Intrusion Detection in The Internet Of Things Based On A Multilayer Combination Of Misuse And Anomaly Detection Systems

Hossein khosravifar[1], Mohammad Ali Jabraeil Jamali*[2], Kambiz Majidzadeh[3], and Mohammad Masdari[3]
[1]Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran
[2] Department of Computer Engineering, Shabestar Branch, Islamic Azad University, Shabestar, Iran
[3] Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran
Email: Mohammad Ali Jabraeil Jamali(Corresponding authors)

Receive Date: 24 November 2024 Revise Date: 15 December 2024 Accept Date: 27 January 2025

Abstract

In light of the rapid and insecure growth of devices connected to the Internet of Things (IoT), intrusion detection systems are recognized as one of the effective security mechanisms in this domain. These systems face significant challenges, including vast amounts of data with numerous features and imbalanced distribution, data imbalance, resource limitations, unknown attack detection, and a high rate of false alarms.This paper introduces a new model for developing an intrusion detection system known as Hybrid Multi-Layer System (HMLS), aimed at reducing false alarms and increasing accuracy in detecting both known and unknown attacks. In the proposed method, a dataset collected from network traffic is preprocessed before being fed into a multi-layer classifier that identifies specific categories of attacks at each layer based on a hybrid intrusion detection framework called Hybrid System of Misuse and Anomaly (HSoMA). Simulation results using the NSL-KDD dataset indicate that the proposed method improves evaluation metrics by 5.49%, 1.09%, and 4.5% in terms of Accuracy, Precision, and False Alarm rates compared to previous works.

Keywords*:* Internet of Things, Intrusion detection, Machin learning, Classifier systems, GMDH neural network

1. Introduction

Ensuring security in IoT is one of the fundamental challenges of this technology[1]. Some ongoing projects aimed at enhancing IoT security include methods that encompass data confidentiality, authentication, access control, privacy preservation, establishing trust between users and devices, and implementing proactive security policies[2]. However, even with these measures in place, IoT networks remain vulnerable to various attacks. Therefore, an additional and complementary defensive approach alongside preventive methods is necessary to detect intrusions by attackers and their malicious activities; intrusion detection systems (IDS) are designed to achieve this objective[2, 3].

The inherent characteristics of the IoT include the presence of heterogeneous and insecure devices that are capable of exchanging information but have limited processing and storage capabilities, resulting in the production of large amounts of diverse and unstructured data [4-6]. Cloud computing has been introduced as a suitable processing model for storing and processing this volume of data; however, due to the increasing demand for real-time applications that are sensitive to delays in IoT, existing issues cannot be resolved solely through cloud computing [7, 8]. Therefore, a fog-based processing model has been proposed as a complement to cloud computing to address these challenges. Fog computing aims to extend cloud services at the network edge so that processing, storage, and communication tasks are brought closer to

end devices [9]. This approach improves aspects such as reducing latency, enhancing mobility, increasing network bandwidth, security, and privacy preservation [10].

Intrusion detection systems deployed at fog nodes can be broadly categorized into two main types: misuse-based and anomaly-based systems [11]. Misuse detection identifies attacks using patterns and signatures that represent these types of attacks, typically resulting in a low rate of false alerts; however, the limitation of this approach is its inability to detect unknown attacks [12, 13]. In contrast, anomaly detection methods create profiles of normal activities and detect deviations from typical behavior, labeling those deviations as intrusions [14]. As a result, anomaly-based intrusion detection systems can identify unknown attacks that misuse-based approaches cannot detect. Nevertheless, one drawback of anomaly detection methods is their high rate of false positives. Therefore, it is essential to design an intrusion detection system that combines the benefits of both approaches while minimizing their respective drawbacks as much as possible.

In this paper, a new structure for intrusion detection, named HMLS and based on fog processing, is introduced to address the challenges of detecting unknown attacks with a low number of false alerts, data imbalance issues, and multi-class problems. In the proposed method, the collected data from network traffic is first preprocessed. Then, the preprocessed data are fed into a multilayer intrusion detection system. The goal of this multilayer system is to transform a multi-class intrusion detection system into several binary classification systems. In each layer of this multilayer system, specific categories of attacks are detected using a proposed combined classifier called HSoMA, which can identify both known and unknown attacks while minimizing false alerts.HSoMA consists of two main phases: 1- misuse-based identification and 2- anomaly-based identification. In the first phase, known attacks are identified first; then normal-labeled training data are divided into subsets that have less diversity in their connection patterns compared to all normal data sets. Subsequently, in the anomaly identification phase for each identified normal-labeled subset, an independent anomaly detection model is utilized because each subset contains more concentrated data; thus its efficiency in creating normal profiles will be higher and consequently more successful in identifying anomalies.

The contributions of this paper are summarized as follows:

- Using multilayer classifiers based on a combination of misuse and anomaly detection methods to overcome problems caused by data imbalance in the dataset and problems related to multi-class classifiers;
- Dividing the dataset with normal labels in each layer into separate subsets using the decision tree algorithm to create a governing order in the data;
- UsingGMDH neural network, for each subset, to detect unknown attacks, increase detection accuracy, and eliminate redundant and ineffective features during automatic training of the intrusion detection system.

The rest of the paper is organized as follows: Section 2 briefly reports the background and related works, and Section 3 discusses and describes the system model and problem statement .After that, Section 4 gives the framework of the proposed method. Section 5 describes the evaluation criteria and simulation results. Finally, Section 6 concludes the study and gives directions for further research.

2. Background and related work

This section discusses some of the main concepts of this article and reviews some of the work done in recent articles related to this area.

2-1. Types of Intrusion Detection Systems

In general, IDSs can be categorized based on different aspects, as shown in Fig. 1. From an architectural perspective, IDSs are divided into three groups: centralized, distributed, and hybrid IDSs [14]. In the centralized category, all processes are carried out within a central system. Nevertheless, in the distributed mode, each device processes packets independently. Furthermore, the hybrid IDS is regarded as the combination of centralized and distributed categories. This way, it benefits from the merits of centralized and distributed IDs and avoids their shortcomings.

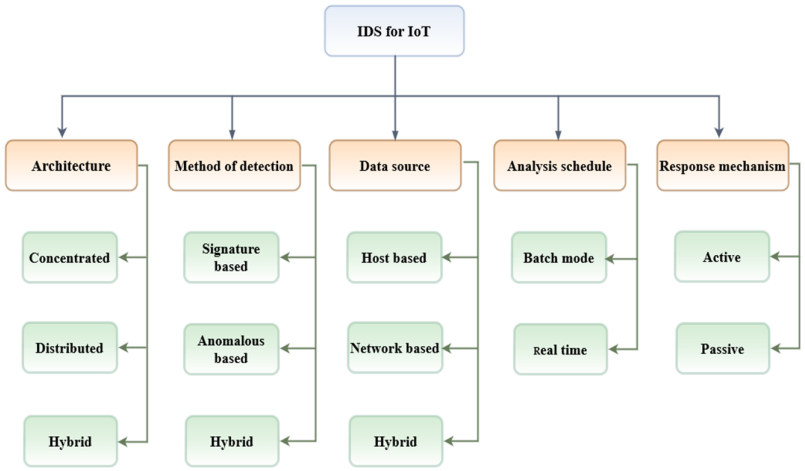

Fig. 1. The classification of IDSs in IoT

Furthermore, from the perspective of intrusion detection and intrusion tackling techniques, IDSs are classified into three groups [14]. In anomaly methods, normal behavior patterns are defined for the system, and behaviors other than those defined are identified as attacks. The drawback of these methods is the generation of an overly high of false alarms. Signature-based methods use the identified and known intrusion patterns for detecting intrusions. Hence, these methods are not effective and efficient in detecting unknownattacks. Hybrid methods apply the concepts of signature-based and anomaly-based detections to maximize the merits of both categories and minimize their demerits.

Concerning the source on which intrusion detection is carried out, IDSs can be divided into three groups [15]. In host-based intrusion detection systems, information is gathered based on all the available events on the host (event letters - operating system audit trails, etc.). In network-based intrusion detection, network traffic is the main information source. In the hybrid method, both sources are used for gathering information.

The scheduler is regarded as an issue that should be considered in data analysis. Data analysis may be done in batch mode or real-time mode. In Batch mode analysis, information related to a time period is collected and sent to the data analyzer. In the real-time method, the information source is given to the analyzer as each event occurs within a short time period [14].

Also, based on the response mechanism, IDSs are divided into active and inactive IDSs [16]. Active IDSs take a specific action to prevent intrusion after an intrusion is detected. For example, network traffic is blocked by using the firewall. In inactive IDSs, IDS has no responsibility for direct reaction in intrusion prevention. That is, it only informs the one who is in charge of network security of the occurrence of an intrusion.

2-2. Fog-based processing

In recent years, a variety of architectures have emerged for fog-based processing patterns, primarily based on a foundational 3-layer structure [17]. The terminal layer is the closest to end devices and the physical environment, encompassing various IoT devices such as sensors, smartphones, smart vehicles, and more. These geographically distributed devices are tasked with collecting data from their surroundings, which is then transmitted to the next layer for processing and storage. The fog layer operates at the network edge and consists of multiple fog nodes situated between end devices and cloud centers. Depending on the application requirements, these fog nodes are capable of processing, monitoring, sharing, and storing data collected from IoT devices. Additionally, they facilitate communication with the cloud to provide necessary processing resources and reliable data storage. Finally, the cloud layer comprises efficient service providers along with a robust storage infrastructure. This layer offers substantial processing capabilities and extensive storage capacity to handle intensive computational tasks while managing large quantities of data effectively.

2-3. Related work

By applying an unsupervised deep learning algorithm, Mirsky et al.[18] proposed a method called Kitsune, which is based on network analysis for detecting an anomaly. The most significant feature of this method is the online detection of patterns. This method has a performance that is comparable to that of offline anomaly detection. The proponents of this method claim that it is practical and economical. The kitsune method includes a set of small neural networks trained to imitate (recreate) network traffic patterns. The operation of this method significantly improves time. Another merit of the Kitsune method is that thanks to its online processing, lightweight, scalability among several IoT devices, and faster execution time, it can be potentially applied on networks with small memory. Nonetheless, Kitsune depends on external libraries for identifying and analyzing raw packets. Also, in this method, anomaly detection is merely based on RMSE. Hence, it is prone to generating false positive alarms during network operation.

Quamar et al. [19] developed and put forth a multiple-vector deep learning-based multiple-vector DDoS attack detection system in a software-defined networking environment. SDN provides flexibility for programming network devices in line with different purposes. Indeed, it eliminates the requirement for specific hardware in each network device as an independent decision unit. They implemented their system as a network application on top of the SDN controller. Also, they applied deep learning to reduce the features of a large set of features received from network traffic. Notable merits of this method are its high intrusion detection speed, automatic extraction of features from packet headers, high accuracy, and low false positive alarm rate. Some demerits of this method are highly time-consuming training time and the loss of some features due to pre-processing.

Rahman et al. [20] proposed a neuro-fuzzy-based IDS for detecting intrusion spread of physical layers (PHY) and controlling access (MAC) in IoT. In this method, NF (ANFIS) is applied for intrusion detection. It collects data by supervising network operations, and the available data in the database is dynamically updated. This method includes some stages such as the network behavior analysis stage, feature identification stage, and feature extraction and selection with a classification procedure. The proposed algorithm, in this method, receives network operation data as an input and indicates attack occurrence probability as an output in IoT. The advantages of this method are high reliability in an IoT secure environment and high accuracy. However, the disadvantage of this method is its high computational overhead.

Hodo et al. [21] developed a supervised learning-based IDS for identifying DDoS attacks. An artificial neural networkis regarded as this method's underlying rationale and framework. They used a multi-layer perceptron neural network for categorizing normal and abnormal behaviors. In the proposed algorithm, three layers were used for training data. Each neural network neuron uses a unipolar Sigmoid transfer function. The notable merits of this method are productivity in incomplete data resources, identification of suspicious known events, high detection accuracy, low false positive alarm rate, and high accurate positive alarm rate. Nevertheless, this method is based on probability estimation and needs more time to achieve acceptable efficiency and efficacy.

Lopez Martin et al. [22] proposed an IDS called ID-CVAE, which is appropriate for IoT. This method is based on deep learning of the variational autoencoder with a particular architecture that enters attack labels into decoder layers. Anomaly-based machine learning is regarded as this method's underlying procedure and approach. Hence, applying a deviation-based method can classify specific traffic samples with attack labels. The most noticeable aspect of this method is that it can reconstruct and recreate features. That is, it can recover the lost features from incomplete training datasets. Attack features and the labels of the attack class are regarded as two inputs of this method. There are several models for creating a classification with VAE, and a specific learning module is required for each model. Each learning module uses only particular related samples of a label. ID-CVAE generates a particular model with a learning module that uses all the learning data regardless of the related labels. Low complexity, computation delay reduction, high detection accuracy, and response time reduction are regarded as the merits of this method. On the other hand, the high false alarm rate and the utilization of several resources in the training stage are the shortcomings of this method.

Diro et al. [23] developed a new deep learning-based distributed IDS for identifying attacks in IoT/Fog. Since Fog nodes are close to IoT smart infrastructures, they are used for training and keeping IDS at the edge of fognetworks. The authorsused the NSL-KDD dataset and the parameters of accuracy, detection rate, false alarm rate, etc., to evaluate their proposed method. Using the parameters mentioned above, they tried to indicate the efficiency of deep learning models compared to shallow models. Evaluation results showed that the proposed distributed IDS outperforms deep learning-based centralized IDS. High detection accuracy, being online, and a low false alarm coefficient are the merits of this method. On the other hand, high training time and the utilization of several

resources in training are considered to be the demerits of this method.

Pamukov et al. [24] developed a new classifying algorithm for IoT intrusion detection systems called negative selection neural network(NSNN). It includes two distinct layers. Firstly, the negative selection algorithm(NSA), one of the artificial intelligence algorithms of security systems, is used for producing a training set via normal network behavior knowledge. Based on this data, a simple neural network is trained for real classification. This multi-layer method can eliminate the computation complexity of training. Furthermore, adding a negativeselection layer provides the opportunity for a neural network to be trained only based on the normal behavior of the network regardless of the need for attack data. NSNN was trained and experimented with based onthe NSL-KDD dataset. Thanks to its low false positive alarm rate, it can detect unknown attacks with high accuracy. The algorithm of this method is not online; also, given the scalability and the issue of security gaps, NSA is not appropriate for normal/abnormal classification on a large scale.

Liu et al. [17] proposed an IDS for IoT using a suppressed fuzzy clustering algorithm (SFC) and principal component analysis (PCA). This algorithm first categorizes data into high-risk and low-risk data, identified by high and low frequencies. Detection frequency is adjusted and tuned according to SFC and PCA algorithms. Due to the increasing data transmission size in IoT, feature extraction might be highly time-consuming. PCA algorithm may reduce the number of variables and eliminate the features with little importance. After extracting the feature vector by the PCA algorithm, classification is done. This method has better compatibility, high accuracy, and a low false alarm rate. However, the efficiency reduction of the algorithm, along with the increased data volume, is regarded as the disadvantage of this method.

Li et al. [18] introduced an IDS based on multiple classifications that use a CNN neural network as the base classifier. In this method, the features related NSL-KDD dataset are divided into four groups based on the correlation among the features. Then, the data related to each category of features is processed by the CNN neural network. Next, the results of these four base classifiers are combined. The merits of this method are high accuracy in detecting different attacks and the reduction of false alarm rate. Nevertheless, the need for more training data, complex model structure, high computational complexity, and poor interpretability should be considered as this method's drawbacks.

Chatterjee and Hanawal[19] introduced an architecture for intrusion detection called PHEC, which is based on combining the results of several simple classifiers. In this architecture, input data is pre-processed, and dimensions are reduced. Then, two base classifiers, based on KNN and Random Forest algorithms, are separately trained using the extracted features. Next, the outputs of these two base classifiers are aggregated according to the aggregated averaging method. Then, the result is compared with a predetermined threshold value. After that, the final label is specified. A high detection rate and low false detection rate are regarded as the merits of PHEC. Nevertheless, the application of a predetermined threshold, in some cases with more noise, maybe over-fitting, is regarded as the drawback of it.

Ying Zhong et al. [25] developed and proposed a multi-level anomaly detection framework called HELAD. In the first

stage, the incremental statistical algorithm *Damped* was used for extracting features from the network traffic. Then, in the second stage, the Autoencoder neural network trains itself with a few labeled data. Next, it begins to tag the anomaly of the data. In the next stage, the data labeled with the anomaly is used for training the LSTM neural network. Finally, the weighting method is used for detecting attacks.

A fog-computing-based two-layer architecture was introduced in [26] to detect IoT intrusion. The first layer uses a two-class and two-level intrusion detection method based on DNN neural network and KNN. It can detect intrusion with optimal accuracy. In this method, the task of the second layer and detection of the attack type have not been elaborated on and discussed.

A deep-learning-based and binary algorithm-based hybrid IDS was proposed in [27] which the binary algorithm operates as an optimizer. Furthermore, the binary genetic algorithm (BGA), binary bat algorithm (BBA), and binary gravitational search algorithm(BGSA) were applied to enhance the attack detection rate.

Y.Wu et al. [28] developed an IDS with a common training model called JSAE-FSVM. This training model from SAE (stacked autoencoder) is used to reduce feature dimension, and random Fourier features are used as a kernel estimation technique for producing a random feature space. Then, linear SVM is used to detect the types of attacks. High accuracy in detecting known attacks, high efficiency and efficacy in large-scale datasets, and little training time are considered to be the merits of JSAE-FSVM. On the other hand, low accuracy in detecting unknown attacks is regarded as the demerit of this method.

A hybrid two-layer method, namely DLHA, was introduced in [29] for detecting intrusions. In the first layer, Naïve Bayse was used for detecting Prob and Dos attacks and intrusions. In the second layer, the SVM algorithm with RBF kernel was used for detecting U2R and R2L intrusions. Moreover, PCA and ICFS algorithms were applied to select important features. High F-score, the high detection rate for R2L and U2R attacks, and being real-time are the advantages of this method. However, the high false positive alarm rate is the demerit of this method.

Multiple classifier systems for detecting anomalies were put forth in [30], which were intended to be applied on the web. This architecture used random forest, gradient boosting machine, and XGBoost as the base classifiers. Also, a generalized model, GLM, was used to integrate the results of the base classifiers. High detection and reduction of false alarm rates are regarded as the merits of this method. Nonetheless, failure to detect and investigate attack types is this method's shortcoming.

A hybrid two-phase IDS, called SAAE-DNN, was proposed in [31]. In the first phase of this method, the stacked autoencoder algorithm with a hidden layer, namely the attention mechanism, was used for automatically extracting important features. Next, the extracted features are used as DNN classier input in the next stage for detecting intrusion. High accuracy in detecting the intrusion is the merit of this method, and the need for more training data is the demerit of this method.

An anomaly-based two-phase IDS was proposed in [32]. In the first phase, thanks to integrating PSO, ACO, and GA algorithms, the dimensions of features were reduced. In the second phase, a two-layer classifier was used to detect the anomaly. Therotationforest algorithm was used in the first layer, and the Baggingalgorithm was used in the second

layer. Then, the results obtained from the two layers were integrated by means of the Majority Voting method for detecting an anomaly. High accuracy and high detection rate are the advantages of this method. However, the high false positive alarm rate and the failure to examine the types of attacks are the disadvantages of this method.

Zhou et al. [33] put forth a hybrid IDS framework called CBS-BA-Ensemble, based on feature selection and combining different learning techniques. In the first stage of this framework, a heuristic algorithm, namely CFS-BA, was used to reduce the dimensions of features. Then, multiple classifiers were used to detect intrusions. This method used Forest-PA, RE, and G4.5 algorithms as base classifiers. Furthermore, the voting technique was used for combining base classifiers. The application of multiple classifiers and high efficiency are the advantages of this method; nevertheless, a high false positive alarm rate is considered the demerit of this method.

A deep-learning-based intrusion detection method, called DAE-DNN, was proposed in [34]. In this method, the deep autoencoder is first used to reduce feature dimension. Then, the DNN neural network was used to identify suspicious behavior.

A two-layer IDS, called DSN, was proposed in [35]. In this method's first and second layers, a 4-level ID3 algorithm and DNN neural network with five hidden layers were used, respectively, for classifying different types of intrusions. Table 1 gives a synopsis of the merits and demerits of the related works, which were briefly reviewed above.

Table 1 Synopsis of the related works

Researcher	Year	Method	Strategy of validation	Merits	**Demerits**
Y. Mirsky et al. [18]	2018	Using an unsupervised deep learning algorithm (Kitsune) based on network analysis to detect the anomaly	Simulation	• Online processing • Lightweight and scalable among several IoT devices • Faster execution time	• Identifying and analyzing raw packets depends on external libraries. • Anomaly detection is merely based on RMSE. Hence, it is prone to generating false positive alarms during network operation.
S. Rahman et al. [20]	2016	a Neuro-Fuzzy (NF)-based IDS for detecting intrusion spread of physical layers (PHY) and controlling access (MAC) in IoT	CAIDA	• High reliability • High accuracy	• High computational overhead
N.Quamar et al. [19]	2016	multiple-vector deep learning-based multiple-vector DDoS attack detection system in software-defined networking (SDN) environment	Simulation	• high intrusion detection speed • automatic extraction of features from packet headers • high accuracy • low false positive alarm rate	• Long training time
E. Hodo et al. [21]	2016	supervised learning-based IDS for identifying DDoS attacksusing ANN	simulation	• Online usage • High accuracy • identification of suspicious known events • productivity in incomplete data resources • high detection accuracy	• This method is based on probability estimation and needs more time to achieve acceptable efficiency and efficacy.

				• low false positive alarm rate	
M. Lopez et al. [22]	2017	deep learning-based IDS of the variational autoencoder with a particular architecture.	NSL-KDD	• Low complexity • computation delay reduction • high detection accuracy • response time reduction	• high false alarm rate • utilization of several resources in the training stage
A. A. Diro and N. Chilamkurti [23]	2017	Deeplearning-based distributed IDS for identifying attacks in IoT/Fog	NSL-KDD	• High detection accuracy, • being online • low false alarm rate	• high training time • utilization of several resources in training
M.E.Pamukov et al. [24]	2018	Using the combination of a negative selection algorithm and a simple neural network for intrusion detection	KDD NSL	• low false positive alarm rate • high accuracy • detecting unknown attacks with high accuracy	• It is not online • Given the scalability and the issue of security gaps, the NSA is not appropriate for self/non-self classification on a large scope
L. Liu et al. [36]	2018	Using fuzzy clustering algorithm (SFC) and principal component analysis(PCA) for	Simulation	• high accuracy • low false alarm rate	• reduction of the efficiency of the algorithm along with the increased data volume
Y. Li et al. [37]	2019	Using several CNN neural networks as a base classifier and combining their results	NSL_KDD	• high accuracy • reduction of false alarm rate	• need more training data • complex model structure • high computational complexity • poor interpretability
S. Chatterjeeand MK. Hanawal [38]	2022	Using two basic classifiers based on KNN and Random Forest algorithms and then combining their results based on the averaging method	NSL_KDD	• High detection rate • The low false detection rate	• the application of a predetermined threshold • in some cases, with more noise, it may be over-fitting
Ying Zhong et al. [25]	2021	a multi-level anomaly detection framework with autoencoder and LSTM classifiers for detecting intrusion	IDS2017	• Good accuracy rate	• Need more training data • complex model structure • high computational complexity • poor interpretability
CA.Souza et al. [26]	2020	two-layer architecture for detecting intrusion based on DNN neural network and KNN	NSL_KDD	• Detection distributed in the fog • high accuracy and recall rate	• failure in detecting intrusion type
K. Atefi et al. [27]	2020	Combining DNN and BA algorithms for detecting intrusion	CICIDS2017	• Optimization scheme	• Few experiments were performed for evaluation
Y.Wu et al. [28]	2020	Using SAE and kernel approximation algorithms to reduce the number of features and then using SVM to classify attacks	NSL_KDD	• High accuracy in detecting known intrusions • high efficiency in large-scale datasets • little training time	• low accuracy in detecting unknown intrusions
T. Wisanwanichthan and M. Thammawichai	2021	Using PCA and ICFS algorithms for selecting important features, then applying Naïve Bayes	NSL_KDD	• High F-score • high detection rate for R2L and U2R attacks	• high false positive alarm rate

[29]		algorithm for detecting Prob and Dos attacks and SVM algorithm with RBFKernel for detecting U2R and R2L attacks.		• being real-time	
B. A. Tama [30]	2020	fusion of the results of random forest, gradient boosting machine, and XGBoost basic classifications using a generalized model called GLM	NSL_KDD CICIDS2017 UNSW-NB15	• High detection • reduction of false alarm rate	• failure in detecting and investigating attack type
CH.Tang [31]	2020	Using the stacked autoencoder algorithm with a hidden layer, namely the attention mechanism for extracting important features. Next, applying DNN is classier for detecting intrusion.	NSL_KDD	• The high intrusion detection rate	• Need and dependence on more training data
B. Adhi Tama [32]	2020	using PSO, ACO, and GA algorithms to reduce the number of features. Next,Combining the results ofBagging and *Rotation forest* algorithms Using the Majority voting technique to detect abnormality	NSL_KDD UNSW-NB15	• High accuracy and intrusion detection rate	• high false positive alarm rate • the failure to examine the types of intrusion
Y. Zhou et al. [33]	2020	Using a heuristic algorithm called CFS-BA to reduce the features' dimensions, then combining the results of the G4.5, RF, and ForestPA base classifications using the voting technique.	NSL_KDD CICIDS-2017 AWID	• application of multiple classifiers • high efficiency	• high false positive alarm rate
Y. Novaria Kunang [34]	2022	Using an autoencoder to reduce feature space, then applying a DNN neural network for intrusion detection.	NSL_KDD	• high accuracy	• Needs more training data
Y.Tang [35]	2023	Using a two-layer system based on ID3 and DNN algorithms for IDS.	NSL_KDD	• High accuracy • Reduction of false alarm rate	• Needs more time and resources

3. System model and Problem Statement

In this section, the system model is introduced, and then the problem statement is specified.

3-1. System model

Fig.2 illustrates the IoT network, which is monitored by the fog-based processing model and uses the proposed HMLSas the intrusion detection method. The model of the architecture system is three-layered, and the proposed IDS system has been designed to operate in the fog layer. Depending on the application, some fog nodes are required to be equipped with HMLS. In the final layer, the required data for detecting the intrusion is gathered by the sensors and is offered to the respective fog node.Each fog node has an HMLS intrusion detection system that detects attacks locally without interacting with the cloud, thus avoiding latency and consuming less bandwidth. Only after detecting an event as a new attackdoes it send the information of that attack to the cloud. As shown in Fig. 2, since the training process is time-consuming and needs a machine with high computational capability, it is carried out in the cloud layer without interacting with the fog nodes. Communication between the cloud and fog nodes is established when the trained model is transmitted to each of them. Also, in this layer, the events identified by the fog nodes as new attacks are stored and used while updating the training model.

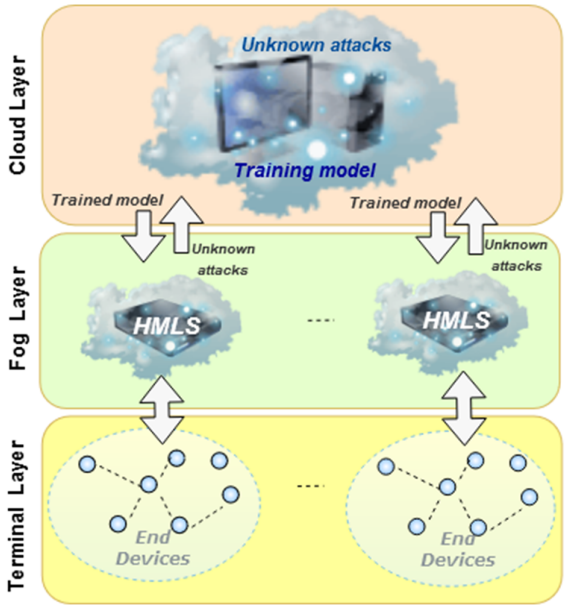

Fig. 2. System model

3-2. Problem statement

Suppose we have a dataset containing n samples. Each sample is characterized by a feature vector $x_i \in R^d$ and a class label$y_i \in \{C1,C2,\ldots,Ck\}$. Where d represents the number of features and k represents the number of different classes.

For each class C_j, we define a model $h_j(x)$whose task is to predict whether the input x belongs to class C_j or not. This model is a binary label function of the form$h_j: R^d \rightarrow \{0,1\}$, which is defined as Eq.(1):

$$h_j(x) = \begin{cases} 1 & x \in C_j \\ 0 & x! \in C_j \end{cases} \qquad (1)$$

These models are trained using a decision function.After training all the models, the classification step is performed as follows:

• For a new input x, for each class C_j, the output $h_j(x)$ is calculated.

• Finally, the input x will belong to the class for which the model h_j produced the highest probability(Eq. (2)):

$$\hat{y} = \underset{j}{argmax}\, h_j(x) \qquad (2)$$

4. HMLS proposed framework

In this section, the structure and components of the proposed HMLS method are introduced. The HMLS method is a hybrid intrusion detection system based on fog and network traffic analysis that uses a multilayer classifier to detect types of attacks. In this framework, first, a series of preprocessing operations are applied to the data set collected from network traffic to prepare the data set. Then, the preprocessed data set is entered as input into a multilayer classifier, in each layer a new hybrid intrusion detection system called HSoMA is used to detect types of attacks. The number of layers depends on the number of classes or attacks in the data set. The HMLS framework is shown in Fig.3, and its components will be described in detail below.

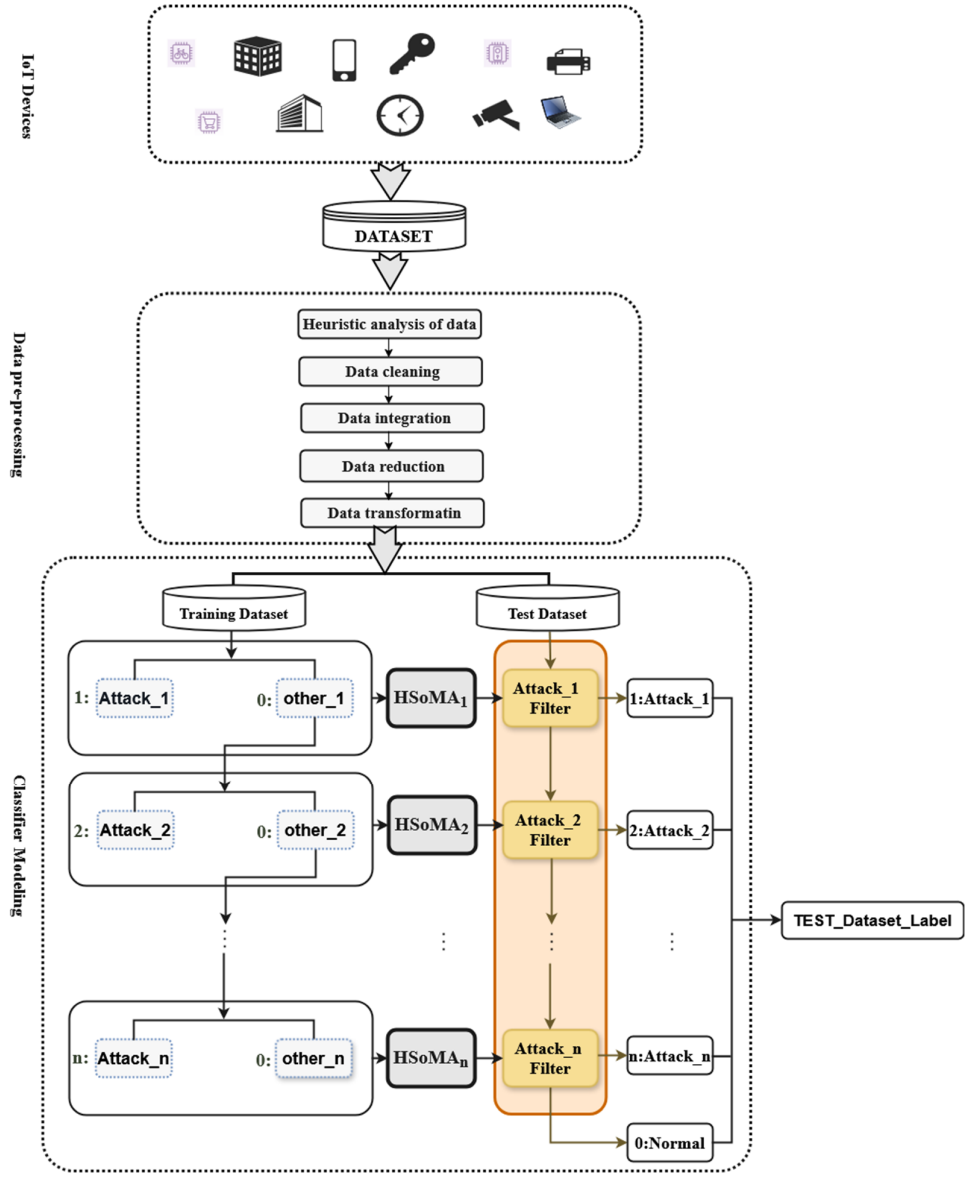

Fig. 3. The architecture framework of the proposed HMLS method

4-1. Dataset

NSL-KDD is a recognized dataset for intrusion detection systems (IDS), updated and revised in 2009 as the latest version of the original kddcup99 dataset [39]. It retains valuable and challenging characteristics from kddcup99 while eliminating redundant and extraneous records, optimizing the sample size, and ensuring diversity among the selected samples. This approach addresses various issues and limitations that were present in the original dataset. NSL-KDD serves as an important resource for evaluating the effectiveness and performance of methods, cybersecurity algorithms, and IDS technologies. The records within this dataset are categorized into normal and attack classes. According to Table 2, 39 types of attacks have been organized into four main categories: DoS, Probing, R2L, and U2R; thus, this dataset is considered a five-class classification problem. It

comprises 43 features; with the first 41 features pertaining to network communication data. The 42nd feature corresponds to labeling each record based on its attack type or normal state where zero denotes normal behavior and labels 1 through 4 indicate DoS, Probing, R2L, and U2R attacks respectively. Additionally, the final feature reflects the difficulty level of IDS but is not factored into analyses concerning this feature.

Table 2 Attack types in the NSL-KDD dataset

Label	Category	Attack type
1	DoS	back, land, neptune, pod, smurf, teardrop, apache2, mailbomb, processtable, udpstorm
2	Prob	portsweep, satan, ipsweep, nmap, mscan, saint
3	R2L	warezclient, warezmaster, spy, multihop, phf, ftp_write, guess_passwd, imap, xsnoop, xlock, worm, snmpguess, snmpgetattack, sendmail, named
4	U2R	rootkit, perl, loadmodule, buffer_overflow, httptunnel, ps, sqlattack, xterm

The statistical description related to the NSL-KDD datasets is given in Table 3.

Table 3 The statistical description of the NSL-KDD dataset

Class	NSL-KDD		
	KDDTrain^{+}	**KDDTest^{+}**	**KDDTest^{-21}**
Normal	67343	9711	2152
DoS	45927	7458	4342
Prob	11656	2421	2402
R2L	995	2754	2754
U2R	52	200	200
Attacks	**58630**	**12833**	**9698**
Total	**125973**	**22544**	**11850**

In this article, the KDDTrain+ dataset was utilized to train and validate the proposed method, while KDDTest+ served as the test dataset for model evaluation. Out of the 39 available attacks, 22 were included in the training process. Additionally, 17 other attacks were incorporated as unknown attacks in the test datasets. As a result, detecting these types of attacks poses a greater challenge for IDS.

4-2. Data pre-processing

The data preprocessing stage is a crucial and necessary step for implementing data mining techniques. Data from the real world may lack the quality required to initiate data mining, potentially compromising the quality of the results; thus, executing the preparation and transformation stage becomes essential. In this phase, exploratory data analysis is first conducted, which is an approach to analyzing a dataset in order to understand its main characteristics. Next, data cleaning is performed to address missing values and noise management issues. If the data has been collected from various sources with different formats and shapes, it becomes necessary to integrate these datasets. To reduce computational costs and processing time, if not all available data needs to be utilized, some excess data will be discarded during the dimensionality reduction process. Data transformation also involves processing information into a suitable format for applying mining algorithms. In the NSL-KDD dataset, out of 41 features related to network communications, there are 7 symbolic features that are not numeric and need to be transformed into numeric form so they

can be understood by machine learning algorithms [40]. Additionally, all numerical features will be normalized in order to eliminate scale differences among them since failure to do so may disrupt algorithm training due to varying input scales. Thus, each initial value such as x within the range between Min_{old} and Max_{old} will be converted into a new value x' within a new range between Min_{new} and Max_{new} according to Eq. (3); in this paper, the new range has been set from -1 to +1.

$$x' = (x - Min_{old})\frac{Max_{new} - Min_{new}}{Max_{old} - Min_{old}} + Min_{new} \quad (3)$$

4-3. Proposed multilayer classifier

One of the proven points in the field of classification is that usually, a few two-class classifiers provide more accurate classification than a single multi-class classifier [41]. From this point, it can be concluded that the more the number of predefined attacks for an intrusion detection system, the lower the accuracy of attack detection and the higher the complexity of the system. In addition, in multi-class classification, it is difficult to detect some attacks due to the imbalance of data in the dataset [42]. Given that the nature of intrusion detection datasets is multi-class due to the diversity of attacks; therefore, in the proposed method, a multilayer classification is used for the dataset.

Initially, the input dataset to this classifier is divided into two parts. One part consists of the labeled training dataset which is used to train the learning model. The other part is the unlabeled test dataset which is used to evaluate the model. It is assumed that there are n attack labels in the training dataset, each layer contains data samples from the training dataset in which the samples related to the i^{th} attack are labeled with label i and the rest with label zero; so that the samples related to the $(i-1)^{th}$ attack are removed from it. Then the i^{th} machine learning model, denoted by $HSOMA_i$, is trained by the generated dataset. The goal of the $HSOMA_i$ model is to detect attacks related to the i^{th} label as much as possible. In the testing and evaluation phase, the test dataset and the trained models $HSoMA_1$, $HSoMA_2$,…, and $HSoMA_n$ are used one by one to label each sample from that dataset. So in the first layer, attacks related to type 1 are detected and labeled, in the second layer, attacks related to type 2, etc., and finally, samples that are not labeled as attack labels are recognized as normal data with a zero label.

The working principles of HSoMA are presented in Fig.4. HSoMA consists of two phases: 1- Misuse detection and 2- Anomaly detection. In the abuse detection phase, first known attacks are detected and then the normal labeled training data is divided into subsets whose connection patterns have less diversity than the entire normal data. Then, in the anomaly detection phase, a separate anomaly detection model is used for each subset identified with the normal label, and since each subset contains more concentrated data, its efficiency in creating more normal profiles and, as a result, anomaly detection will be more successful. In this paper, the C4.5 decision tree is used for misuse detection and the GMDH neural network is used in the anomaly detection phase for the categories that are identified as normal.

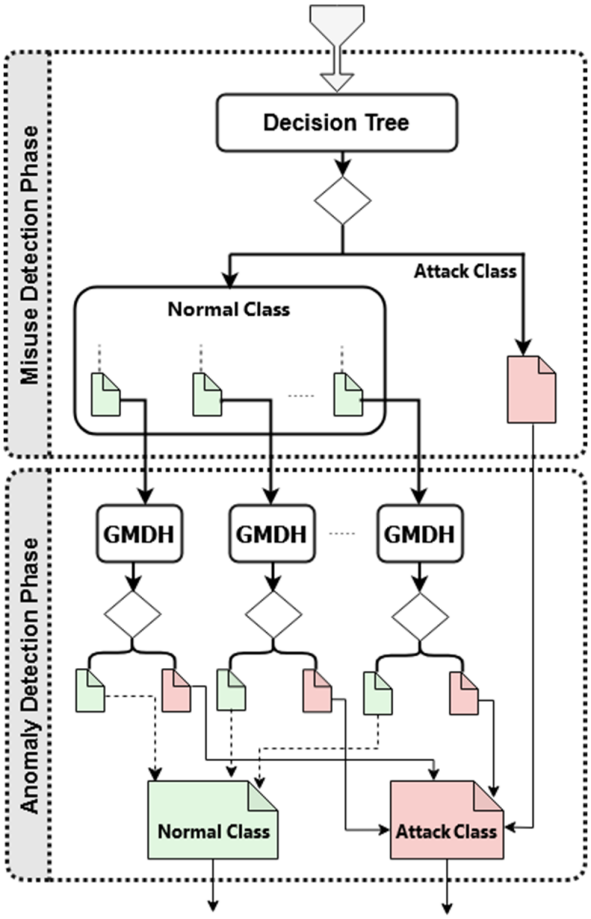

Fig. 4. Internal structure HSoMA

Misuse Detection Phase: After data preprocessing, the dataset enters Phase 1. In this phase, a C4.5 decision tree is used. In this learning stage, the order of the preprocessed data is identified according to the data mining method that the decision tree creates, and the generated model is transferred to the next phase for evaluation. Since dividing the training data into separate subsets based on different rules is a time-consuming process, creating excessively small sets leads to excessively slow operation. On the other hand, the limitation and excessive lack of dispersion of the neural network inputs in the training phase leads to a decrease in its generalization power; therefore, after training the decision tree in Phase 1, the tree is pruned; to the point where only 5 leaves with normal labels remain in the remaining leaves.

Anomaly Detection Phase: In this phase, for each of the remaining leaves with the normal label in phase 1, separate GMDHs are used, the input data to each of which was equal to the data that had reached the leaf corresponding to that neural network according to the classification performed by the decision tree. To obtain the data in each of the leaves, for all leaves with the normal label, their constituent conditions are obtained, and then, based on those conditions, the training data set is divided into separate input subsets, for each of which separate neural networks are used. This increases the accuracy of the network due to the use of a more homogeneous data set, and the anomaly detection system is

better able to identify anomalies that do not follow the normal pattern due to its familiarity with normal patterns that have less dispersion.

GMDH algorithm: In some cases, neural networks have limitations, such as the need for lots of input data for network training and the lack of sufficient relations among the input and output variables. The drawback of statistical models and neural networks in predicting and presenting an optimal model motivated generating GMDH. It is a data-based self-organizing smart method for statistical network training [43]. The purpose of designing GMDH neural networks was to prevent network growth and divergence. Also, it was intended to regulate the form of the network structure into one or more numerical parameters in such a way that as the numerical value of these parameters changes, the network structure changes. In this algorithm, complex models are evolved and gradually established according to multiple data's initial input and output. There is no theoretical background knowledge about the operation manner of data by the researchers in this domain. Also, the designer had a minor contribution in determining effective input features, the number of layers, neurons in the hidden layer, and optimal model structure. Hence, it is the network itself that automatically and gradually updates and evolves its structure. Network parameters are trained according to the Least Squares Estimation Approach. The schema of the GMDH Network, along with a view of neuron structure, is given in Fig. 5.

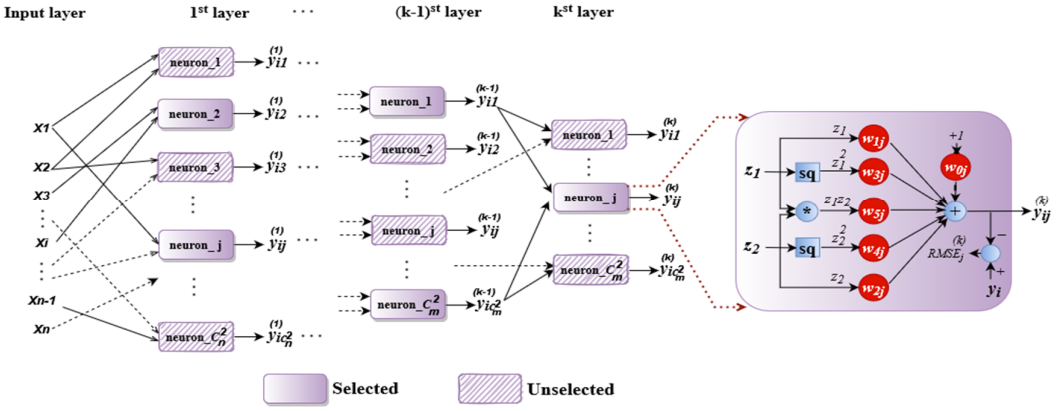

Fig. 5. The structure of the GMDH neural network

The first step in constructing a multi-layer GMDH network is to select the input variables of the problem. All the variables likely to affect the respective issue may be selected. In the first layer, all the binary compounds of the input variables are constructed, and each pair of the variables enters a neuron. The number of neurons in each layer is obtained through Eq. (4), where n denotes the number of input variables.

$$\binom{n}{2} = \frac{n(n-1)}{2} \qquad (4)$$

As shown in Fig. (5), *sq* and * refer to the square and the product, respectively. *z1* and *z2* indicate the input pair related to each neuron. y_i denotes the real output, and $y_{ij}(k)$ refers to the output of the j^{th} neuron from the k^{th} layer regarding the input of the ith sample. Each neuron in the GMDH structure executes a non-linear function of the inputs. Given the dominant nature of the problem in the form of first-degree and second-degree non-linear functions, hyperbolic sine functions, etc., this function can be selected. This paper used a

second-degree non-linear function in the form of Eq. (5) as the driving function or transfer function in each neuron.

$$y_{ij}^{(k)} = w_{0j} + w_{1j}z_1 + w_{2j}z_2 + w_{3j}{z_1}^2 + w_{4j}{z_2}^2 + w_{5j}z_1z_2 \quad (5)$$

Sextet coefficients of w_{0j} to w_{5j} related to the jth neuron are computed via the least square approach [43]. These coefficients should be computed so that the Root Mean Square Error(RMSE) should be the minimum amount between the computational and real outputs. This criterion for the output of the j^{th} neuron should be in the form of Eq. (6).

$$RMSE_j^{(k)} = \sqrt{\frac{\sum_{i=1}^{N}(y_i - y_{ij}^{(k)})^2}{N}} \quad (6)$$

In this equation, N refers to the number of input samples and m in j$\epsilon\{1,2,3,\ldots,C_m^2\}$ indicates the number of selected neurons in the previous layer. The threshold criterion is aimed at determining network structure and selecting neurons of each layer in the form of Eq. (7).

$$RMSE_t^{(k)} = \alpha\, RMSE_{min}^{(k)} + (1 - \alpha)RMSE_{max}^{(k)} \quad (7)$$

In this equation, 0<=α<=1 indicates selection pressure. The higher the α value, the higher the selection pressure. On the other hand, the lower the α value, the lower the protection pressure on the neurons. The parameters $RMSE_{min}^{(k)}$ and $RMSE_{max}^{(k)}$ indicates the minimum and maximum root mean square error of the neurons in the k^{th} layer. The computed *RMSE* criterion for all the neurons is compared with the specified threshold value in each layer. If the related *RMSE* value of each neuron is greater than the threshold value, the respective neuron will be eliminated. Otherwise, it will be used for generating the next layer.

4-4. Evaluation criteria

The confusion matrix is the simplest method for measuring and investigating a classification problem's efficiency. It is a 2-dimensional *n*n* matrix in which n denotes the number of available labels in the dataset. One dimension of this matrix is concerned with real labels, and the next is related to the labels predicted by the classifier system. Table 4 gives the confusion matrix for a 5-class dataset in which *s(i, i)* indicates the number of records with the real label class_i, which the algorithm has detected accurately as class_i. *s(i, j)* denotes the number of records with the real label class_ithat have been falsely detected as class_ j.

Table 4 Confusion matrix of a 5-class dataset

Actual class \ *Predicted class*	**Class_0**	**Class_1**	**Class_2**	**Class_3**	**Class_4**
Class_0	s(0,0)	s(0,1)	s(0,2)	s(0,3)	s(0,4)
Class_1	s(1,0)	s(1,1)	s(1,2)	s(1,3)	s(1,4)
Class_2	s(2,0)	s(2,1)	s(2,2)	s(2,3)	s(2,4)
Class_3	s(3,0)	s(3,1)	s(3,2)	s(3,3)	s(3,4)
Class_4	s(4,0)	s(4,1)	s(4,2)	s(4,3)	s(4,4)

We applied five criteria to investigate the proposed method's performance: Recall (detection rate), False alarm rate, F-score, Accuracy,and Precision. These criteria were measured and taken into consideration by using Eqs. (8) to (12) and the confusion matrix.

$$Accuracy_{class\_i} = \frac{s(i,i) + \sum_{j=0}^{n-1}\sum_{l=0}^{n-1} s(j,l)}{N} \quad | \ j \neq i \ , l \neq i \tag{8}$$

$$Precision_{class\_i} = \frac{s(i,i)}{s(i,i) + \sum_{j=0}^{n-1} s(j,i)} \quad | \ j \neq i \tag{9}$$

$$Recall_{class\_i} = \frac{s(i,i)}{s(i,i) + \sum_{j=0}^{n-1} s(i,j)} \quad | \ j \neq i \tag{10}$$

$$Fals\ alarm_{class\_i} = \frac{\sum_{j=0}^{n-1} s(j,i)}{\sum_{j=0}^{n-1} s(j,i) + \sum_{j=0}^{n-1}\sum_{l=0}^{n-1} s(j,l)} \quad | \ j \neq i \ , l \neq i \tag{11}$$

$$F - score_{class\_i} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{12}$$

5. Simulation results

The proposed framework was simulated and run on the NSL-KDD dataset in MATLAB 2019b on a system with Intel(R) Core(TM) i7-3720QM CPU @ 2.60GHz 16 GB RAM specifications. In the implementation, the KDDTrain+ dataset was considered as the model training set and KDDTest+ as the test set. Fig.6 shows the confusion matrix related to the performance of the proposed system. In these matrices, the rows represent the actual states and the columns represent the performance of the proposed multilayer classifier.

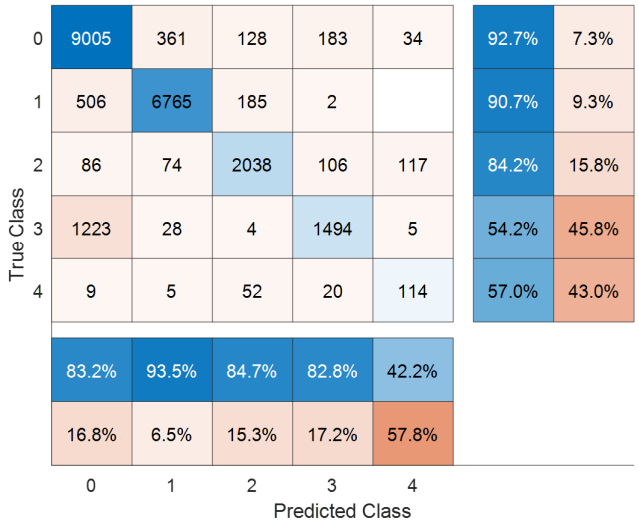

Fig. 6. Confusion matrix related to the performance of HMLS

Considering this confusion matrix, the evaluation criteria of the proposed method were calculated and the results are given in Table 5.

Table 5 Evaluation criteria related to the performance of the proposed method

Label	Accuracy	Recall	Precision	False alarm	F-score
0:Normal	88.78%	92.72%	83.15%	14.21%	87.67%
1:Dos	94.85%	90.70%	93.52%	12.09%	92.08%
2:Probe	96.66%	84.18%	84.66%	1.83%	84.41%
3:R2L	93.03%	54.24%	82.77%	1.57%	65.53%
4:U2L	98.92%	57.00%	42.22%	0.06%	48.50%

5-1. Discussion and review

As can be seen in Table 5, in the proposed method, the highest detection accuracy with a value of 98.92% is related to U2L attacks, and the lowest detection accuracy is related to the normal class. In terms of detection rate, the proposed method was able to have the highest efficiency in detecting normal classes with a value of92.72% and the lowest efficiency in detecting R2L attacks. Also, in terms of accuracy, the highest and lowest accuracy are related to DoS and U2L attacks, respectively. In terms of false alarm rate, the proposed method was able to detect U2L attacks with a value of 0.06% with the lowest alarm rate and normal classes with a value of 14.21% with the highest alarm rate.

Next, to further evaluate the proposed approach presented in this paper, it was compared with several methods proposed in recent papers that have presented their simulations on the NSL-KDD dataset. Table 6 compares the performance of each of these methods based on the specified evaluation criteria.

Table 6 Comparison of evaluation criteria of the proposed model and other methods proposed in the paper

Method	Dataset	Recall(%)	Precision(%)	F1-score(%)	False alarm(%)	Accuracy(%)
Multi-CNN [37]	KDDTest+	81.33	82.51	79.64	10.45	81.33
DAE-DNN [34]	KDDTest+	83.30	86.02	82.04	-	83.30
DLHA[29]	KDDTest+	-	88.17	**90.57**	11.82	88.96
TSE-IDS[32]	KDDTest+	**86.80**	-	-	11.70	85.80
SAAE-DNN[31]	KDDTest+	67.89	87.28	76.37	-	82.14
HMLS	KDDTest+	84.96	**89.264**	75.38	**5.952**	**94.45**

From the results obtained, it can be concluded that the proposed system, in general, considering all parameters, has good performance compared to other methods presented in the recent articles mentioned above, and also provides acceptable Accuracy and false alarm rates.

6. Conclusion and future work

Multilayer classifiers work by converting a multi-class classification problem into multiple binary classification problems using the one-versus-all technique. In this method, for each class, a binary classifier is trained that discriminates that class against all other classes. Breaking down the complex multi-class problem into simpler problems, allows each classifier to focus on one class specifically and to specialize in distinguishing between that class and the rest, increasing the overall accuracy of the system. This method also typically runs much faster than other approaches (such as one-versus-one) due to its simpler structure and lower computational resource requirements.Hybrid intrusion detection systems that use both misuse-based and anomaly-based methods are efficient because they combine the strengths of both approaches. Misuse-based systems are very accurate in detecting attacks that have

known patterns and provide accurate and reliable warnings. On the other hand, anomaly-based systems can detect new and unknown attacks that do not match the defined patterns. By combining these two methods, hybrid systems can detect both known and unknown attacks with higher accuracy and coverage, and as a result, significantly increase the overall level of network security. Simulation results showed that the proposed system, on the one hand, detected types of attacks with high accuracy and low false alarms, and on the other hand, it eliminated the problems caused by data imbalance in the dataset. The proposed model is highly efficient compared to the methods presented in the recent papers reviewed, considering all parameters.In future work, to increase the speed of intrusion detection, the dimensions of the feature space can be reduced using a variety of effective feature selection methods. The structure proposed in this paper can also be tested and investigated with a variety of classification algorithms and deep neural networks with different datasets.

References

[1] T. Oh, "Blockchain-Enabled Security Enhancement for IoT Networks: Integrating LEACH Algorithm and Distributed Ledger Technology," Journal of Machine and Computing, 2025.

[2] A. M. Banaamah and I. Ahmad, "Intrusion Detection in IoT Using Deep Learning," Sensors, vol. 22, no. 21, p. 8417, 2022.

[3] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," Journal of Information Security and Applications, vol. 50, p. 102419, 2020/02/01/ 2020.

[4] L. Kane, J. Chen, R. Thomas, V. Liu, and M. McKague, "Security and Performance in IoT: A Balancing Act," IEEE Access, vol. PP, pp. 1-1, 07/06 2020.

[5] P. Williams, P. Rojas, and M. Bayoumi, "Security Taxonomy in IoT – A Survey," in 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), 2019, pp. 560-565.

[6] M. Hasan, M. Islam, I. Islam, and M. M. A. Hashem, "Attack and Anomaly Detection in IoT Sensors in IoT Sites Using Machine Learning Approaches," p. 100059, 05/20 2019.

[7] H. Lin, Q. Xue, J. Feng, and D. Bai, "Internet of things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine," Digital Communications and Networks, vol. 9, no. 1, pp. 111-124, 2023/02/01/ 2023.

[8] N. F. Syed, M. Ge, and Z. Baig, "Fog-cloud based intrusion detection system using Recurrent Neural Networks and feature selection for IoT networks," Computer Networks, vol. 225, p. 109662, 2023/04/01/ 2023.

[9] L. Yi, M. Yin, and M. Darbandi, "A deep and systematic review of the intrusion detection systems in the fog environment," Transactions on Emerging Telecommunications Technologies, vol. 34, no. 1, p. e4632, 2023.

[10] K. Kethineni and G. Pradeepini, "Intrusion detection in internet of things-based smart farming using hybrid deep learning framework," Cluster Computing, 2023/06/03 2023.

[11] I. Alrashdi, A. Alqazzaz, E. Aloufi, R. Alharthi, M. Zohdy, and H. Ming, "AD-IoT: Anomaly Detection of IoT Cyberattacks in Smart City Using Machine Learning," in 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), 2019, pp. 0305-0310.

[12] C. A. de Souza, C. B. Westphall, R. B. Machado, J. B. M. Sobral, and G. d. S. Vieira, "Hybrid approach to intrusion detection in fog-based IoT environments," Computer Networks, vol. 180, p. 107417, 2020/10/24/ 2020.

[13] O. AbuAlghanam, H. Alazzam, E. a. Alhenawi, M. