Cihan University-Erbil Scientific Journal (CUESJ)

Review Article

Detecting Denial of Service Attacks in Internet of Things Using Software-Defined Networking and Ensemble Learning

Shima Rashidi[1], Adil H. Mohammed[2], Yusra. A. Salih[3]

[1]Department of Computer Science, College of Science and Technology, University of Human Development, Sulaymaniyah, Kurdistan Region, Iraq, [2]Department of Communication and Computer Engineering, Faculty of Engineering, Cihan University-Erbil, Kurdistan Region, Iraq, [3]Department of Database Technology, College of Informatics, Sulaimani Polytechnic University, Sulaymaniyah, Kurdistan Region, Iraq

ABSTRACT

The internet of things (IoT) is a novel approach to automate connections between smart devices without involving humans. The utilization of this structure is growing, and its application range is continually expanding. We confront additional issues as the usage of these networks grows, such as the presence of attackers and combating their attacks. These networks' performance may be improved, and their development can be accelerated, with new solutions to these difficulties. A new method for improving IoT security is proposed in this research, which is based on software-based network and collaborative learning. The suggested solution divides the network domain into numerous subdomains, each with its own controller for exchanging security rules with other subdomains. All of a subnet's node traffic are routed through the subnet's control node in this topology. As a result, each controller node employs an integrated learning model to continually evaluate network traffic data and detect assaults. This learning model incorporates an artificial neural network, a decision tree, and a New Biz model that uses statistical information gathered from each data stream to identify the likely existence of assaults. NSL-KDD database data were utilized to assess the proposed method's performance, and its accuracy in identifying denial of service attacks was compared to earlier approaches.

Keywords: Denial of service attacks, intrusion detection system, cumulative learning, software-based network, internet of things

INTRODUCTION

The internet of things (IoT) is a notion that was first presented in the late 20th century and refers to the intelligent interaction between machines and humans over a large communication platform such as the internet. All smart gadgets in this communication system have a unique identity and the capacity to interact with other devices. Each device that can be connected to the structure of the IoT is called an object, and the result of the integration of all objects in a global information network will form the IoT. In general, the interactions of objects in this communication structure are beyond the traditional machine-to-machine communication and can lead to huge changes in areas such as transportation, industry, treatment network, and so on. Despite these benefits, several of the network's fundamental properties, such as its breadth, variety of objects, and usage of numerous protocols, have posed management issues.[1] A big network, in most cases, cannot function properly without some form of organization. This is due to issues such as user authentication and network architectural scalability. Some of the existing difficulties can be overcome in a cost-effective manner using software defined networks (SDN).[2] A software-centric network is a novel communication network design that has been proposed to achieve aims such as more efficient network dynamics confrontation, improved network flexibility, and improved network manageability. The control layer is separated from the data layer in this design, which is made up of two primary components: Forwarding elements and SDN controllers. Delivery equipment's role is to exchange packets through proprietary hardware or software. The controller, on the other hand, is a piece of software that operates on a hardware platform and is designed to carry out specific functions. To authenticate users on the network, the suggested solution

Corresponding Author:
Adil H. Mohammed,
Department of Communication and Computer Engineering, Faculty of Engineering, Cihan University-Erbil, Kurdistan Region, Iraq.
E-mail: adil.mohammed@cihanuniversity.edu.iq

Received: June 17, 2022
Accepted: July 29, 2022
Published: August 20, 2022

DOI: 10.24086/cuesj.v6n2y2022.pp49-56

employs an SDN-based architecture. It separates the network's domain into SDN subdomains, with each subdomain in charge of authenticating users in its own domain. On the other side, the IoT ecosystem gives greater opportunity for attackers to conduct threats such as denial of service (DoS) assaults due to the exponential growth in the number of connected devices and apps. Application layer DoS attacks are a type of malicious attack that targets the OSI model's top layer (where common internet requests such as HTTP GET and HTTP POST occur). Identifying the origins of attacks and security risks are another topic to consider while constructing the proposed model in this context. To this purpose, the controller nodes in each SDN subnet manage the network traffic pattern and identify the existence of probable attacks on their subnet using a model based on ensemble learning techniques.[3] Several learning models (such as decision tree, neural network, and New Biz) are utilized in the proposed cumulative learning model to identify the origins of assaults, and the final diagnosis is determined by voting among the output findings of the learning models. The following is the order in which this article will be continued: In the second section, we'll look at works that are linked to the research topic. The proposed method is described in detail in the third section, and the results of evaluating its performance in a simulated environment are discussed in the fourth section. Finally, in the research's fifth section, the findings are summarized and recommendations for further research are provided.

RELATED STUDIES

Perrone *et al*.[4] examined the message queuing telemetry transport (MQTT) protocol's security and defined the different security requirements for establishing the IoT and safeguarding devices from application layer attacks using the MQTT protocol. Andy *et al*.[5] looked at certain IoT attack scenarios and assessed IoT security to mitigate these attacks in another study. The authors' study focuses on IoT messaging protocol security and attack scenarios against open authentication servers; they've described it. Although the viability of such attacks is currently debatable, the open authentication functionality is deactivated in most deployments of message servers in industrial contexts to decrease the danger of unauthorized access and associated assaults.[6] Firdous *et al*.[7] investigated a model of SYN flood denial attack and its influence on message mediators to better understand security vulnerabilities in IoT application layer protocols. When inserting fuzzy data between the user and the server, their suggested technique examines the behavior of applications at the application layer. They employed a fuzzy proxy mechanism combined with a non-standard closed variable header data pattern to evaluate both the broker and the user's behavior when presented with unexpected data. Experiments suggest that this method may uncover application layer vulnerabilities in messaging intermediates to some extent.[8] Cipla has released F-Secure MQTT-FUZZ, a comparable tool that has been designed to assess application layer vulnerabilities in the MQTT protocol for commercial reasons.[9] Moustafa *et al*.[10] suggested a technique for identifying IoT-based threats that rely on characteristics derived from TCP protocol analysis. A set of attributes characterizing the communication protocol between users is utilized as input to learning models for this purpose, and the sort of attack is detected by categorizing these features. The fundamental difficulty with this study is that it uses restricted information as input to learning models to detect assault types. Accurate attack detection at the IoT application layer necessitates access to a large amount of data. Syed *et al*.[11] suggested a machine learning-based method to identify application-layer assaults on the IoT to overcome this problem. To determine the sort of attack, this method additionally leverages statistical information from the exchanged packets, as well as aspects of the communication protocol between the objects. The effectiveness of artificial neural networks and decision trees in accurately diagnosing the types of assaults has been investigated in this study, and the results reveal that the decision tree is superior in correctly diagnosing the types of attacks. Kharkongor *et al*.[12] suggested an IoT routing protocol that takes into account the energy consumption of heterogeneous network devices. This solution proposes an SDN controller that serves as a management center for network security and preventing hostile nodes from gaining access to the network. In this paper, first, routing algorithms are classified, then the proposed method for secure routing in the IoT is presented. The routing algorithm proposed in this paper consists of six steps, which are: Registration of nodes by the controller; monitor the entire network by the controller; receive neighbor information by each node in the network; calculate the remaining energy of neighbors; select and send data based on the remaining energy of the nodes to the neighbor; and block malicious nodes and prevent them from reaccessing the network using the controller. A machine learning-based approach for identifying DoS assaults on grid intelligent networks was proposed by Zhe *et al*.[13] Pre-processing procedures, feature extraction, and classification are all part of their suggested strategy. Principal component analysis was utilized to minimize the feature dimensions in the feature extraction stage, and a support vector machine (SVM) was employed to identify DoS assaults in the classification step. The simulation results suggest that SVM outperforms categorization methods such as decision trees and New Bay. In[14] Dong and Sarm suggested two ways for identifying software-based DoS attacks the first one is adopts the degree of DDoS attack to identify the DDoS attack , the second on is improved K-Nearest Neighbors (KNN) algorithm based on Machine Learning (ML) to discover the DDoS attack.

SUGGESTED METHODS

In this part, we'll go through the suggested software-based network and aggregate learning approach for detecting service denial threats in the IoT framework in detail. To provide a safe communication platform between network objects, the suggested solution employs a software-centric network. The network topology is separated into subnets in this situation. An SDN controller node is assigned the responsibility of authentication and communication management of the members of each subnet in this topology. In addition to this communication structure, network traffic is monitored using an integrated learning model based on neural networks, decision trees, and the Niobiz model. As a result, each controller node on its subnet uses this learning model to detect assaults and security concerns. It is required to first define some of the terms that will be used in the following sections; let's pay.

First Definition: An active node is a wireless node that is connected to one of the network's active nodes. Otherwise, we refer to it as an inactive node.

Definition 2: Subnet: A portion of the imagined global network that is administered centrally by an SDN controller node, and whose members can connect with other subnets through the controller node's administration and rules. Network nodes in wireless networks have varying communication qualities due to different manufacturing processes for radio equipment.

As a result, the network that is assumed is heterogeneous. Each software-driven network controller node is equipped with a learning model that can record and interpret the data traffic that passes through it. This is a learning model that combines artificial neural networks, decision trees, and the New Bayes model; it is used to identify security assaults and threats in the network that corresponds to the controller node. Figure 1 depicts the suggested method's steps in detail as a diagram (1). The SDN domain is separated into numerous subdomains in the first phase of the proposed technique, and the responsibility of monitoring each subdomain is assigned to a controller who may exchange security rules with other subdomains. Each controller will deliver a list of certified users relevant to its section to other controllers in the suggested way. If communication between two users is required, the user's trustworthiness is determined by exchanging messages between the controllers. The data exchange activity will take occur if at least one controller has authenticated each of the two participants to the connection. According to the software-centric software structure suggested in this study, all communications between nodes in a subnet are routed through the subnet's control node. As a result, each controller node employs a learning model to continually evaluate network traffic data and detect assaults. This model is an integrated learning system that uses an artificial neural network, a decision tree, and New Biz to identify the presence of DoS assaults using statistical data gathered from each traffic flow.

Each of these steps will be described below.

SDN-Based Network Communication Structure

The network domain is separated into multiple subdomains in the first phase of the proposed technique, and the responsibility of monitoring each subdomain is assigned to a controller who exchanges security rules with other subdomains. The position information of items in the network may be used to segment them in a pattern. As a result, each subnet is treated as a cluster. Each cluster node on the network only communicates with its SDN controller directly (it will not even communicate with its neighbors). The goal is to have network users authenticate through the SDN controller to avoid security threats both inside and outside the clusters. Furthermore, using this structure, each node is forced to communicate its traffic with others through the controller node, making it possible to monitor and identify attacks for all information transmitted in the network using the learning model. The controllers of each subdomain must communicate information about their members for data to be securely exchanged across mobile nodes in the network. When a source node wants to communicate data to another node, it sends the target node ID to its subdomain controller first. If the source and destination nodes are on the same subdomain, the two nodes are connected by sending a reply message to the source node. Otherwise, the message supplied from the source is transmitted to the central controller C by the controller node. Following receipt of this message, the C_t node sends packets to the controllers of the other subdomains with the target node ID. A confirmation message is sent to the source node by the controller that has the destination node in its subdomain through the central C_t node. A link will be formed between the two nodes in this manner. Figure 1 shows an example of the suggested algorithm's communication process between objects (2).

Figure 2 assumes that a node in subdomain 1 such as A wants to connect with a node in subdomain 3 such as B. In this scenario, node A sends a message to controller C1 specifying the target node ID. This controller transfers the received packet to the central controller Ct since node B is not in the C1 domain. This message is also sent to other controllers by this controller (C2 and C3). Because the destination node B is in

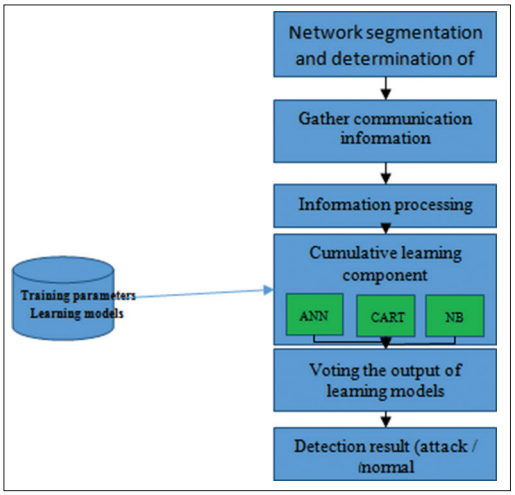

Figure 1: Block diagram of the proposed method

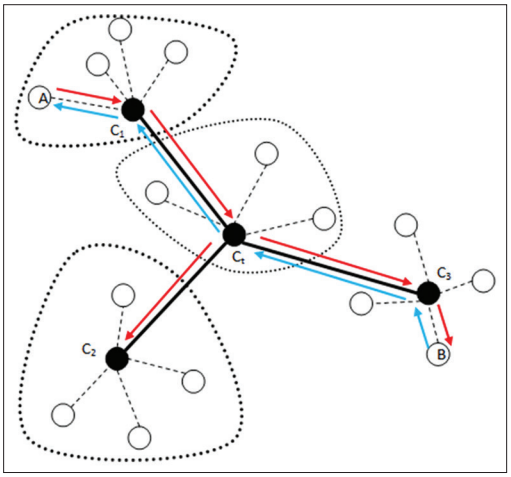

Figure 2: An example of the process of communication between objects in the proposed method

the subdomain of node C3, this subdomain sends a response packet to the source node. Finally, using the discovered route, the data packet is transmitted between the two nodes. The suggested aggregation approach is used to analyze traffic information and detect intrusions while data are being routed by controller nodes. The structure of this learning model will be described in the following sections.

Detection of Attacks Based on Cumulative Learning

As mentioned earlier, each controller node in the software-driven network is equipped with an integrated learning model that is capable of recording and processing the data traffic flowing through it. This learning model consists of the following learning models:

1. Artificial neural network
2. Decision tree
3. New Biz model.

Each of the above learning models analyzes the traffic patterns that travel through their respective controllers, and then classifies them using the voting mechanism. The learning model based on these controller nodes will solely evaluate traffic received by its subnet nodes to lessen the complexity and computational strain imposed on them. By doing so, malicious malware may be avoided from infecting network equipment and routers from the start of the transmitting process, and the rogue node can be easily recognized. Figure 1 shows an example of this procedure (3). On Figure 3, it is assumed that the two nodes are in the same subnet to keep things simple. Node A is a malicious node, whereas node B is a regular node. Assume that each of these nodes wants to send each other a message. As previously stated, all nodes in the network exchange data through their subnet controller, which processes all messages supplied by subnet members using the integrated system's learning models.

When node A transmits a malicious message to the controller, the communication features are retrieved and categorized by three models of artificial neural networks, decision trees, and the Nivebase model before each operation. The aggregate system's ultimate output is then decided using the voting process, depending on regular communication or the existence of an assault. The connection will be stopped and the message will be erased if the integrated system qualifies the collected characteristics. This happened in the case of a hypothetical message sent from node A to node B. The message sent by node B, on the other hand, is noticed by the integrated system in the regular controller and forwarded to node A. The process of recognizing assaults using the integrated learning system will be described in the following sections.

Data processing

The first step in the process of identifying DoS attacks by the proposed aggregation system is data preprocessing. Data preprocessing is done through the following steps:

- Numerically quantify the nominal characteristics of the traffic flow being processed. For example, the "connection type" attribute can have one of the ICMP, UDP, and TCP modes, which are replaced by numbers 1–3. The numerical properties obtained for the traffic flow are normalized using the following equation.

$$N_i = \frac{n_i - n_{min}}{n_{max} - n_{min}} \quad (1)$$

The input property vector for normalization is represented by n_i and the minimum and maximum values in the n_i property vector are represented by n_{min} and n_{max}, respectively. Thus, the traffic flow characteristics are translated into a numerical representation in the range,[0,1] and these characteristics are employed as input to learning models in the proposed method's subsequent phases.

Classification of features based on artificial neural network

The first learning model used in the proposed integrated system for detecting DoS attacks is the artificial neural network. This neural network is a prosthetic network with a hidden layer. The secret layer of this network has 10 neurons and its transmission function is determined by logarithmic sigmoid type. Furthermore, the number of input layer neurons is equal to the number of traffic flow characteristics, and the number of output layer neurons is equal to the number of types of attacks. The amount of output from these neurons determines the type of attack determined by the neural network. The structure of this network is shown in Figure 4. The artificial neural network is the initial learning model utilized in the proposed integrated system for detecting DoS threats. This neural network is a hidden layer prosthetic network. This network's hidden layer comprises 10 neurons, and its transmission function is of the logarithmic sigmoid type.

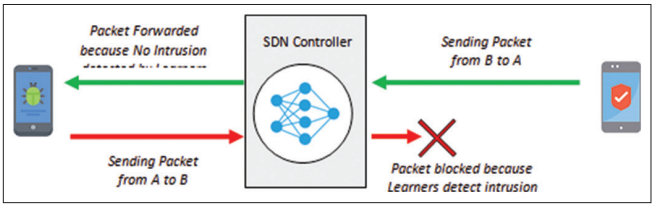

Figure 3: How controller nodes work in identifying attacks based on cumulative learning

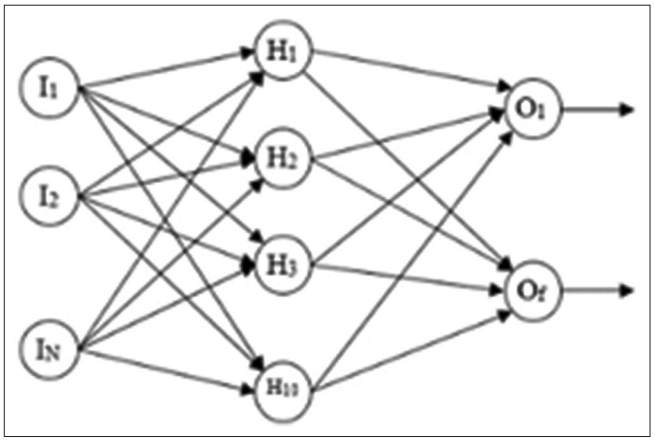

Figure 4: Neural network structure to determine the type of attacks in the aggregate model of each controller node

Classification of features based on artificial neural network

In addition, the number of input layer neurons equals the number of traffic flow characteristics, whereas the number of output layer neurons equals the number of attack kinds. The sort of assault decided by the neural network is dictated by the amount of output from these neurons. Figure 1 depicts the network's structure (4). In addition, the number of input layer neurons equals the number of traffic flow characteristics, whereas the number of output layer neurons equals the number of attack kinds. The sort of assault decided by the neural network is dictated by the amount of output from these neurons. Figure 1 depicts the network's structure (4).

The neural network learning algorithm for training based on the input data is as follows:

1. At the beginning of the neural network training, the initial values of the weights for the w_c vector are set equal to random numbers and t = 0 is considered
2. For each layer in the neural network, the input vector Di is applied to the neurons of the current layer of the neural network and based on the weight vector of the neurons in the current iteration and the logarithmic sigmoid activation function, the output d is calculated
3. The values of neural network weights in the w_c vector are updated using the following relation:[15]

$$w_j(t+1) = w_j(t) + \mu \times E \times x \quad (2)$$

In the above relation, μ is the learning rate, E is the difference between the actual output and output of the neuron, and x is the input data.

Steps 2–4 are repeated until the neural network error is less than the threshold value or $t < t_{max}$, increasing by t one value. Otherwise, the neural network training will end.

Classification of properties based on the decision tree

The decision tree is the second learning model utilized in the proposed integrated system for identifying assaults. The decision tree algorithm is a data mining approach that, despite the fact that it does not involve sophisticated computations and is simple to grasp, has comparable accuracy to other classification methods. As a result, it may be used to solve a variety of categorization problems. As a result, a structure based on decision tree and regression classification and regression trees are described in this part to identify assaults. The decision tree created by the division and regression tree attempts to anticipate and classify future observations. The goal of this strategy is to eliminate contaminants in each of the categories. When all of the items in a subset belong to the same goal category, a node is totally free of impurities. Intervals and categories are two sorts of predictive and objective adjectives. All divisions will be binary, meaning that each node will have only two subgroups. Instead of employing stop laws, the decision and regression tree creates a succession of subtrees by first creating a big tree and then pruning it until only the root node remains. The classification cost of each subtree is then estimated using cross-validation. Finally, the decision tree model is built by selecting the subtree with the lowest projected cost.

Classification of properties based on New Biz

The root node and other parent nodes in the final tree define the branch, while the leaf nodes define the target classes. The tree traversal begins at the root node and continues until it reaches a leaf to classify the input data. The class that the data reach by scrolling the tree branches will be the decision tree's output for the input data. The proposed integrated system employs a New Base model as the third learning model for detecting DoS threats in controller nodes. The Bayesian technique is basically a way of categorizing events based on their likelihood of occurring or not occurring. The chance of an event occurring in the future may be calculated in the New Business categorization by looking at prior events. Bayesian classification is used for problems in which each instance of x is selected by a set of attribute values and the objective function f (x) from a set such as V. The Bayesian mechanism for classifying a new sample is to identify the most likely class or target value of v_{MAP} by having the attribute values $<a_1, a_2\ldots, a_n>$ that describe the new sample:[15]

$$v_{üüüü} = argmax_{j\in} P\ v \quad a_1\ a_2\ \ldots\ a \quad (3)$$

Using Bayes' theorem, the above statement can be rewritten as follows:[15]

$$üüüü_{MAP} = \frac{üü(üü_j|\ _1,\ _2,\ldots,\ _n)\ (\ _j)}{P(a_1,a_2,\ldots,a_n)} = argmax_{v_j\in V} P(a_1,a_2,\ldots,a_n \mid v_j)P(v_j) \quad (4)$$

Now, using educational data, we try to estimate the two sentences of the above equation. It is easy to calculate from educational data how much vj is repeated in the data. But calculating different sentences *P* (a_1, a_2,…, a_n/v_j) will not be acceptable in this way unless we have a large amount of educational data. Therefore, we have to look at each sample several times to get a good estimate of it. The Bayesian method of classification is based on the premise that attribute values are conditionally independent of having objective function values. In other words, this assumption implies that provided that the output of the objective function is observed, the probability of observing the attributes a_1, a_2…, a_n is equal to multiplying the probabilities of each attribute separately. If we replace this concept with Equation 4, the Bayesian classification method results:[15]

$$v_{üüüü} = argmax_{\ j\in}\ P(v\ )\prod_i P\left(a\mid v\ \right) \quad (5)$$

VNB is the output of Bayesian classification for the objective function. The number of P sentences (a_i/v_j) to be calculated in this method is equal to the number of attributes multiplied by the number of output sets for the objective function, which is much less than the number of P sentences (a_1, a_2,…, a_n/v_j).

Detection of attacks in a cumulative model based on voting technique

The voting approach is the final stage in detecting DoS threats in the proposed aggregate system. The goal of the voting procedure is to increase the accuracy of algorithm

classification when compared to when each algorithm is employed individually. This is known as aggregation-based learning or voting. Each of the classification algorithms may have errors in classifying some samples; therefore, the purpose of voting-based techniques is to reduce the resulting error and increase the accuracy of sampling samples. It has been theoretically proven that the use of voting techniques can improve the results.

SIMULATION AND RESULTS

As a consequence, in the final stage of the proposed technique, the neural network, New Biz, and decision tree classification models each do the classification operations of the test samples individually, and the final output of the system is decided by voting on the results of all three models. The suggested model was evaluated using data from the NSLKDD database[16] for this purpose. More than 25,000 data records in the field of information shared in the network are contained in the database used to evaluate the performance of the proposed model. Packet data sent during various forms of network incursions are stored in these records. Table 1 shows the data from the NSLKDD database (1).

Table 1 shows that around 80% of the data in the database are allocated to assaults. This database comprises 42 statistical aspects of traffic flow characteristics in each data record. During the simulation, it is assumed that each node in the network exchanges information with other nodes depending on one of the records in this database. The data flow traffic information is categorized by the learning models in the controller node integration system throughout the data exchange process, and the existence of assaults is identified. The performance of the suggested aggregate model in distinguishing regular traffic flows from streams associated with DoS assaults will be examined in the following sections.

To ensure the validity of the results, the simulation operation was repeated 10 times. During this process, in each iteration, 90% of the database samples are used as learning model samples and the remaining 10% are used as test samples. Thus, after 10 repetitions of the experiment, the exchange of all samples belonging to normal categories and DoS attacks in the NSLKDD database will be simulated by network nodes. Furthermore, the performance of the proposed aggregate model in detecting DoS attacks is compared with two SVM learning models in Zhe *et al*.[13] and the nearest neighbor K algorithm KNN in.[14]

The accuracy results of each of the compared algorithms in detecting network attacks are shown in Figure 5.

During several iterations, Figure 5a depicts the accuracy of the proposed technique and the methods compared to it. The average accuracy of the various approaches is also shown in Figure 5b. As shown in Figure 5, the attack detection operation may be done with 99.6% accuracy employing a combination of artificial neural network, decision tree, and New Biz in the suggested aggregate system, which is a little improvement. It's 2% more precise than the approaches described in Zhe *et al*.[13] and Dong and Sarem.[14] The disruption matrix will be examined in greater depth to analyze the efficacy of the suggested aggregation mechanism in identifying assaults. The

Table 1: Types of information available in the nslkdd database

Type of attacks	Percentage of available data
Normal (no attack)	19.48
DoS attacks	73.9
U2R attacks	1.34
R2L attacks	5.2
Probe attacks	0.07

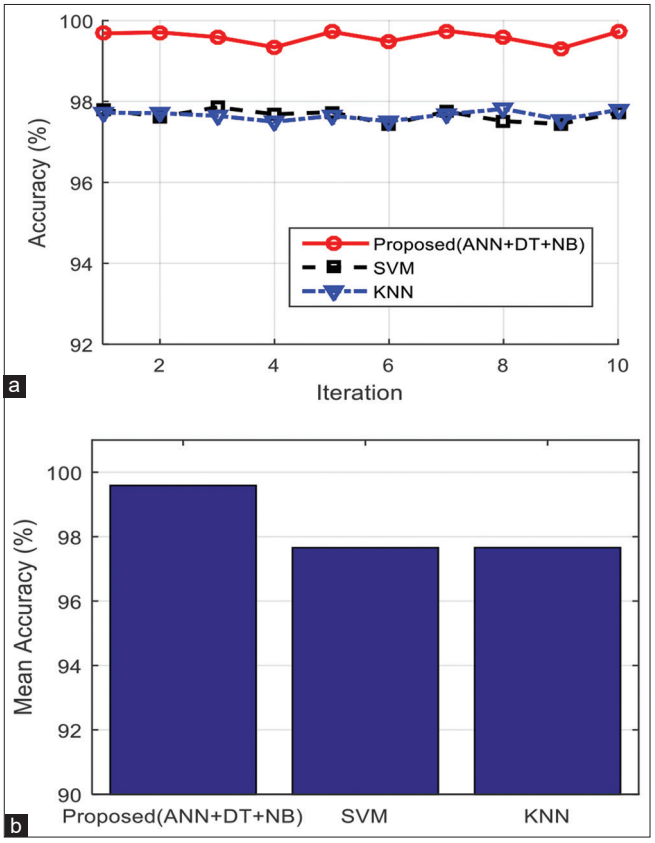

Figure 5: Accuracy of learning algorithms in detecting network attacks by controller node. (a) Accuracy in different iterations and (b) average accuracy

clutter matrix resulting from the detection of DoS assaults during 10 replication simulations for the proposed aggregation system is compared to the clutter matrix coming from the SVM[13] and KNN[14] approaches in Figure 6.

The value 35,201 in the first row and column of the perturbation matrix shown in Figure 6a reflects the number of normal connections examined that were accurately recognized as normal by the suggested aggregation method. The clutter matrix identifies this number as TN. The number 92 in the second row and first column represents the number of regular communications that the suggested DoS attack model mistakenly detected. In the clutter matrix, this quantity is designated as FP. The clutter matrix's second row and second column, which reflect the number 40,068, identify service denial assaults that the proposed aggregate system accurately classifies as TP occurrences. In addition, the number 219 in the first row and second column represents DoS attempts that the

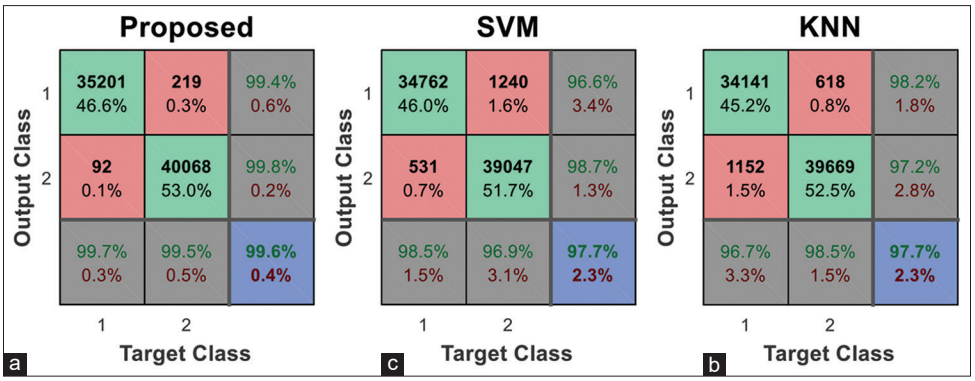

Figure 6: Disruption matrix (a) Proposed aggregate system, (b) support vector machine, and (c) K-nearest neighbor in attack detection

Table 2: Comparison of the performance of the proposed integrated system with other learning models

Specificity percentage	Percentage of sensitivity	Percentage accuracy	Title
96.9221	98.4955	97.6568	SVM[13]
98.4660	96.7359	97.6581	K-nearest neighbor[14]
99.4564	99.7393	99.5885	Proposed integration system

suggested aggregating system mistook for regular. The clutter matrix identifies this number as FN. The suggested method's improved efficiency in detecting DoS assaults is confirmed by a comparison of its matrix with the examined approaches.

Table 2 compares the results of different learning models with the test results of the proposed integrated system for detecting DoS assaults in the network. The sensitivity and specificity criteria are compared in this table. The ratio of total assaults accurately recognized by the learning model is called sensitivity, and it's determined as follows: $Sensitivity = \frac{TP}{TP + FN}$ (6)

In this relation, TP is the number of attack streams that have been correctly detected and FN is the number of attack traffic streams that have been erroneously identified as normal streams. The property criterion is used to measure normal streams that are correctly classified. This criterion is calculated as follows:

$$Specificity = \frac{TN}{TN + FP} \quad (7)$$

TN is the number of accurately recognized normal traffic streams, whereas FP denotes the number of normal traffic streams that have been identified as attack traffic streams.

When the efficiency of the suggested approach is compared to the efficiency of the compared methods, it is clear that the provided solution may improve the criteria of accuracy, sensitivity, and specificity in the process of detecting DoS assaults.

CONCLUSION

A new technique for detecting DoS attacks in the structure of IoT is provided in this research, which is based on software-based network and machine learning. The suggested solution divides the SDN domain into numerous subdomains, each with its own controller for exchanging security rules with other subdomains. Each controller will deliver a list of certified users relevant to its section to other controllers in the suggested way. In this example, if two users need to interact, the user's credentials are passed back and forth between the controllers. The data exchange activity will take occur if at least one controller has authenticated each of the two participants to the connection. In the topology suggested in this study, all traffics between nodes in a subnet are routed through the subnet's control node. As a result, each controller node employs an integrated learning model to continually evaluate network traffic data and detect assaults. This learning model incorporates an artificial neural network, a decision tree, and a New Biz model that uses statistical information gathered from each data stream to identify the likely existence of assaults. MATLAB software was used to implement and assess the suggested approach, and the results of simulating the new method's performance were compared to earlier methods. The data from the NSLKDD database were utilized in the tests, and the suggested learning model's accuracy in identifying various forms of network assaults was assessed. The results of these tests revealed that the suggested method's cumulative learning model can identify DoS attacks in network traffic flows with accuracy of 99.6%. Other machine learning models, such as probabilistic neural networks, deep learning approaches, and others, might be used in the future to identify assaults. It appears that optimization algorithms may be used to rank learning models and then establish a weight value for each learning model in the integrated system based on that ranking. As a result, the future research might focus on resolving this problem.

REFERENCES

1. J. Li, M. Siddula, X. Cheng, W. Cheng, Z. Tian and Y. Li. Approximate data aggregation in sensor equipped IoT networks. *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 44-55, 2019.
2. J. Marietta and B. Mohan. A review on routing in internet of things. *Wireless Personal Communications*, vol. 111, no. 1, pp. 209-233, 2020.
3. X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, X. A survey on ensemble learning. *Frontiers of Computer Science*, vol. 14, no. 2, pp. 241-258, 2020.
4. G. Perrone, M. Vecchio, R. Pecori, and R. Giaffreda. The Day After Mirai: A Survey on MQTT Security Solutions After the Largest

Cyber-Attack Carried Out through an Army of IoT Devices. In: *2nd International Conference on Internet of Things, Big Data and Security*, IoTBDS, pp. 246-25, 2017.
5. S. Andy, B. Rahardjo and B. Hanindhito. Attack Scenarios and Security Analysis of MQTT Communication Protocol in IoT System. In: *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pp. 1-6, 2017.
6. X. Liu, T. Zhang, N. Hu, P. Zhang and Y. Zhang. The method of internet of things access and network communication based on MQTT. *Computer Communications*, vol. 153, pp. 169-176, 2020.
7. S. N. Firdous, Z. Baig and A. Ibrahim. Modelling and evaluation of malicious attacks against the IOT MQTT protocol. *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 748-755, 2017.
8. H. Ramos, S. Villalba, R and Lacuesta. MQTT Security: *A Novel Fuzzing Approach*. Wireless Communications and Mobile Computing, 2018.
9. A. Vähä-Sipilä, "*mqtt fuzz*"., 2015. Available from: https://github.com/F-Secure/mqtt_fuzz
10. N. Moustafa, B. Turnbull and K. Choo. An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815-4830, 2018.
11. N. F. Syed, Z. Baig, A. Ibrahim and C. Valli. Denial of service attack detection through machine learning for the IoT. *Journal of Information and Telecommunication*, vol. 4, no. 4, pp. 482-503, 2020.
12. C. Kharkongor, T. Chithralekha and R. Varghese. A SDN controller with energy efficient routing in the internet of things (IoT). *Procedia Computer Science*, vol. 89, pp. 218-227, 2016.
13. W. Zhe, C. Wei and L. Chunlin. DoS attack detection model of smart grid based on machine learning method. In: *2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pp. 735-738, 2020.
14. S. Dong and M. Sarem. DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks. *IEEE Access*, vol. 8, pp. 5039-5048, 2019.
15. M. Bharati and M. Ramageri. Data mining techniques and applications. *Indian Journal of Computer Science and Engineering*, vol. 1, no. 4, pp. 301-305, 2010.
16. R. Bala and R. Nagpal. A review on kdd cup99 and nsl nsl-kdd dataset. *International Journal of Advanced Research in Computer Science*, vol. 10, no. 2, pp. 64-67, 2010.