sockets). Such instruments make local measurements and deliver telemetry data via the local network to central analyzers (which might be in the home or remote cloud installations). On receiving these telemetry data, the analyzer tool individually identifies the monitored equipment (by recognizing the digital signature or generating the code by means of the embedded/attached artifact), store the information, and even allow the process to be monitored via the web or mobile application. This kind of solution emerges as efficient means of monitoring the consumption that each electro-electronic takes. On the other hand, there are specific artifacts designed to monitor each piece of equipment, which increases the cost according with the density of electro-electronics. Hence, the high costs this entails might make this economically unfeasible, especially with proprietary equipment (which seriously limits the scope for data interoperability).

Among the recent research studies on this subject, it is

[1] http://home.google.com/
[2] http://www.lg.com/us/lg-thinq
[3] http://www.amazon.com/alexa-smart-home

worth highlighting the works of [Quek *et al.*, 2016], which employs methods that combine two machine learning techniques: unsupervised clustering of K-means and techniques of supervised classification of k-nearest neighbors. These models train systems that can effectively determine the low voltage DC electronic load and simultaneously detect if it is in its steady state. [Huang *et al.*, 2011], deals with the question of classification by examining information retrieved from energy signatures and current harmonics. [Hayvaci *et al.*, 2019] classify electronic devices by means of a harmonic radar, where variant signals are transmitted in single-tone time with variable power for the electronic circuits under test. [Hayvaci *et al.*, 2021] explores the non-linear reradiation of electronic circuits under test (ECUT) for classifying electronic devices using harmonic radar, as well as a linear model to relate measurements to unknown parameters representing non-linear ECUT features. [Hayvaci *et al.*, 2021] classifies electronic devices by adopting a Scanning Frequency Harmonic Radar (FSHR) approach.

Some works, such as that of [Huang *et al.*, 2011], explore measurements in the harmonic field. However, this approach restricts the use of many *smart meters* since not all of them have this type of measurement available. As a result, the classic meters for voltage, current, and power measurements, are rendered obsolete in this approach, and this results in the added costs of creating a new infrastructure compatible with this type of measurement.

Energy companies in several cities worldwide have already announced their decision to adopt AI-based solutions in their SG ecosystems for managing their electrical installations. The Chinese company Huawei, for example, reported they had conducted an intelligent inspection system in the city of Shenzhen. The solution generally uses AI algorithms that are designed to process and analyze images and videos created by surveillance cameras (including drones). The purpose of this is to detect physical problems in towers or poles within the SG domain. The company estimates that by combining this intelligent system with human manual judgment, the inspection period can be reduced from the original 20 days to just two hours [Huawey, 2020].

The above literature shows that [Huang *et al.*, 2011], [Hayvaci *et al.*, 2019] and [Hayvaci *et al.*, 2021] explored the harmonic field, and only [Quek *et al.*, 2016] examined energy data through classical signals. The purpose of this work is to use classical measurements, so that it is possible to reuse the smart meters that can be found in the market today.

3 Prototyping and Testing Methodology

This Section outlines the setup of the experimental test environment and describes the methodology used to achieve the central objective - to evaluate how this analytical task affects the way the classification of electrical and electronics systems can be supervised by analyzing the telemetry of research tests and currents in the power grid.

Figure 1 provides an overview of the system, from the capture of the fundamental voltage and current signals, the arrangement of this data and request for data processing, responsibility for training and validation, and the selection of the ML algorithm for the identification of the device.

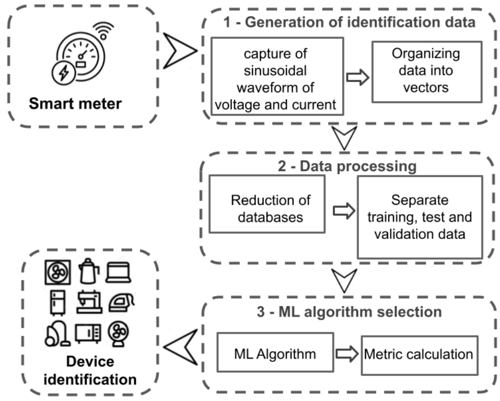

Figure 1. System Overview.

3.1 IoSGT

We extended our IoSGT Testbed [Santos *et al.*, 2022] by including three key layers: Extreme Edge, Edge Cloud Data Center (DC), and Central Cloud DC, Figure 2. The Extreme Edge layer, consisting of IEDs, gathers energy-consumption information at a given period of time. The IED measures voltage, current, active and reactive power and other factors for an embedded low-capacity storage system. An embedded Edge Machine Learning Framework (EMLF), with Classifier and ML Model, relies on an ML Model that is stored in the Edge Cloud to detect electro-electronic devices. The updates of the ML Model are taken from the Edge Cloud DC itself, for each group of IEDs. The sets of IEDs arranged in groups have similar installations - location, version, technical specifications, and consumption profile (residential, commercial, and industrial). The IED transmits data to the Edge Cloud DCs using a well-known Internet of Things (IoT) communication protocol, called Message Queuing Telemetry Transport (MQTT). The deployed IEDs deliver energy data to the chosen Edge Cloud DC instances. The Edge Cloud DC acts as a gateway and manager for the IED groups, by mediating data transmission to ML services atop our proposed IoSGT platform.

The IoSGT takes account of the FIWARE platform architecture, by means of harnessing an open-source ecosystem that is based on Docker containers, thus, enabling greater scalability to merge or fork Edge Cloud DCs. The Extreme Edge interacts with the Edge Cloud DC through the IoT Agents, which are suitable for the resource-constrained approach that the IoT devices yield. The IEDs interact by means of the Ultralight 2.0 protocol. The Context Broker acts as an interface of the Ultralight protocol, which enables the querying of contextual information. For instance, we used the Orion Context Broker as a Publish/Subscribe interface between Ultralight and the Next Generation Service Interfaces (NGSI).

The Connector Framework divides the history of contextual data into persistent databases. In particular, we used Cygnus to automate and manage the data flow for Storage in third-party databases. For instance, we stored real-time data in a MongoDB database to keep the latest readings.

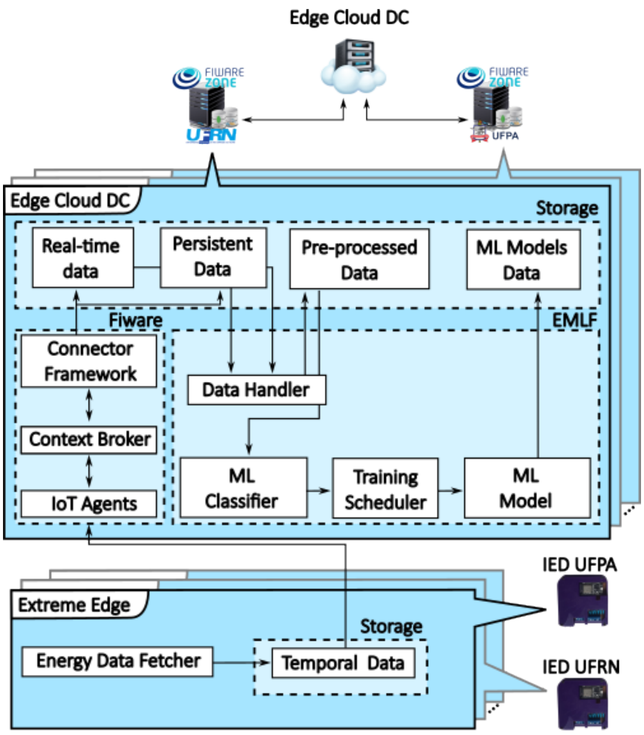

Figure 2. IoSGT Architecture

Moreover, we stored persistent long-term data in a relational database using MySQL, to arrange groups of IEDs. The stored data is used to extract useful information by attaching our proposed EMLF service.

The EMLF service consists of an ML classifier. The persistent data collected from IEDs needs to be formatted to feed the service. For this reason, we designed a Data Handler in charge to carry out the pre-processing, such as cleaning up errors, and removing null values, detecting outliers, transforming data types, and resampling data into different granularity. We stored pre-processed data into our storage system and fed the ML Classifier. Periodically, the classifier updates its models using new collected and pre-processed data with the aid of the Training Scheduler. The training device creates enhanced ML Models for each group of IEDs and saves them in a persistent database for further usage so that, for example, frauds in energy consumption can be detected.

Finally, within the IoT-to-Edge-to-Cloud continuum, the Central Cloud DC layer complements the IoSGT with high processing and storage capacity, which enables it to have a global view of the SG system. For instance, the Central Cloud DC collaborates with the Edge Cloud DCs services by providing the following: powerful analytics, ML models merged with more accurate models and decision-making schemes, personalized IED domains with predictions from the global model, and others. Our prototyping adopts an edge-based approach, and thus, the Central Cloud DC instance is not in the scope of our current work.

3.2 Methodology for Preparatory Dataset

The data used for analysis and training was extracted from a real-world SG system, harnessing a smart meter designed by the group. The methodology employed for the preparatory dataset process took place in the following phases:

1. 3 different devices were added, one at a time and individually, in the electrical network, while its measurements were being made by the smart meter. The devices consisted of a cooler, a heat blower, and a monitor. These devices were chosen because they could be found in the research laboratory and belong to different classes.
2. Preparation of a basic dataset: These datasets are intended to represent the energy consumption pattern that active electronics yield within the electricity grid of a consumer unit. To this end, they include measurements of voltage, current and electrical power over time with samples being captured at 1ms granularity.
3. Preparation of asset datasets: To analyze the patterns in the sinusoidal formats of their voltage and current signals and not just data points, the database was adjusted so that each signal sample had 100 voltage, 100 current, and 100 power attributes

3.3 Dataset handling

The dataset is of paramount importance to allow classification operations that will assist in decision-making. The dataset used in this research includes measurements of voltage, current, and power in a temporal way, where information is picked up at various points of the sine wave.

The purpose of this is to classify electro-electronic devices that coexist in the same electricity grid. In this case, four environments are used as a parameter: the first without load, where no equipment is connected to the electrical grid, and the other three environments are divided between the respective electronic devices: heat blower, monitor, and cooler.

Figure 3 displays the average power values of each device, which allows a quick visual comparison to be made between them. In general, the Figure illustrates the difficulty in differentiating information between devices, which can affect the analysis and interpretation of the results. It should be stressed that the presented average values are only a general representation and that there may be variations between individual measurements.

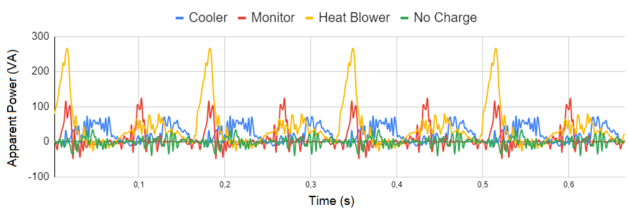

Figure 3. Apparent Power Chart

The use of ML algorithms in this work is necessary to assist the targeting identification, through an analysis of sine waves generated by equipment connected to the same electricity grid. After this, it is possible to quantify each device's energy consumption in a given consumer unit.

For better data visualization and better signal processing, we decided to concatenate 100 temporal instances into just one. In this way, we now had a dataset with 300 attributes. Owing to the environment used for data acquisition, four datasets were created (one for data with no added load, one for monitoring, one for the heat blower, and one for the cooler), each with 87 instances and 300 attributes. After validating these experiments, we intend to scale the density of the electro-electronic devices connected to the same electricity grid , and classify as much equipment as possible in the same consumer unit on the basis of the analysis of electrical signals of voltage, current, and power.

3.3.1 Pre-processing

The data pre-processing phase is one of the most delicate steps, as it involves preparing the data for use in ML algorithms, with the aim of ensuring the availability of information so that the algorithms work correctly and consistently, and achieve the best performances and results. Thus, we seek to carry out standard procedures in accordance with the recommendations of [Sivakumar and Gunasundari, 2017], which are aimed at to cleaning, integrating , transforming and reducing data. The Data Handler in the Edge Machine Learning Framework (EMLF) Internet of Smart Grids for IoT (IoSGT) architecture transforms the data as specified in the pre-processing phase and keeps it in the Pre-processed Data storage facility.

Removing unnecessary information is the first step taken with datasets. It is common for a database to contain empty fields and incoherent values, which means it is necessary to process these data. The alternative way of overcoming this problem is to eliminate these instances. After following this procedure, we obtained a dataset with 348 instances and 300 numeric attributes.

Subsequently, data integration was carried out to prevent the occurrence of errors, inconsistencies, and redundancies in the attributes of the dataset. Soon after, these data were transformed and saved in the Pre-processed Data Storage system in their own formats so that they could be used by the models. Thus, the data normalization was carried out, and the creation of 3 new datasets which used some of the Weka software filters [WEKA, 2022]. A different filter was used for each of the 3 datasets; in the reduced dataset 1 (R1), the ”Remove percentage” filter was used. This was configured to remove 50% of the instances, and led the reduced dataset 1 to have 300 attributes and 174 instances.

The reduced dataset 2 (R2) made use of the ”Principal Components” filter, which significantly reduced the number of attributes, with 348 instances and only 63 attributes. In reduced dataset 3 (R3), the filter used was ”Remove Folds”, which reduced the number of instances, leaving it with 35 instances and 300 attributes. After creating the three datasets from the original, we obtained a total of 4 datasets (which are arranged in Table 1), to carry out the training of the ML algorithms.

Table 1. Datasets used.

Dataset	Nº instances	Nº attributes
Original	348	300
R1	174	300
R2	348	63
R3	35	300

3.4 ML models

Supervised learning is one of the most important tasks in the ML process, as it consists of i) learning functions that map results through inputs and outputs, ii) understanding functions from already classified training data, and iii) is formed of a set of training examples. In supervised learning, each model consists of a combination of input and output. At the input, there is usually a vector of values; at the output, there is the desired result for that specific value or supervision signal.

The supervised learning algorithm analyzes the training data and reaches a conclusion, which it uses to map out new examples. The algorithm can correctly determine class labels for unknown instances in an ideal environment. This process will require the learning algorithm to have a suitable generalization of the training data in unknown situations. In the following section, we describe the main supervised learning techniques that are used for the correct classification of individualized electronics.

3.4.1 K-Nearest Neighbors (K-NN)

K-Nearest Neighbors (K-NN) [Altman, 1992] is an unparameterized and ”lazy” (*lazy*) supervised ML algorithm, which assesses the distance among problem instances and a possible calculation object. Given an unlabeled instance, the KNN algorithm selects the k nearest neighbors of that instance and adopts a voting-based strategy capable of choosing the majority-voting label. The algorithm receives k-nearest neighbors as input, and calculates instance similarity by adopting the Euclidean distance. Thus, the KNN-supervised ML algorithm distinguishes itself by both simplicity and flexibility main advantages.

3.4.2 Decision Trees (J48)

The Decision Trees algorithm (J48) [Quinlan, 1986] is a widely used ML technique that proceedings data examination in a categorically and continuously manner. The J48 algorithm follows a top-down approach that is based on a recursive divide and conquer strategy. On the basis of an attribute selected to split on at the root node, a branch is created according with each possible attribute value. Then, J48 splits instances into subsets, one of which associated to each branch extending from the root node. The highest classification accuracy depends on using the best attribute to split on set with the greatest information. The J48 algorithm resolution feature is based around decision trees, as such, it ends up suffering from a sort of problems that using the decision tree approach yields, such as empty branches, insignificant branches, and overfitting. The use of pruning methods can be a solution to overcome the overfitting problem, by removing the nodes which results in partitions with few instances.

3.4.3 Naive Bayes (NB)

Naive Bayes (NB) [Hand and Yu, 2001] is an ML algorithm that uses probabilistic models to describe datasets. Although Naive Bayes (NB) is a widely used algorithm in data classification, its simplicity and built-in assumptions can lead to limitations in certain cases, such as the following: a) situations where the data displays a high correlation between its features, b) there are missing values, c) the feature distributions are complex, d) there is an imbalance in the classes , or e) there are irrelevant features present. Since it is regarded as ”naive”, it assumes that the features of the data are independent of each other, which is not always the case [McCallum and Nigam, 1998].

Furthermore, NB assumes that the attributes are equally important and statistically independent. These requirements may often be violated, where too many redundant attributes will not affect the algorithm's performance. The NB formula is derived from Bayes' Theorem and is used to calculate the probability of an event belonging to a specific class, based on the characteristics or attributes of the data. The formula is written as: $P(C|X) = \frac{P(X|C)P(C)}{P(X)}$ where: P(C|X) is the probability of the event belonging to class C, given a set of characteristics X. P(X|C) is the probability of the characteristics X occurring in data belonging to class C. P(C) is the prior probability of an event belonging to class C. P(X) is the probability of the characteristics X occurring in all the data. When the data are unsuitable for the model, calculating probability through the Gaussian distribution can cause a serious problem.

3.4.4 Support-vector machine (SVM)

Support Vector Machine (SVM) [Cortes and Vapnik, 1995] is an optimization-based supervised ML method that aims at finding a hyperplane along an N-dimensional space (where *N* is the number of features). Hyperplanes stand to decision boundaries that assist in the task of distinctly classifying classes of data points. Algorithmically speaking, each instance is projected with a feature vector within the hyperdimensional space. Afterwards, SVM proceeds the data classification through finding an optimal separation hyperplane. Although the use of linear functions as parameters requires to reduce computational costs, SVM is highly suitable for scenarios with high-dimensional data and problems linearly separable. Moreover, the high sensitivity of SVM to variations makes very difficult to find the best parameter settings.

3.4.5 Multi Layer Perceptron (MLP)

Multi Layer Perceptron (MLP) [Meyer-Baese and Schmid, 2014] is a class of fully connected Artificial Neural Networks (ANN) and is characterized by having one or more intermediate layers, which makes it possible to handle complex and non-linear systems [Haykin, 2001]. The most widely used algorithm in MLPs *backpropagation*, which injects a signal at the network's input that is propagated forward to the output layer. Then the error can be calculated, since it corresponds to the difference between the real outcome and the result produced by the network. A signal sent to the previous layers calculates the new synaptic weights. The algorithm is repeated until the error reaches a pre-defined value or a maximum number of repetitions [Haykin, 2001].

4 Analysis on the Experiment Outcomes

The goal of this session is to assess the performance that the main ML methods and techniques takes in our edge-cloud infrastructure. We compared the results obtained in the experiments carried out with implementations of the k-NN, J48, SVM, Naive Bayes, and MLP algorithms. This was achieved by making use of the four datasets (Original, Reduced 1, Reduced 2, Reduced 3), divided into 80% for training and 20% for testing, with a kfold of 10. Furthermore, the confusion matrix was extracted for each case, where the actual data was compared with the predicted data in the separate test set for each algorithm. The results and corresponding analysis are described in the following subsections:

4.1 K-NN set of Experiments

In the K-NN algorithm, the impact of the K parameter was verified. The Jupyter tool was used to do the tests, which involved carrying out experiments with each dataset, and several different scenarios were created for this. In each dataset, there was a training session and 10-fold cross-validation was used for different values of K(k=1, k=3, k=5, and k=7). Jupyter is the most used system for interactive literary programming [Shen, 2014]. The best results of each dataset used can be seen in the Table 2.

The literary programming paradigm is designed to assist in the communication of programs by alternating formatted natural language text, executable code extracts, and the results of calculations. Jupyter's interactivity allows this paradigm to be used in real time for data analysis, particularly the Python language. After carrying out the tests with the K-NN algorithm, it was found that the two reduced datasets achieved the best performance, since it obtained a 100% degree of accuracy, according to the Table 2 shows the importance of scaling the values to obtain a better result. The confusion matrix of the dataset that obtained the best result can be seen in Figure 4, where the assertiveness¨ of the data classification should be noted.

Table 2. Results using K-NN.

K-NN			
Dataset	Amount of K	Time	Accuracy
Original	3	0.08 seconds	99.71%
R1	1	0.01 seconds	97.12%
R2	1	0.01 seconds	100%
R3	3	0.01 seconds	94.28%

4.2 Decision Tree set of Experiments

In the J48 algorithm, the effects of creating or not pruning the tree were determined. For this reason, together with the Jupyter tool, we carried out experiments for each of the 4 datasets (original, reduced 1, reduced 2 and reduced 3). For each one we configured the training for 10-fold cross-validation and ran the tests both with and without pruning; the results can be seen in Table 3.

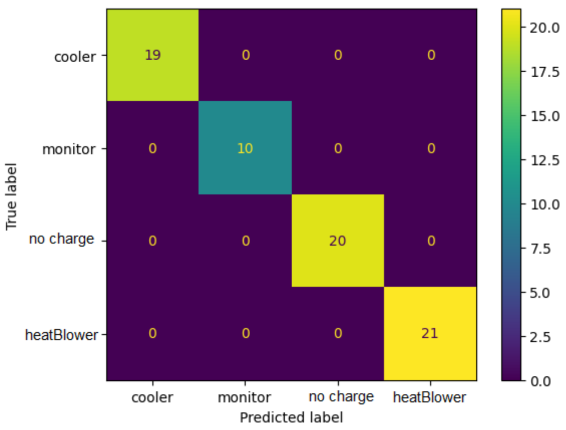

Figure 4. K-NN algorithm confusion matrix.

Table 3. Results using decision tree.

J48				
Dataset	Time	Accuracy	Tree size	Nº leaves
Original	0.05 s.	67.14%	8	37
R1	0.05 s.	68.57%	8	22
R2	0.06 s.	84.28%	8	23
R3	0.05 s.	57.14%	4	6

By analyzing the results present in the Table 3 it can be seen that the DT algorithm obtained the best performance with the reduced dataset 2, configured without pruning, and obtained a degree of accuracy of 84.28%. In addition to these data, it can be seen that the tree is 8 nodes high with a total of 23 leaf nodes. On the basis of all these tests, it can be stated that there was no overfitting, either before or after the pruning. The confusion matrix of the reduced dataset 2 can be seen in Figure 5, where it shows the false positives that were taken into account during the data classification.

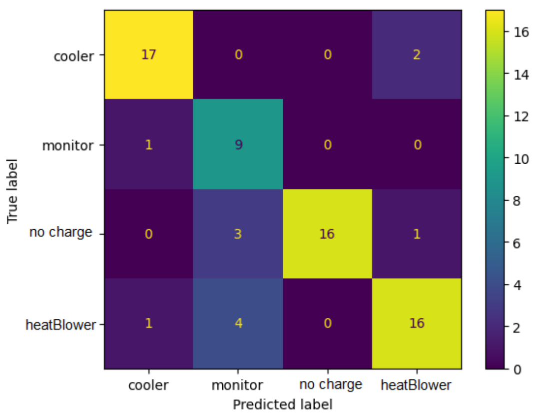

Figure 5. Decision tree confusion matrix.

4.3 Naive Bayes set of Experiments

In the NB algorithm, the impact of the two assumptions was verified, by means of the Jupyter tool. Experiments were carried out with each dataset, and, different scenarios were created for this and following this, training was carried out with each dataset. The results of the experiment using the NB algorithm can be viewed in Table 4.

Table 4. Results using Naive Bayes.

Naive Bayes		
Dataset	Time	Accuracy
Original	0.06 s.	84.77%
R1	0.06 s.	87.93%
R2	0.06 s.	94.54%
R3	0.06 s.	91.42%

By analyzing the results present in the Table 4, it can be seen that the dataset that obtained the best result with the NB algorithm was the reduced 2, with an accuracy of 94.54%. In the case of all the tests , there was a normal distribution, as both data are related to each other. The confusion matrix of the NB algorithm using the reduced dataset 2 can be seen in Figure 6. By analyzing the confusion matrix, it can be seen that the algorithm obtained a good rate of assertiveness in classifying the data, but a number of false positives were detected.

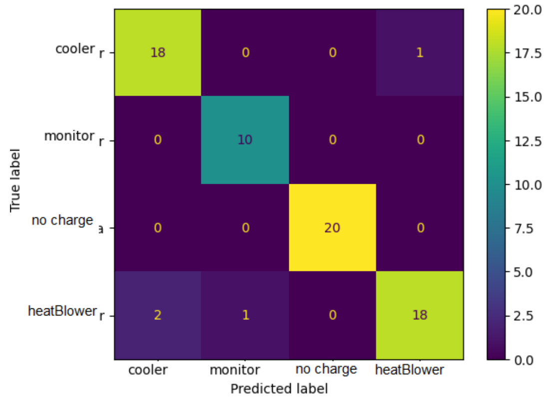

Figure 6. Confusion matrix of the Naive Bayes algorithm.

4.4 SVM set of Experiments

In the SVM algorithm, the impact of the Kernel type on the performance of support vector machines (SVM) was determined. The Jupyter tool was used to carry out the tests, which involved experiments with each dataset, and several different scenarios were created for this. In each dataset, training was carried out by means of different kernels (exponential, Gaussian, and linear), as well as different values for the Cost parameter. Among the three kernels used, exponential had the best performance, and its results can be seen in Table 5.

Table 5. Results using SVM.

SVM				
Dataset	Kernel	Cost	Time	Accuracy
Original	Exponential	2	0.14 s.	98.85%
R1	Exponential	1	0.08 s.	97.12%
R2	Exponential	4	0.08 s.	99.71%
R3	Linear	1	0.03 s.	94.28%

After carrying out the experiments using the SVM algorithm, it was found that the best results were obtained in the reduced dataset 2, using the Exponential Kernel, in which the

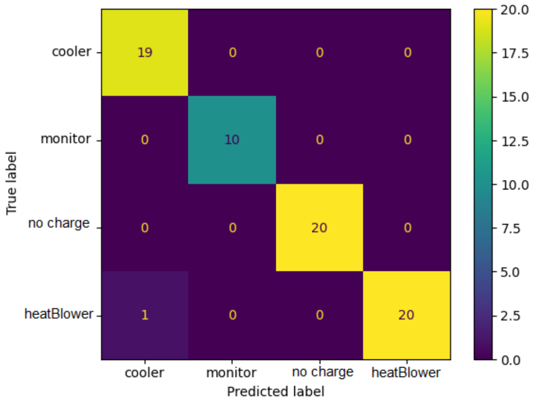

Figure 7. Confusion matrix of the SVM algorithm.

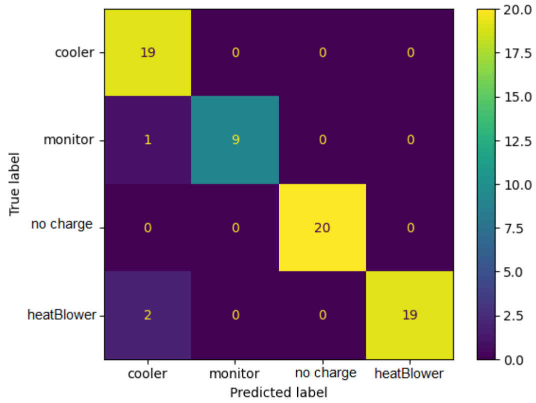

Figure 8. Confusion matrix of the MLP algorithm.

Cost parameter value was equal to 4. It should be noted here that the cost parameter is quite a significant requirement to obtain a good degree of accuracy. Figure 7 shows the confusion matrix of the classification of the data using the reduced dataset 2, since it had the best performance. It can be seen that there is an assertiveness¨ of 99.71% in the classifications, just missing a single rating.

4.5 MLP set of Experiments

In the MLP algorithm, the impact of the number of neurons (NN) on the hidden layer was determined . The Jupyter tool was used to carry out the tests, which included experiments with each dataset; several different scenarios were created for this. In each database, there were training sessions using different iterative values (100, 1.000 and 10.000), as well as different values of neurons in the hidden layer, alternating between 4, 8, and 12. In addition to these settings, the variation of the learning rate (LR) was established and ranged between 0.1, 0.01 and 0.001. the best results obtained for each dataset can be viewed in Table 6.

Table 6. Results using MLP.

0.8

Multilayer Perceptron (MLP)					
Datasets	Iterations	NN	LR	Time	Accuracy
Original	1000	12	0.001	2 ms	85.91%
R1	1000	12	0.01	1 ms	84.48%
R2	100	8	0.01	1 ms	95.71%
R3	10000	4	0.01	1 ms	88.57%

After applying the MLP algorithm, it was found that the best result was obtained using the reduced dataset 2, with 8 neurons in the hidden layer and a LR of 0.01. The resulting degree of accuracy for the test was 95.71%. Figure 8 shows the confusion matrix of the best result obtained regarding the classification of data using the reduced dataset 2, and also reveals that the algorithm only missed three classifications.

4.6 Outcome Analysis Wrap Up

On the basis of the results obtained from the performance tests carried out with each algorithm, a comparative an detailed analysis can be conducted. This shows the algorithm that raised the best performance regarding the classification of electrical and electronic devices. Table 7, shows the best result obtained by each algorithm, and it can be seen that k-NN had the best performance, with an accuracy rate of 100%.

Table 7. Best algorithm results

Algorithm	k-NN	J48	NB	SVM	MLP
Dataset	R2	R2	R2	R2	R2
Nº K	1	-	-	-	-
Pruning	-	w.p.	-	-	-
Kernel	-	-	-	Expo	-
Cost	-	-	-	4	-
Nº Neurons	-	-	-	-	4
LR	-	-	-	-	0.001
Time(s)	0.06	0.06	0.06	0.08	0.05
Accuracy(%)	100	84.28	94.5	99.7	95.7

Account should be taken of the conditions under which each algorithm was used when drawing conclusions about its performance, and our conclusions should not simply be based on the best result obtained by each one. In light of this, Table 8 highlights the worst performance of each algorithm, since this allows us to obtain a more complete view of their performance. When analyzing this Table, it can be seen that although the J48 algorithm performed well overall, it also had the worst performance among all the algorithms, with an accuracy rate of only 14.28%. The MLP algorithm obtained the second worst rate, with 49.42%. However, if we compare the best performance with the worst, the k-NN algorithm stands out with an accuracy of 88.57%.

Table 8. Comparison of the worst results

Algorithm	k-NN	J48	NB	SVM	MLP
Dataset	R3	R3	Orig.	R3	R1
Nº K	7	-	-	-	-
Pruning	-	w.p.	-	-	-
Kernel	-	-	-	Gauss.	-
Cost	-	-	-	1	-
Nº Neurons	-	-	-	-	4
LR	-	-	-	-	0.01
Time(s)	0.26	0.05	0.06	0.05	0.02
Accuracy(%)	88.57	14.28	84.77	51.42	49.42

So it can be concluded and stated that the k-NN algorithm had the best performance. Thus we decided to create Table 9, with the average of all the results of each algorithm, so that it could be determined which algorithm had the best performance in terms of the average rate of accuracy.

Table 9. Summary of results.

0.8

Summary of results		
Algorithm	Time	Accuracy
k-NN	0.06 s	96.61%
J48	0.05 s	62.58%
Naive Bayes	0.06 s	89.67%
SVM	0.08 s	92.14%
MLP	0.07 s	76.70%

By analyzing what is shown in Table 9, we can state that the k-NN algorithm obtained an average accuracy of 96.61%, which confirms the hypothesis that the k-NN algorithm is the best algorithm to perform the classification of electronic devices by means of classical energy data (voltage, current and power).

