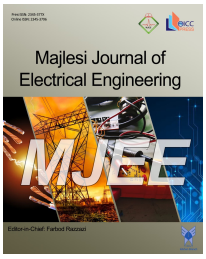

Volume 18, Issue 3, 182441 (1-14)

Majlesi Journal of Electrical Engineering (MJEE)

https://doi.org/10.57647/j.mjee.2024.180341

A two-layer cloud-fog intrusion detection system using enhanced KNN and MLP neural networks

Ali Kaffash[1], Seyed Reza Kamel[2,*], Maryam Kheirabadi[1]

[1]*Department of Computer Engineering, Neyshabur Branch, Islamic Azad University, Neyshabur, Iran.*
[2]*Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran.*

[*]**Corresponding author:** rezakamel@ieee.org

Original Research

Received:
3 April 2024
Revised:
2 June 2024
Accepted:
15 July 2024
Published online:
30 September 2024

Abstract:

The concept of the Internet of Things (IoT) and its countless applications are considered as an inseparable part of the modern technology era. The placement of IoT- based devices and their limitations make the environment more vulnerable due to its openness. Security plays a critical role in IoT applications due to the pervasiveness of the IoT in all of the aspects in daily life. On the other hand, final devices such as their limited computing power, large number of devices connected to each other, and communication between devices and users do not allow for using traditional methods to solve security issues. Intrusion detection systems (IDSs) which can separate malicious traffic from normal mode are among the effective solutions in this field. the installed IDS should be highly accurate and lightweight to affect accuracy. In order to bring services closer to electronic devices, a concept called “fog” has emerged. A large number of studies have been conducted to make the light IDS for IoT networks utilizing various methods. The present study aims to propose a two-layer hierarchical IDS based on machine learning, which detects attacks by considering the limitations of IoT resources. In order to create an efficient and accurate IDS, the combination of two improved K-nearest neighbor (KNN) algorithms and a multi-layer perceptron (MLP) neural network was applied in the fog and cloud to separate the attacks from normal traffic, respectively. we evaluated our proposed method using IOT23 dataset. The results prove the improvement in accuracy, compared to the previous methods.

Keywords: IoT security; Intrusion detection system; Fog; Cloud; KNN; MLP

1. Introduction

The concept of the Internet of Things (IoT) has entered almost all of the fields in the modern world. In addition, electronic devices with the ability to connect to the Internet are constantly increasing. More than 500 billion connected electronic devices are predicted to be on the planet by 2025 [1]. Small, portable, and not expensive electronic devices and their ability to connect to the Internet lead to their use by various users. The IoT allows several different physical electronic devices to observe, hear, feel, think, and communicate to share information and make decisions to perform specific tasks. Further, IoT-based programs can improve people’s lives effectively. Opening and closing the garage door, turning on the coffee maker, or adjusting the temperature in the house are among the examples related to applying the IoT to human life [2]. Measures should be taken to transfer and process the data of IoT-based devices due to their limited resources. Furthermore, a large number of IoT applications use cloud computing to process and store data [2].

Cloud computing is considered as an innovative computing technology which provides ease of work and access to computing resources based on strategy in a common pool according to network demand with the ability to publish and supply easily and quickly. However, security issues play a critical role when businesses begin to outsource data and applications to cloud computing providers [3].

Cloud computing [4] suffers from delays created by the distance between IoT electronic devices and data centers [5]. A concept called fog computing (FC) has emerged in order to bring services closer to devices [6]. The concept of

cloud computing eases processing and storing the data close to IoT devices, reduces the traffic sent to the cloud [7], and allows applications that require immediate and real-time response to receive their response faster [1, 2] and optimize various key Quality of Service (QoS) [8].
Fig. 1 shows the FC architecture including IoT layer, FC, and cloud computing layer. As observed, FC is considered as an interface between cloud and IoT devices which brings connected devices closer to network, memory, and computing services. FC is regarded as an approach to bring computing and storage services together for becoming operational in real time and becoming closer to networked devices. FC can be considered as an extension of cloud computing to adapt to IoT data. FC supports the process of computing offloading and related services close to users, resulting in increasing computing speed and reducing burden on traditional cloud computing data centers. Such new computing model provides intelligent services to meet data optimization, privacy, security, and real-time processing requirements in collaboration with cloud computing and mobile edge network infrastructure.
The method of securing devices and protecting data against attacks is considered as the main concern regarding IoT systems. Cyber-attacks include deliberate exploitation or unauthorized access to personal or organizational infrastructure [9]. Protecting IoT devices against attacks due to the heterogeneity of devices and protocols, direct access to devices through the Internet, and resource limitations in devices are among the critical challenges in the IoT [10]. Traditional security devices face various obstacles to be utilized in the IoT due to the following basic limitations [11]:

1. Limited computing power of IoT
2. Abundancy of devices connected to each other
3. Constant transfer and sharing of data among things and users to achieve particular goals

The fog layer, which is considered as the closest processing layer to the low-power devices of the IoT, is as more powerful than devices in terms of processing and exhibits a limited computing power compared to the computing clouds. In addition, the data related to the aforementioned layer fail to suffice for an accurate and comprehensive training due to its access to only its own local data. Therefore, more complex detection models such as machine, deep, and group learning can be implemented although the processing power and available data fail to suffice to apply a comprehensive model [1].
A large number of cyber-attacks occur in this area, despite the methods presented so far for the security of the IoT. Such attacks can damage IoT devices, data exchanged between such devices, and various service requests generated. The above-mentioned attacks seek to provide access to remote local systems, prevent response to requested service, ease more access permission such as root user, and observe potential targets for attacks such as probing and the like. Generally, attacks occur to limit or prohibit the provision of services requested by devices [12]. Intrusion detection system (IDS) is considered as a key technique to eliminate the aforementioned obstacles. Such technique aims to effectively reduce the delay and security threats using FC.
Most of the IDSs presented so far have identified the attack mode from normal utilizing binary classification and traditional datasets. Such IDSs do not identify the type of attack, which is critical for countermeasures. In most of the cases [13–21]. In addition, most of the offered methods exhibit shortcomings in terms of high false positive (FP) rate, time overhead, low detection of unknown attacks, and the problems related to delay.

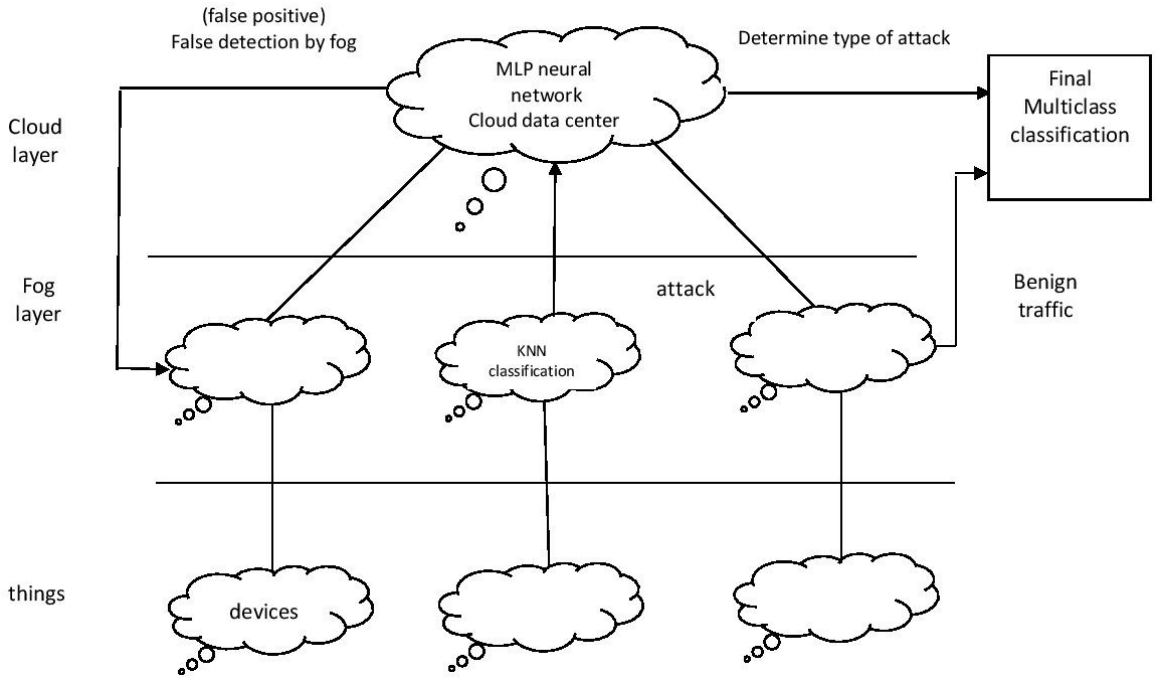

Figure 1. Proposed FC architecture.

Here, an efficient two-layer architecture is presented, which applies heterogeneous computing capability of fog and cloud computing to provide intrusion detection related to a wide range of IoT attacks by combining improved K-nearest neighbor (KNN) algorithm and MLP neural networks. Due to deployment of the intrusion detection module based on the improved KNN algorithm in the fog level, the speed of detecting normal traffic from attack increased by order of O(n). In the next step, the fog is updated to increase the accuracy of the IDS in the fog when the normal traffic is mistakenly detected as an attack in the cloud. In the next procedure, the method of reducing the delay of the IDS due to detection of attacks in the initial stage compared to the cases without fog and the mode of using MLP in the fog is displayed. Then, the performance of the proposed method in providing more optimal time complexity with mathematical calculations is shown.
Finally, the performance of the proposed model is evaluated experimentally utilizing the IOT23 dataset. Based on the results, the approach proposed in multiple performance measures performs better than the existing ones.
The remining sections of the study are as follows. Section 2 describes the types of IDSs. Section 3 explains the proposed approach. Section 4 examines the results and compares this study with the most recent one. Finally, conclusions and summaries are presented.

2. Literature review

Most of the studies conducted in the field of FC architecture focus on detecting the malicious fog nodes and threats. Here, the latest studies and solutions regarding intrusion detection based on fog and cloud computing for the IoT environment are investigated. The IDSs based on machine learning have attracted a lot of attention among the presented solutions and are applied as one of the most popular and widely used methods.
A FC-IDS framework was proposed by [13] using the hypergraph clustering model. Such method can effectively describe the communication between fog nodes under DDOS attack. The results indicate that the offered method can detect DDOS attacks effectively.
A light IDS [14] was proposes for the fog environment for the IoT. The offered solution utilizes different base learners applying several well-known algorithms and building different group classifications for attack detection and classification. Experiments performed on the NSL-KDD dataset [22] demonstrated that the proposed IDS model is considered as more appropriate than other ones presented in the FC environment. The presented model exhibits high accuracy in binary and multiclass classifications.
In another study [15], provided IDS based on artificial neural networks for IoT fog nodes. The proposed approach identifies the compromised fog node and takes necessary actions to ensure communication availability. Based on the results, the proposed method can identify anomalies related to system failures or cyber-attacks.
A light IDS was presented by [16], based on the hidden MLP to identify cyber-attacks such as Hydra-FTP, Hydra-SSH, Add user, and Java-Meterpreter. Meterpreter and Web Shell attacks occur at the fog node layer. A feature extraction technique was used by changing the vector space representation through n-gram transformation in order to simplify the IDS. A sparse matrix is utilized for its format compression. The linear correlation coefficient (LCC) is applied to compensate for zero values, and a mutual information feature is selected to reduce the number of features. In another study, [23] tested the proposed method on the Australian Defense Force Academy Linux Dataset (ADFA-LD).
A fully automatic synthetic IDS was presented by [17] for the security of fog against cyber-attacks. The proposed model uses several layers of recurrent neural networks (RNNs) designed to implement the security of FC, which brings closeness to end users and IoT devices. The NSL-KDD dataset [24] is utilized in the above-mentioned method and the performance of the model is measured applying a variety of common metrics.
In another study [18], a method is proposed that anomaly-based IDS dedicated to Bluetooth mesh networks. Machine learning algorithms are used to classify traffic and identify malicious behaviors in the IoT. The proposed solution includes collaborative decision-making conducted by several watchdogs distributed in different areas of the considered network, which process local traffic. The optimal placement of watchdogs is proposed based on the simulations performed by BMWatchsSim software. The results indicated that the placement of the watchdog proposed by the simulation allows the effective detection of real-world intrusions.
In research [25], a method is presented an algorithm to detect anomalies in IoT traffic based on vector convolutional deep learning (VCDL). The term convolutional vector is utilized because the method works on the vector, not the matrix. The offered model benefits from two convolutional layers and two mixed ones arranged alternately, as well as two hidden layers with nine and seven neurons, and rectified linear unit (ReLU) activation function. The SoftMax activation function is applied in the output layer. A framework with a master fog was presented for general learning and a large number of active fog nodes for local learning, despite facing a lot of obstacles in identifying normal traffic from attack. The aforementioned method only detects about 90.5 %, indicating an increase in the number of FPs. Each local node processes the data received from connected IoT devices. Then, the results are stored in the main fog and applied to detect unknown intrusions after the learning process is finished. The aforementioned process makes the model share the best-obtained parameters with learning to avoid overfitting the proposed framework [1]. The idea of using the main and local fogs is implemented here appropriately. However, the high computational cost due to the algorithm utilized and its relatively high FP rate compared to other methods are among the disadvantages of the above-mentioned method.
A solution was presented by [26] for the limitations related to the centralized IDS for devices with limited resources in two semi-distributed and distributed methods which are regarded as a combination of feature extraction with opti-

mal performance and coordinated selection and analysis of potential edge-fog intrusion. The parallel machine learning models related to the partition attack dataset were developed separately in order to distribute the computational tasks. Based on the semi-distributed mode, parallel models running on the edge side are applied for feature selection next to each other, followed by a single MLP classifier on the fog side. Based on the distributed mode, parallel models individually perform feature selection and MLP classification, after which the outputs are combined by a coordinated edge or fog to make the final decision.

In another study, researchers [19] presented a new approach based on anomaly implemented by a hybrid feature selection engine that selects the most relevant features. In addition, the random forest (RF) algorithm classifies each traffic as abnormal or normal. The efficiency is studied by one of the latest datasets called IOTID20, which is based on the real environment of the IoT. The results indicate an accuracy of 99.95, 99.07, and 99.96 % for DOS, MITM, and scan attacks, respectively. Reviewing with a dataset, limitation of the aforementioned types of attacks, and use of binary classification alone are among the weaknesses of the proposed method.

A model was presented by [20] for the IoT based on cloud and fog for anomaly detection applying the improved simple Bayes classification algorithm and principal component analysis. Integrating the IoT with cloud and fog computing result in providing an improved framework to support smart city programs. Cloud computing is used to store data at a high level of management, while fog computing can provide various services of smart city programs based on the IoT. The integrated architecture of the IOT-FOG-CLOUD computing model is presented in the above-mentioned study. The proposed solution utilizes an improved naïve Bayes (INB) classification algorithm based on principal component analysis (PCA) in order to analyze according to the network-based IDS. The UNSW-NBA15 dataset [27] which is used to assess the attack detection model is applied in the aforementioned study. PCA algorithm is utilized to extract features from the dataset, while INB classifier is applied to classify attacks. The efficiency of evaluation parameters is improved significantly in the proposed method. The proposed INB-PCA model improves the accuracy and discovery rate up to 92.48 and 95.35 %, respectively. Conducting experiments with a dataset, checking the results with three common machine learning algorithms such as RF, KNN, and support vector machine (SVM), not comparing the efficiency with similar studies, and a relatively high FP rate are among the weaknesses of the above-mentioned study.

A new model was presented by [28], to reduce the feature dimensions of large-scale traffic data of the IoT using the encryption stage of the long short-term memory (LSTM) autoencoder (LAE). The relevant long-term changes are analyzed in the set of low-dimensional features produced by LAE utilizing deep bidirectional long short-term memory (BLSTM) in order to correctly classify network traffic samples. To validate the effectiveness of the proposed hybrid deep learning method, more experiments were conducted applying the BOT IoT dataset, indicating that LAE significantly reduced the memory space required for large-scale network traffic data storage by 89.91 % and performed better than the feature dimension reduction methods between 18.92–27.03 %. Conducting experiments with a dataset, and FP rate are among the disadvantages of the aforementioned method.

In another study, [29] presented their proposed method which is considered as new distributed and light IDS. Variable LAE and MLP were combined to provide efficiency and accuracy for the above-mentioned method. The IDS operates in a two-layer fog architecture including an anomaly detector within the fog node and an intrusion detection module within the cloud. The proposed method is measured by two datasets presented recently for cyber-attacks. Based on the results, the built system can accurately determine the normal behavior in fog nodes and identify different attacks. Various types of attacks such as DDOS exhibit a high detection rate of 99.98 % and a low false alarm rate of less than 0.01 %. The offered system performs better than the existing techniques in terms of detection and FP rates. However, processing by the MLP algorithm is among the disadvantages of the aforementioned method.

In another study, [30] presented a two-stage approach. Traffic is analyzed with an additional binary tree classifier in order to detect and identify the intrusion during the first stage. Events detected as intrusions are analyzed during the second step by a group approach containing redundant tree, RF, and deep neural network (DNN). Then, an extensive evaluation is performed using the BOT-IOT, NSL-KDD, IOTID20, and CICIDS2018 datasets. The results indicate that the proposed method can perform better than other machine learning methods in all of the databases. In other words, the above-mentioned method acts as a hybrid one, the power of which increases due to the combination of several methods in the evaluations. The high prediction time and comparison with the usual machine learning methods instead of similar ones are among the disadvantages of the aforementioned method.

In another study, [21] presented an IDS for the IoT based on fog for DDOS attacks utilizing deep intelligence (DI-ADS). The proposed framework applies a deep learning model (DLM) to detect DDOS attacks in the network. A DLM is installed on the computing module of the fog node to predict the behavior of IoT end devices. The performance on deep neural MLP (DNMLP) and LSTM models are compared with conventional machine learning models such as SVM, KNN, logistic regression (LR), and RF algorithm in order to select the best DLM in the fog layer. Simulation is conducted using Python Anaconda framework considering Mendeley new dataset named DDOS-SDN [31] containing three DDOS attacks such as UDP Flood, TCP Syn, and ICMP attacks. Based on the results, the accuracy of the proposed method equals 99.44 % compared to other machine and deep learning algorithms. The proposed model (DNMLP) is considered in the proposed framework for implementation in the fog layer in order to perform better in detecting DDOS attacks.

A two-layer hierarchical intrusion detection mechanism was

presented by [12] based on machine learning, which can effectively detect IoT network intrusions. The offered model exploits the resources in the fog layer of the IoT network effectively by deploying multilayer feedforward neural networks (MFFNNs) in cloud-fog pair infrastructure efficiently to detect network attacks. To this aim, analysis is dynamically distributed across the fog and cloud layer, resulting in bringing real-time analysis of traffic data closer to IoT devices and end users. Then, two datasets of NSL-KDD and CICIDS 2017 are utilized in the experiments. The results indicate that the proposed method performs better than other ones, especially with the NSL-KDD dataset in the performance of some criteria such as accuracy, recall, and F score. The offered model is evaluated in terms of service time improvement, lower delay, and optimal energy in the case where there is only a cloud, indicating a significant improvement.

Most of the offered IDSs operate based on binary classification, while the type of attack should be identified for countermeasures. A large number of methods apply traditional datasets such as NSL-KDD and KDD-CUP, which are not regarded as efficient for detecting new attacks. All of the reviewed methods are considered as admirable, despite exhibiting defects in detection, low detection of unknown attacks, time overhead, high FP rate for less known attacks, and delay issues. An intrusion detection architecture based on FC is proposed for IoT network monitoring and intrusion detection in order to eliminate the above-mentioned obstacles.

As indicated, traditional IDSs themselves on existing attacks and the conditions and limitations of IoT networks. In addition, light and efficient IDS should be provided. Here, a method is presented to identify and separate normal traffic from attack (in the fog) and detect the type of attack (in the cloud) with high accuracy considering the conditions and limitations of IoT environments. The details of the study are as follows.

As illustrated in Fig. 2, the architecture of the proposed model benefits from three modules that analyze and classify network traffic. A fog node processes the received network traffic on different modules including preprocessing, detection, and countermeasure module, respectively. The preprocessing module replaces string data with numbers and normalizes the data. The detection module analyzes and classifies the characteristics of the recorded traffic into normal or attack. The incoming traffic is sent to the cloud environment to detect the type of attack and update the neural network model located in the cloud layer when the traffic is classified by the detection module as attack. The detection level in the cloud layer allows FP correction in that of the fog layer. In other words, some records which are not regarded as attacks in the fog layer are mistakenly

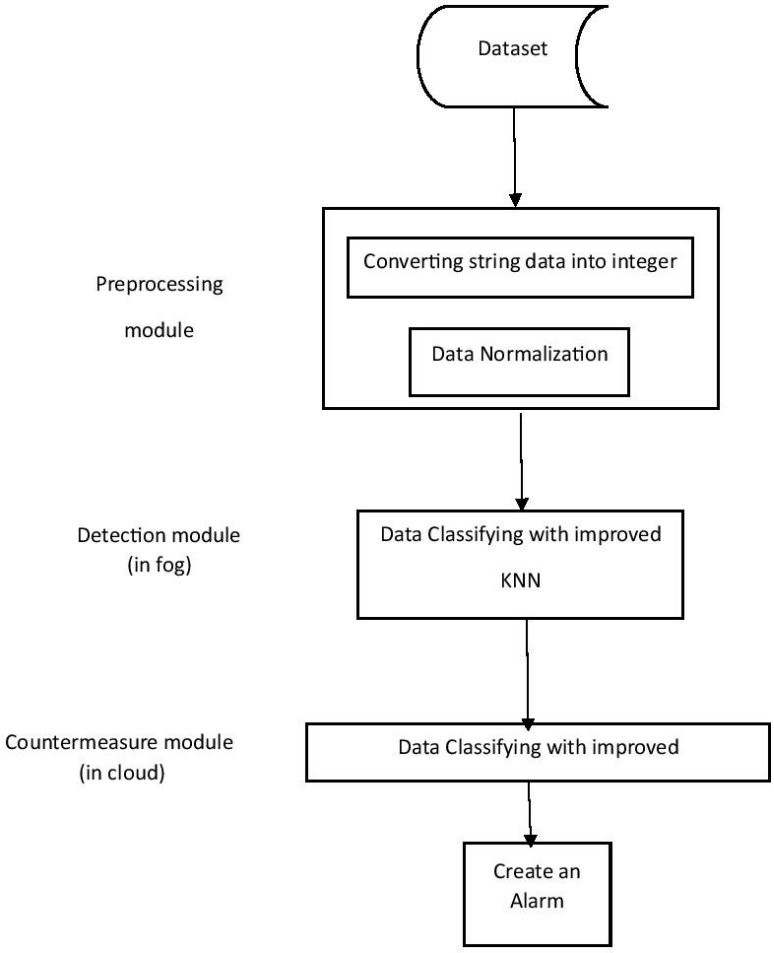

Figure 2. Overall view of the proposed technique.

detected as attacks. The record specifications are sent to the fog layer to update and classify the detection module into the normal category to reduce the FP rate in subsequent detections when such records are detected by the cloud layer as a normal one. Fig. 2 shows the proposed model. The details of the study are as follows.

3. Modules

3.1 Preprocessing module

The preprocessing module converts the string data into an integer and normalizes the whole data to improve the intrusion detection performance. Text features need to be converted into numerical values since KNN classification only uses numerical data.

The preprocessing of each feature is normalized during the second stage. In other words, the features are scaled between 0–1. The range of correct values for the features differs, meaning that some benefit from a large range of integers, while others exhibit values of 0 or 1 or are considered as continuous. Therefore, such features should be normalized. Equation 1 is utilized to normalize the data in the features which need to be normalized.

$$N(m,n) = \frac{C(m,n) - \min(n)}{\max(n) - \min(n)} \tag{1}$$

$\max(n)$: Maximum value related to the nth column
$\min(n)$: Minimum value related to the nth column
$C(m,n)$: Integer related to the mth row and nth column
$N(m,n)$: Normalized value of the mth row of nth feature
The values for all of the signs are in the range of 0–1 after preprocessing, which can be applied.

3.2 Detection and countermeasure modules

Basically, low latency is needed to allow a quick response to reduce the possible damage that an abnormal traffic can create [29]. Thus, an improved KNN model is used in the fog layer to increase the detection speed. The improved KNN algorithm compares only nodes that are close to each other in space and avoids unnecessary computations [32].

The improved algorithm assigns nodes to a spatial network which acts as an index. Finally, the required distances between the nodes are determined by adding the edges corresponding to each node. To this aim, the algorithm is repeated through the grid cells and their corresponding nodes. The distances to all of the nodes in the grid cell ($r = 0$), those related to neighboring grid cells ($r = 1$), and those related to the next grid cells ($r = 2$, and the like) are considered until the nearest node is found for each node in the main grid cell. Such minimum distances are applied to find all of the edges starting from a node in the considered grid cell, which requires examining all of the nodes in the corresponding hypercube grid cells [32].

The μ size in the spatial grid affects the running time. Ideally, a running time of $O(n)$ can be achieved by selecting an appropriate μ, which means keeping the number of grid cells to be regarded at each iteration step constant (determining the nearest neighbor for each node and adding edges). Therefore, the number of nodes in each cell should be independent of the total number of nodes n. The length of μ side equals to $\frac{n}{2p^d}$ when such number should remain constant at the value of c. The aforementioned selection ensures that the average number of grid cells to consider while finding nearest neighbor nodes remains constant regardless of the total number of nodes [32]. In fact, n, d, and p represent the total number of data, the number of data dimensions or features, and the density of the data network, respectively. According to [32], the density value of p network should be equal to 1.6.

Thus, nodes and edges related to each other are identified using algorithm 2 [32]. First, the data are divided into μ^d grid cells, followed by determining the ID of the grid cell that contains such data. Then, the distance between the data and edges is calculated. To this aim, the distance of all of the data in the grid cell is compared to that of the neighboring ones until close data is found for each data related to the main grid cell. Finally, all of the edges in the grid cell are found utilizing the above-mentioned minimum distance. The overall complexity of the algorithm can reduce to $O(n)$ by selecting an appropriate cell size. The acquired value is considered as more appropriate compared to the complexity of the KNN algorithm, which equals $O(n)^2$.

The algorithm can be improved by developing methods to index nodes by a spatial network and determining neighborhoods in the network. Applying the spatial grid indicated before makes the algorithm more efficient compared to the simple one due to its ability to allow the exploitation of the resulting grid position [32].

A multilayer neural network is used to detect the type of attack in the cloud layer. The neural network produces a model based on training data and expresses the problem of IDSs in pattern classification. Stored patterns should be updated continuously. Fig. 3 displays the structure of the proposed method in the classification stage. The records detected as normal in the fog layer (true negative or TN) enter the cloud layer with the normal label and are utilized to learn the neural network in order to improve the accuracy of such network in the cloud layer.

The MLP contains a two-layer feedforward neural network. In other words, a neural network with one hidden and one output layer is applied (the input layer is not counted because such layer acts like a buffer without processing). The structure of the aforementioned two-layer network is regarded as (x_1 10 x_2), in which x_1 represents the number of characteristics for incoming traffic, meaning the existence of x_1, 10, and x_2 nodes in the input, middle, and output layer, respectively. The value of x_1 and x_2 at the IOT32 dataset equal 21 and 15, respectively. Output nodes benefit from binary values, meaning that only one of the output nodes equals 1 and the rest equal 0 for each of the output classes. The obtained training dataset is used to fine-tune the MLP classifier. Here, the trainscg function is utilized as a neural network training function. Finally, the experimental features are introduced to the trained MLP classifier to detect the attacks.

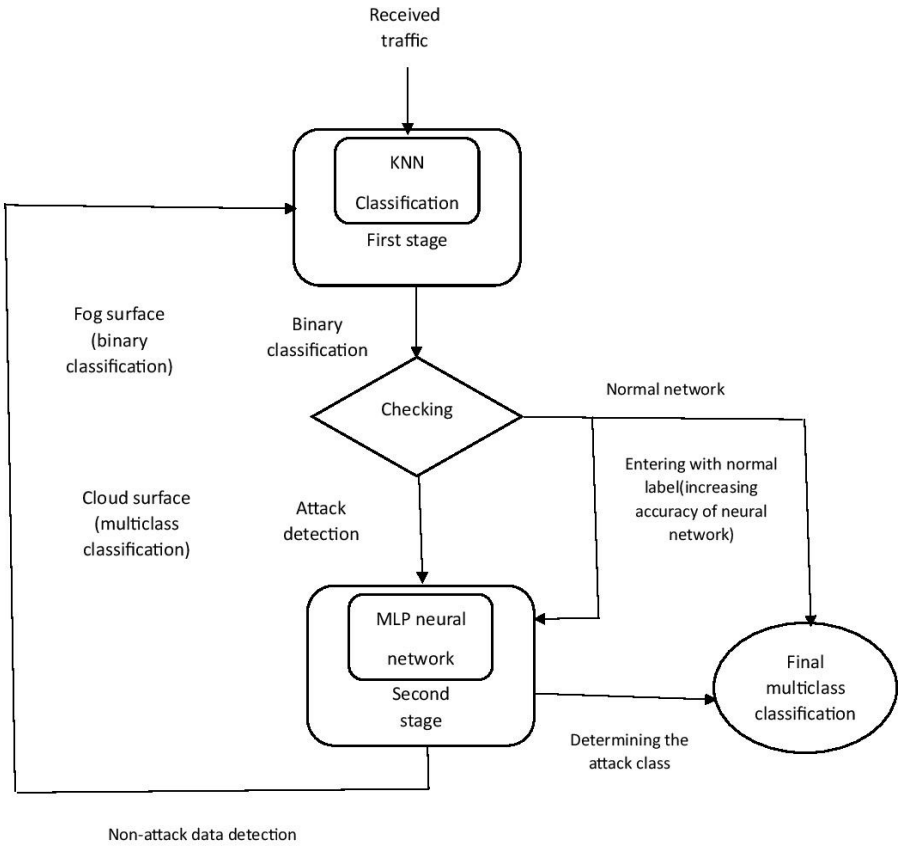

Figure 3. Structure of the proposed method in the classification stage.

4. Evaluation and efficiency of the proposed IDS

Different tests should be conducted based on training samples in order to study the classification model. Table 1 indicates the occurrence of different situations for the categories considering the input dataset for the category with TP (true positive), FP (false positive), TN (true negative), and FN (false negative) values.

Table 1. Clutter matrix for a two-category classification problem.

Record type	Predicated Records		
Actual records	Category type	− Category	+ Category
	− Category	TN	FP
	+ Category	FN	TP

TN: Records which are not considered as attack and are correctly identified as normal.
TP: Records which are regarded as attack and are correctly detected as attack.
FP: Records which are not considered as attack, while are mistakenly detected as attack.
FN: Records which are regarded as attacks, while are mistakenly detected as correct activity.

According to the parameters indicated in the following equations, various evaluation criteria are presented, the most significant of which include the accuracy, precision, recall, and F-measure criteria.
Accuracy is considered as the most significant criterion for determining the efficiency of a classification algorithm. The aforementioned criterion calculates the overall accuracy of a category and indicates what percentage of the entire dataset is correctly classified. Equation 2 demonstrates the method of calculating the accuracy criterion.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{2}$$

The two values of TN and TP are regarded as the most significant values which should be maximized to improve the classification efficiency.
Precision indicates the percentage correctly classified out of all of the categories assigned to the intended category by the classifier. In other words, precision determines the classification accuracy of the category i according to all of the cases where the label i is proposed by the classifier for the examined sample. Equation 3 displays the method of calculating the above-mentioned criterion. The index i means that the parameters should be calculated for each category i.

$$\text{Precision}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \tag{3}$$

Recall indicates the percentage correctly classified among all of the attack categories belonging to the intended category. In other words, recall determines the classification

accuracy of category i according to the total samples with label i. Equation 4 demonstrates the method of calculating the aforementioned criterion.

$$\text{Recall}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \tag{4}$$

It is worth noting that the recall displays the performance of the classifier according to the number of occurrences of the category i, while the precision is based on the accuracy of the category prediction and indicates the reliability of the output of the classifier.

The F-measure is achieved by combining the precision and recall and is applied when no special significance can be attached to such criteria. Equation 5 displays the method of calculating the F-measure.

$$\text{F-measure} = \frac{2 * \text{Precision}_i * \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{5}$$

There are other criteria in real situation such as TPR and FPR, which play a critical role.

The above-mentioned criteria, which focus on the positive classifier, determine its ability to detect the positive category. The TPR determines the accuracy of detecting the positive category, while the FPR expresses the false alarm rate with respect to the negative category. Equation 6 shows the FPR criterion, which expresses the false alarm rate according to the negative category. Equation 7 illustrates the TPR criterion.

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{6}$$

$$\text{TPR} = \frac{\text{TP}}{\text{FN} + \text{TP}} \tag{7}$$

The proposed algorithm is implemented by MATLAB on a system with 64-bit processor specifications, 8GB RAM, and 7-core CPU. Table 2 shows the results of the proposed method for the IOT23 dataset. In addition, the number of training sessions equals 1000 and the intended error rate equals 0.001.

According to [29], the proposed method provides better results than the other ones presented. Comparing the results of the proposed method with those reported during 2022 confirms the aforementioned claim. Such issue is observed clearly by comparing the evaluation factors reflected in the proposed method with other ones presented, especially the last one which was presented by [29].

As observed, C&C-Hakai, C&C-Torii1, and DDoS-Mirai attacks and normal data exhibit the best F-score value. Therefore, such attacks can be detected with better accuracy and precision, while the C&C-HeartBeat-Gagfyt attack exhibits the lowest detection value.

As perceived in Fig. 4, the detection accuracy of the proposed method for the IOT23 dataset equals 99.9 %.

As observed in Table 3, the proposed method is compared with VAE-MLP method [29] in terms of accuracy, recall, precision, F1-score, and TPR factors which were reviewed before. Based on the results, the proposed method performs better in 55 cases considering that there are 15 classes of attacks including the normal mode and 5 criteria are regarded for each class. However, the method offered by [29] exhibits better results than the proposed one in only 20 cases. The highlighted parameter for the proposed method indicates a better result than that offered by [29].

4.1 Time complexity analysis

As indicated in 3.2, the time complexity of the fog layer equals O(n). The time complexity of the neural network used in the cloud layer is calculated as follows. Assume

Table 2. Implementing the proposed method on the IOT23 dataset.

Type of Attack	Accuracy	Recall	Precision	F-measure	TPR	FPR
Benign	99.999 %	100.000 %	100.000 %	100.000 %	99.999 %	0.012 %
C&C-Mirai	99.997 %	100.000 %	99.117 %	99.557 %	99.938 %	0.025 %
DDoS-Mirai	99.990 %	100.000 %	100.000 %	100.000 %	99.954 %	0.018 %
Port-scan-mirai	100.000 %	99.818 %	99.999 %	99.908 %	99.128 %	0.019 %
C&C-Torii1	100.000 %	100.000 %	100.000 %	100.000 %	100.000 %	0.000 %
C&C-Torii2	99.510 %	99.338 %	99.581 %	99.459 %	99.999 %	0.029 %
C&C-Troja	99.382 %	99.712 %	99.369 %	99.540 %	100.000 %	0.000 %
C&C-FileDownload-Troja	99.911 %	98.993 %	99.991 %	99.489 %	99.274 %	0.143 %
FileDownload-Troja	100.000 %	99.829 %	100.000 %	99.914 %	99.992 %	0.346 %
C&C-HeartBeat-Gagfyt	99.378 %	99.995 %	98.342 %	99.162 %	99.927 %	0.000 %
DDoS-Gagfyt	99.592 %	99.462 %	99.605 %	99.533 %	100.000 %	0.017 %
C&C-HeartBeat-Okiru	99.940 %	100.000 %	99.951 %	99.975 %	100.000 %	0.288 %
Okiru	99.975 %	100.000 %	99.983 %	99.991 %	99.957 %	0.000 %
Okiru-Attack	99.913 %	99.990 %	99.957 %	99.973 %	100.000 %	0.015 %
C&C-Hakai	100.000 %	100.000 %	100.000 %	100.000 %	99.958 %	0.000 %

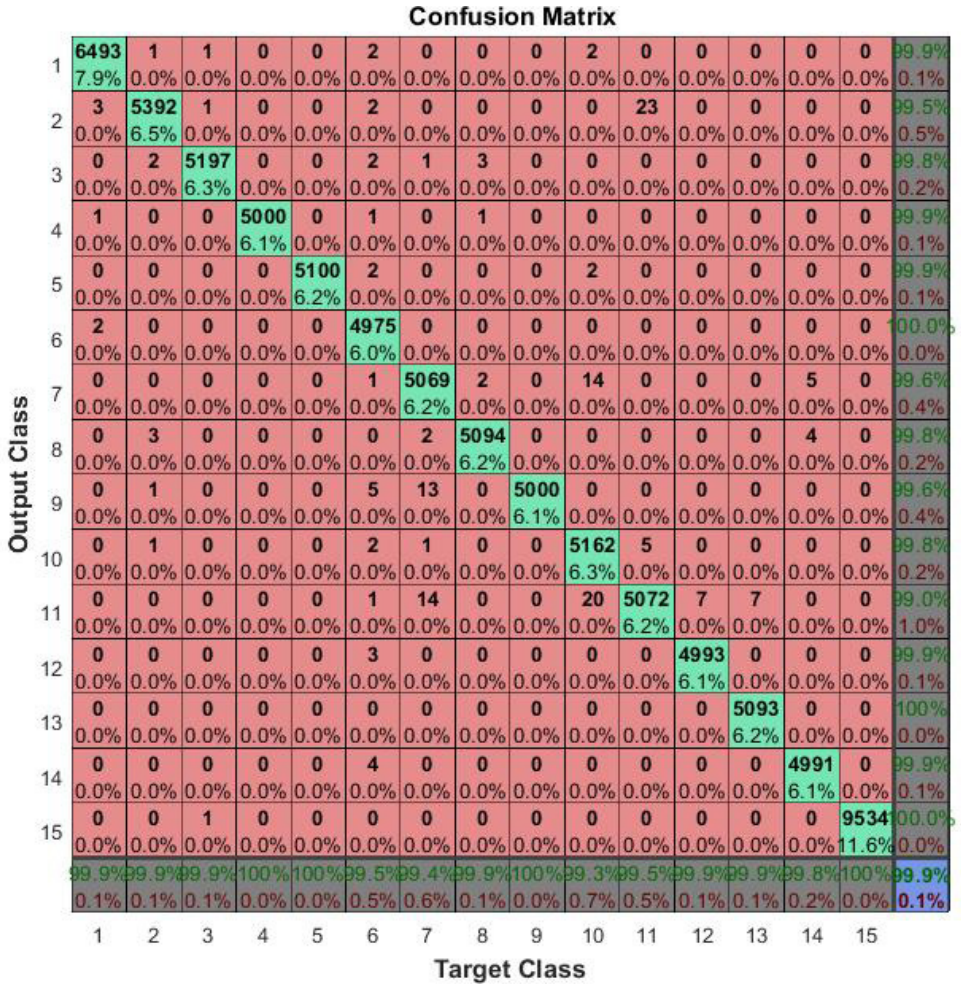

Figure 4. Clutter matrix of the proposed method for the IOT23 dataset.

Table 3. Implementing the proposed method on the IOT23 dataset.

Algorithm	Attack Type	Accuracy	Recall	Precision	F1-score	TPR
VAE-MLP [29]	Benign (Normal)	99.993 %	99.999 %	99.999 %	99.948 %	99.988 %
	C&C-Mirai	99.967 %	100 %	98.967 %	100 %	99.918 %
	DDoS- Mirai	99.990 %	100 %	99.990 %	99.730 %	99.922 %
	Port Scan - Mirai	99.998 %	99.698 %	99.998 %	100 %	98.990 %
	C&C-Torii1	100 %	99.999 %	100 %	99.888 %	100 %
	C&C-Torii2	99.230 %	99.118 %	99.230 %	99.118 %	99.987 %
	C&C-Troja	99.132 %	99.076 %	99.132 %	100 %	99.999 %
	C&C-FileDownload -Troja	99.841 %	98.483 %	99.841 %	99.756 %	98.998 %
	FileDownload-Troja	100 %	99.700 %	100 %	99.990 %	99.978 %
	C&C-HeartBeat-Gagfyt	99.118 %	100 %	98.118 %	98.888 %	99.818 %
	DDoS-Gagfyt	99.342 %	99.122 %	99.342 %	99.999 %	100 %
	C&C-HeartBeat-Okiru	99.912 %	99.998 %	99.912 %	100 %	100 %
	Okiru	99.955 %	100 %	99.955 %	100 %	99.923 %
	Okiru-Attack	99.833 %	99.900 %	99.833 %	99.948 %	99.997 %
	C&C-Hakai	100 %	100 %	100 %	99.930 %	99.900 %
Proposed Algorithm	Benign (Normal)	**99.999 %**	**100.000 %**	**100.000 %**	**100.000 %**	**99.999 %**
	C&C-Mirai	99.938 %	99.557 %	**99.117 %**	**100.000 %**	**99.997 %**
	DDoS- Mirai	99.954 %	**100.000 %**	**100.000 %**	**100.000 %**	**99.990 %**
	Port Scan - Mirai	99.128 %	**99.908 %**	**99.999 %**	99.818 %	**100.000 %**
	C&C-Torii1	**100.000 %**	**100.000 %**	**100.000 %**	**100.000 %**	**100.000 %**
	C&C-Torii2	**99.999 %**	**99.459 %**	**99.581 %**	**99.338 %**	99.510 %
	C&C-Troja	**100.000 %**	**99.540 %**	**99.369 %**	99.712 %	99.382 %
	C&C-FileDownload -Troja	99.274 %	**99.489 %**	**99.991 %**	**98.993 %**	99.911 %
	FileDownload-Troja	99.992 %	**99.914 %**	**100.000 %**	99.829 %	**100.000 %**
	C&C-HeartBeat-Gagfyt	**99.927 %**	99.162 %	**98.342 %**	**99.995 %**	99.378 %
	DDoS-Gagfyt	**100.000 %**	**99.533 %**	**99.605 %**	99.462 %	99.592 %
	C&C-HeartBeat-Okiru	**100.000 %**	99.975 %	**99.951 %**	**100.000 %**	99.940 %
	Okiru	**99.957 %**	99.991 %	**99.983 %**	**100.000 %**	**99.975 %**
	Okiru-Attack	**100.000 %**	**99.973 %**	99.957 %	**99.990 %**	99.913 %
	C&C-Hakai	99.958 %	**100.000 %**	**100.000 %**	**100.000 %**	**100.000 %**

that i, j, and k indicate the number of nodes in the input, hidden, and output layer, respectively. Here, two matrices including W_{ji} and W_{kj} are needed to represent the weights between the above-mentioned layers because three layers are observed. W_{ji} is considered as a matrix with j rows and i columns, and contains the weights which go from layer i to j.
Assume that there are n training records. Equation 8 is utilized to propagate from layer i to j.

$$S_{jn} = W_{ji} * Z_{in} \tag{8}$$

The matrix multiplication operation exhibits a time complexity of $O(j*i*n)$. In addition, the time complexity of the activation function equals to $O(j*n)$.

$$Z_{jn} = f(S_{jn}) \tag{9}$$

Thus, the total time complexity for transferring from the first layer to the second one equals $O(j*i*n)$. $O(k*j*n)$ is used for going to $j \rightarrow k$ applying the same logic. Therefore, the time complexity equals $O(n*i*j*k)$. The final result for time complexity in the cloud layer neural network equals $O(n*t*i*j*k)$ since the algorithm should be repeated t times for training.
Table 4 represents the time complexity for the two layers of fog and cloud in the proposed method and VAE-MLP one presented by [29] for n records.
The study focusing on VAE-MLP method [29] used a five-layer neural network on the cloud level. Thus, the time complexity for the aforementioned method equals $O(n*t*i*j*k*l*m)$. In addition, the time complexity equals $O(n*i*j)$ on the fog level hypothesizing two matrices of $i*j$ dimensions.

Table 4. Comparing the time order of the proposed method and VAE-MLP one presented by [29].

Method	Time complexity for Fog	Time complexity for Cloud
Proposed method	$O(n)$	$O(n*t*i*j*k)$
VAE-MLP [29]	$O(n*i*j)$	$O(n*t*i*j*k*l*m)$

4.2 Detection and countermeasure modules

To conduct the latency analysis, 550 IoT nodes including fog ($N = 50$) and edge devices ($N = 500$) was deployed in a simulated environment. Each edge device is hypothesized to be connected to at least one fog one. The distance between the IoT network and remote cloud eqauls about 500 miles, while that between the IoT device and its local fog nodes equals 10 m. IoT traffic was produced utilizing the Poisson distribution, which is widely applied for modeling network traffic. The number of input packets and their length are modeled as an exponential distribution during the Poisson process. Queuing delay (D_{Queueing}) and packet drop rate are proportional to traffic density. The simulation test takes 200 time samples. The speed of data release equals 2×10^8 m/s.
Transferring device to the cloud uses a theoretical net bit rate capacity of up to 100 and 50 Mbps in the downlink and uplink, respectively [12]. Latency is defined as the average time taken to respond to a service request. Here, the time required to detect a normal or abnormal activity is considered, which can be expressed as follows.

$$D_t = \frac{\sum_{i=1}^{N}(D_{\text{Propagation } i} + D_{\text{Transmission } i} + D_{\text{Queueing } i} + D_{\text{Processing } i})}{N} \tag{10}$$

where i, D, t, and N represent the ith device, delay, time, and total number of active devices, respectively. The propagation delay is regarded as the time taken to transfer a bit of data from the sender to the receiver, which is defined as follows.

$$D_{\text{Propagation}} = d/S$$

where d and S indicate the distance between the two communication parties through the link and transmission speed, respectively. Transmitted data travel over a link in the form of electromagnetic signals at a certain speed determined by the medium through which the data pass.
Transmission delay indicates the time required to send all of the data in a packet to the transmission link.

$$D_{\text{Transmission}} = L/R$$

where L and R represent the packet length and transmission rate, respectively. $D_{\text{Transmission}}$ is determined based on the data size and bandwidth of the transmission link (in bps). D_{Queueing} is considered as the waiting time which the packet needs to stay in the router buffer to be processed. D_{Queueing} is determined by three factors including the rate of data arriving at the router, bandwidth of the router outgoing link, and state of network traffic.
Processing delay ($D_{\text{Processing}}$) is regarded as the time spent by the router to process the packet. Such delay depends on the processing speed of the router. Here, $D_{\text{Processing}}$ and D_{Queuing} are hypothesized to be fixed.
The delay time is compared in three cases.

1. KNN-MLP two-level IDS (proposed method)
2. MLP-MLP two-level IDS
3. Cloud IDS without fog

As perceived in Fig. 5, the IoT network is simulated to test its average delay over a period of 200 times (horizontal axis). In addition, the vertical axis displays the delay time. the proposed method exhibits approximately 6 and 107 % less delay than the MLP-MLP and cloud IDS without fog, respectively. Further, the fluctuation in the two charts stems from the random selection of the records in the dataset, which includes the number of different attacks.
In short, such two-layer IDSs can effectively detect network intrusions while reducing latency.

5. Conclusion

The presented intrusion detection approach based on fog calculations in the cloud environment with two levels of analysis was evaluated with IOT 23 dataset [32]. Experiments were carried- out to evaluate the proposed approach

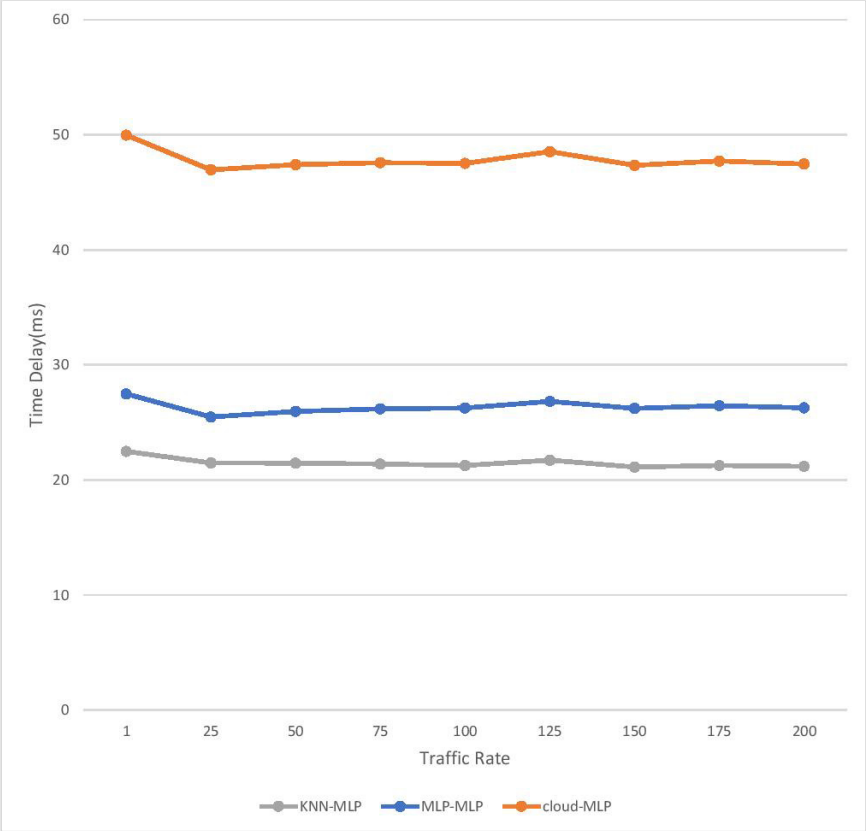

Figure 5. Comparing the implemented delay time for three modes.

with machine learning techniques and state-of-the-art approaches. The proposed approach was able to obtain similar or superior performance in IOT- 23 dataset, demonstrating the robustness of the proposed approach. The performance of a binary analysis with improved KNN in the first detection stage allows for the release of benign traffic in an agile manner and the performance of a more robust inspection, with an MLP neural network approach, in the events that could not be identified as benign. The aforementioned method is improved significantly in terms of time order in the level of fog and clouds. In addition, from an implementation point of view, the delay time of the proposed method is improved compared to that of utilizing MLP neural network in the fog and cloud with less delay than the state with sole cloud. Some solutions can be provided to detect computational efficiency and intrusion detection for a wider range of attacks. We intend to focus in future studies on more cyber-attacks in order to provide a more efficient method in the evaluation of the proposed IDS.

Authors Contributions
All authors have contributed equally to prepare the paper.

Availability of Data and Materials
The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflict of Interests
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C.A. de Souza et al. "**Intrusion detection and prevention in fog based IoT environments: A systematic literature review**.". *Computer Networks*, 214:109154, 2022. DOI: https://doi.org/10.1016/j.comnet.2022.109154.

[2] C.A. de Souza et al. "**Hybrid approach to intrusion detection in fog-based IoT environments**.". *Computer Networks*, 180:107417, 2020. DOI: https://doi.org/10.1016/j.comnet.2020.107417.

[3] V. Chang et al. "**A Survey on Intrusion Detection Systems for Fog and Cloud Computing**.". *Future Internet*, 14(3):89, 2022. DOI: https://doi.org/10.3390/fi14030089.

[4] P. Mell and T. Grance. "***The NIST definition of cloud computing***.". Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, 2011. DOI: https://doi.org/10.6028/NIST.SP.800-145.

[5] M. Satyanarayanan. "**A brief history of cloud offload: A personal journey from odyssey through cyber foraging to cloudlets**.". *GetMobile: Mobile Computing and Communications*, 18(4):19–23, 2015. DOI: https://doi.org/10.1145/2721914.2721921.

[6] F. Bonomi et al. "**Fog computing and its role in the internet of things**.". In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC '12, page 13–16, New York, NY, USA, 2012. Association for Computing Machinery. DOI: https://doi.org/10.1145/2342509.2342513.

[7] E. Marín-Tordera et al. "**Do we all really know what a fog node is? Current trends towards an open definition**.". *Computer Communications*, 109:117–130, 2017. DOI: https://doi.org/10.1016/j.comcom.2017.05.013.

[8] J. Singh, P. Singh, and S.S. Gill. "**Fog computing: A taxonomy, systematic review, current trends and research challenges**.". *Journal of Parallel and Distributed Computing*, 157:56–85, 2021. DOI: https://doi.org/10.1016/j.jpdc.2021.06.005.

[9] K.S. Kiran et al. "**Building a intrusion detection system for IoT environment using machine learning techniques**.". *Procedia Computer Science*, 171:2372–2379, 2020. DOI: https://doi.org/10.1016/j.procs.2020.04.257.

[10] W.H. Hassan. "**Current research on Internet of Things (IoT) security: A survey**.". *Computer Networks*, 148:283–294, 2019. DOI: https://doi.org/10.1016/J.COMNET.2018.11.025.

[11] S. Sicari et al. "**Security, privacy and trust in Internet of Things: The road ahead**.". *Computer Networks*, 76:146–164, 2015. DOI: https://doi.org/10.1016/J.COMNET.2018.11.025.

[12] S. Roy, J. Li, and Y. Bai. "**A Two-layer Fog-Cloud Intrusion Detection Model for IoT Networks**.". *Internet of Things*, 19:100557, 2022. DOI: https://doi.org/10.1016/j.iot.2022.100557.

[13] X. An et al. "**Hypergraph clustering model-based association analysis of DDOS attacks in fog computing intrusion detection system**.". *EURASIP Journal on Wireless Communications and Networking*, 2018 (1):1–9, 2018. DOI: https://doi.org/10.1186/s13638-018-1267-2.

[14] P. Illy et al. "**Securing fog-to-things environment using intrusion detection system based on ensemble learning**.". In *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2019. DOI: https://doi.org/10.48550/arXiv.1901.10933.

[15] J. Pacheco et al. "**Artificial neural networks-based intrusion detection system for internet of things fog nodes**.". *IEEE Access*, 8:73907–73918, 2020. DOI: https://doi.org/10.1109/CISDA.2009.5356528.

[16] B. Sudqi Khater et al. "**A lightweight perceptron-based intrusion detection system for fog computing**.". *Applied Sciences*, 9(1):178, 2019. DOI: https://doi.org/10.3390/app9010178.

[17] M. Almiani et al. "**Deep recurrent neural network for IoT intrusion detection system**.". *Simulation Modelling Practice and Theory*, 101:102031, 2020. DOI: https://doi.org/10.1016/j.simpat.2019.102031.

[18] B.A. N.G. and S. Subramanian. "**Anomaly detection framework for Internet of things traffic using vector convolutional deep learning approach in fog environment**.". *Future Generation Computer Systems*, 113:255–265, 2020. DOI: https://doi.org/10.1016/j.future.2020.07.020.

[19] S. Manimurugan. "**IoT-Fog-Cloud model for anomaly detection using improved Naive Bayes and principal component analysis**.". *Journal of Ambient Intelligence and Humanized Computing*, page 1–10, 2021. DOI: https://doi.org/10.1007/s12652-020-02723-3.

[20] N. Moustafa and J. Slay. "**UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)**.". page 1–6, 2015. DOI: https://doi.org/https://doi.org/10.1109/MilCIS.2015.7348942.

[21] M.S. Elsayed, N.-A. Le-Khac, and A.D. Jurcut. "**InSDN: A novel SDN intrusion dataset**.". *IEEE Access*, 8:165263–165284, 2020. DOI: https://doi.org/10.1109/ACCESS.2020.3022633.

[22] M. Tavallaee et al. "**A detailed analysis of the KDD CUP 99 data set**.". In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, page 1–6. IEEE, 2009. DOI: https://doi.org/10.1109/CISDA.2009.5356528.

[23] "**ADFA IDS Datasets**.". URL https://research.unsw.edu.au/projects/adfa-ids-datasets.

[24] M. Krzysztoń and M. Marks. "**Simulation of watchdog placement for cooperative anomaly detection in bluetooth mesh intrusion detection system**.". *Simulation Modelling Practice and Theory*, 101:102041, 2020. DOI: https://doi.org/10.1016/j.simpat.2019.102041.

[25] M.A. Rahman et al. "**Scalable machine learning-based intrusion detection system for IoT-enabled smart cities**.". *Sustainable Cities and Society*, 61:102324, 2020. DOI: https://doi.org/10.1016/j.scs.2020.102324.

[26] P. Maniriho et al. "**Anomaly-based intrusion detection approach for IoT networks using machine learning**.". In *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, page 303–308. IEEE, 2020. DOI: https://doi.org/10.1109/CENIM51130.2020.9297958.

[27] S.I. Popoola et al. "**Hybrid deep learning for botnet attack detection in the internet-of-things networks**.". *IEEE Internet of Things Journal*, 8(6):4944–4956, 2020. DOI: https://doi.org/10.1109/MilCIS.2015.7348942.

[28] Y. Labiod, A. Amara Korba, and N. Ghoualmi. "**Fog Computing-Based Intrusion Detection Architecture to Protect IoT Networks**.". *Wireless Personal Communications*, 125:231–259, 2022. DOI: https://doi.org/10.1007/s11277-022-09548-7.

[29] C.A. de Souza, C.B. Westphall, and R.B. Machado. "**Two-step ensemble approach for intrusion detection and identification in IoT and fog computing environments**.". *Computers & Electrical Engineering*, 98:107694, 2022. DOI: https://doi.org/10.1016/j.compeleceng.2022.107694.

[30] S.P.K. Gudla et al. "**DI-ADS: a deep intelligent distributed denial of service attack detection scheme for fog-based IoT applications**.". *Mathematical Problems in Engineering*, 2022(1):3747302, 2022. DOI: https://doi.org/10.1155/2022/3747302.

[31] F.-B. Mocnik. "**An improved algorithm for dynamic nearest-neighbour models**.". *Journal of Spatial Science*, 67(3):411–438, 2020. DOI: https://doi.org/10.1080/14498596.2020.1739575.

[32] S. Garcia, A. Parmisano, and M.J. Erquiaga. "**IoT-23: A labeled dataset with malicious and benign IoT network traffic**.". *Zenodo*, 2021. DOI: https://doi.org/10.5281/zenodo.4743746.