5 Conclusion and Recommendations for Future Work

In this paper, we address the challenges of identifying and classifying different types of electro-electronic devices within the same electrical grid, a significant step in improving energy efficiency and good consumption practices. To this end, we harnessed the IoSGT architecture to orchestrate the architectural components across a Smart Grid IoT-to-edge-to-cloud continuum for the task of collecting regular measures provided by smart meters. Moreover, we carried out a detailed study about the effects of the most significant machine learning techniques to identify electro-electronic devices connected to the same electricity grid.

In order to assess how the ML techniques perform in classifying electro-electronics individually through fundamental sinusoidal electric signals (voltage, current, and power) processing, we set up an experimental test environment. For this, we created a *dataset* providing historical data of electro-electronic devices inside the same power grid, with fundamental sinusoidal signals (voltage, current, and power) obtained from real smart meters. The outcomes suggested that, among all the ML algorithms applied, the K-NN performed better by obtaining an accuracy rate of 100% for almost all the datasets, while achieving, at the same time, lower time taken to carry out classification functions.

On the basis of the satisfactory outcomes obtained in our tests, we aim to explore the KNN-based classification strategy in greater depth, in future work. In light of this, we intend to devise new parameters and/or attributes, such as harmonic parameters, which will raise the project to a new level by enabling simultaneous analysis of high-dense electro-electronics at the extreme edge (i.e., directly in the SM). Finally, but not least, we aim to advance beyond device detection by inspecting different consumption patterns and the importance of Electric Power Quality (QEE).

Acknowledgment

This study was financed by the R&D project entitled "*Sistema IoT-Cloud de Medição Centralizada de Energia Voltado a Rede CEA - 001/2021*".

Declarations

Authors' Contributions

All authors contributed to the writing of this article, read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Data can be made available upon request.

References

Alonso, A. M., Martín, E., Mateo, A., Nogales, F. J., Ruiz, C., and Veiga, A. (2022). Clustering electricity consumers: Challenges and applications for operating smart grids. *IEEE Power and Energy Magazine*, 20(3):54–63. DOI: 10.1109/MPE.2022.3150826.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185. DOI: 10.1080/00031305.1992.10475879.

Assembly, G. (2015). Transforming our world: the 2030 agenda for sustainable development. Technical report, United Nations General Assembly. Available at: `https://www.unep.org/resources/report/transforming-our-world-2030-agenda-sustainable-development`.

Barja-Martinez, S., Aragüés-Peñalba, M., Munné-Collado, Í., Lloret-Gallego, P., Bullich-Massagué, E., and Villafafila-Robles, R. (2021). Artificial intelligence techniques for enabling big data services in distribution networks: A review. *Renewable and Sustainable Energy Reviews*, 150:111459.

Cabrini, F. H., de Castro Barros Filho, A., Maciel, D. B., Valiante Filho, F., Neto, A. J. V., and Kofuji, S. T. (2022a). Helix multi-layered: a context broker federation for an efficient cloud-to-things continuum. *Annals of Telecommunications*. DOI: 10.1007/s12243-022-00920-z.

Cabrini, F. H., de Castro Barros Filho, A., Maciel, D. B., Valiante Filho, F., Neto, A. J. V., and Kofuji, S. T. (2022b). Helix multi-layered: a context broker federation for an efficient cloud-to-things continuum. *Annals of Telecommunications*, 77(11):867–879. DOI: 10.1007/s12243-022-00920-z.

Cabrini, F. H., Valiante Filho, F., Rito, P., Barros Filho, A., Sargento, S., Venâncio Neto, A., and Kofuji, S. T. (2021).

Enabling the industrial internet of things to cloud continuum in a real city environment. *Sensors*, 21(22). DOI: 10.3390/s21227707.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297. DOI: 10.1007/BF00994018.

Gopstein, A., Nguyen, C., O'Fallon, C., Hastings, N., and Wollman, D. (2021). Nist framework and roadmap for smart grid interoperability standards, release 4.0, special publication (nist sp). Technical report, National Institute of Standards and Technology (NIST). [online]: Accessed on April 1, 2022.

Hajam, S. S. and Sofi, S. A. (2021). Iot-fog architectures in smart city applications: A survey. *China Communications*, 18(11):117–140. DOI: 10.23919/JCC.2021.11.009.

Hand, D. J. and Yu, K. (2001). Idiot's bayes: Not so stupid after all? *International Statistical Review / Revue Internationale de Statistique*, 69(3):385–398. DOI: 10.1111/j.1751-5823.2001.tb00465.x.

Hart, G. W. (1992). Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891. DOI: 10.1109/5.192069.

Hashmi, S. A., Ali, C. F., and Zafar, S. (2021). Internet of things and cloud computing-based energy management system for demand side management in smart grid. *International Journal of Energy Research*, 45(1):1007–1022. DOI: 10.1002/er.6141.

Haykin, S. (2001). Redes neurais artificiais: Princípios e práticas. 2ª edição. Book.

Hayvaci, H. T., Ilbegi, H., and Yetik, I. S. (2019). Classification of electronic devices with power-swept signals using harmonic radar. *IEEE transactions on aerospace and electronic systems*, 56(3):2292–2301. DOI: 10.1109/TAES.2019.2949326.

Hayvaci, H. T., Shahi, M., Ilbegi, H., and Yetik, I. S. (2021). A linear model for classification of electronic devices using harmonic radar. *IEEE Transactions on Aerospace and Electronic Systems*, 57(6):3614–3622. DOI: 10.1109/TAES.2021.3079572.

Huang, S.-J., Hsieh, C.-T., Kuo, L.-C., Lin, C.-W., Chang, C.-W., and Fang, S.-A. (2011). Classification of home appliance electricity consumption using power signature and harmonic features. In *2011 IEEE Ninth International Conference on Power Electronics and Drive Systems*, pages 596–599. IEEE. DOI: 10.1109/PEDS.2011.6147311.

Huawey (2020). Digital engine empowers smart grids - huawei enterprise. Available at:`https://e.huawei.com/ma/news/industries/2020/digital-engine-empowers-smart-grids` Accessed on January 12, 2023.

Kaur, I. (2021). Chapter 29 - metering architecture of smart grid. In Azar, A. T. and Kamal, N. A., editors, *Design, Analysis, and Applications of Renewable Energy Systems*, Advances in Nonlinear Dynamics and Chaos (ANDC), pages 687–704. Academic Press. DOI: 10.1016/B978-0-12-824555-2.00030-7.

Keith Stouffer, Victoria Pillitteri, S. L. M. A. and Hahn, A. (2015). Nist special publication 800-82 revision 2: Guide to industrial control systems (ics) security. Technical report, National Institute of Standards and Technology (NIST).

McCallum, A. and Nigam, K. (1998). Naive bayes: The unreasonable effectiveness of probability estimation. In *AAAI-98 Workshop on Learning for Text Categorization*. Available at: `http://www.kamalnigam.com/papers/multinomial-aaaiws98.pdf`.

Meyer-Baese, A. and Schmid, V. (2014). Chapter 7 - foundations of neural networks. In Meyer-Baese, A. and Schmid, V., editors, *Pattern Recognition and Signal Analysis in Medical Imaging (Second Edition)*, pages 197–243. Academic Press, Oxford, second edition edition. DOI: 10.1016/B978-0-12-409545-8.00007-8.

Modesto, W., Bastos, L., Venâncio Neto, A., Rosário, D., and Cerqueira, E. (2022). Towards automating the integration of legacy ieds into edge-supported internet of smart grid things. *Journal of Internet Services and Applications*, 12(1):33–46. DOI: 10.5753/jisa.2022.2374.

Omitaomu, O. A. and Niu, H. (2021). Artificial intelligence techniques in smart grid: A survey. *Smart Cities*, 4(2):548–568. DOI: 10.3390/smartcities4020029.

Portilla, J., Mujica, G., Lee, J.-S., and Riesgo, T. (2019). The extreme edge at the bottom of the internet of things: A review. *IEEE Sensors Journal*, PP:1–1. DOI: 10.1109/JSEN.2019.2891911.

Quek, Y., Woo, W. L., and Logenthiran, T. (2016). Dc equipment identification using k-means clustering and knn classification techniques. In *2016 IEEE Region 10 Conference (TENCON)*, pages 777–780. IEEE. DOI: 10.1109/TENCON.2016.7848109.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106. DOI: 10.1007/BF00116251.

Ramalho, F. and Neto, A. (2016). Virtualization at the network edge: A performance comparison. In *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6. DOI: 10.1109/WoWMoM.2016.7523584.

Saleem, Y., Crespi, N., Rehmani, M. H., and Copeland, R. (2019). Internet of things-aided smart grid: technologies, architectures, applications, prototypes, and future research directions. *IEEE Access*, 7:62962–63003. DOI: 10.1109/ACCESS.2019.2913984.

Santos, H., Eugênio, P., Marques, L., Oliveira, H., Rosário, D., Nogueira, E., Neto, A., and Cerqueira, E. (2022). Internet of smart grid things (iosgt): Prototyping a real cloud-edge testbed. In *Proceedings of the 14th Brazilian Symposium on Ubiquitous and Pervasive Computing (SBCUP)*, pages 111–120, Porto Alegre, RS, Brazil. SBC. DOI: 10.5753/sbcup.2022.223205.

Shen, H. (2014). Interactive notebooks: Sharing the code. *Nature*, 515(7525):151–152. DOI: 10.1038/515151ax.

Singh, R. and Gill, S. S. (2023). Edge ai: A survey. *Internet of Things and Cyber-Physical Systems*, 3:71–92. DOI: 10.1016/j.iotcps.2023.02.004.

Sivakumar, A. and Gunasundari, R. (2017). A survey on data preprocessing techniques for bioinformatics and web usage mining. In *International Journal of Pure and Applied Mathematics*, pages 785–794. Available at:`https://acadpubl.eu/jsi/2017-`

117-20-22/articles/20/68.pdf.

Wang, Y., Chen, Q., Hong, T., and Kang, C. (2019). Review of smart meter data analytics: Applications, methodologies, and challenges. *IEEE Transactions on Smart Grid*, 10(3):3125–3148. DOI: 10.1109/TSG.2018.2818167.

WEKA (2022). Weka 3: Machine learning software in java. Available at: https://www.cs.waikato.ac.nz/ml/weka/.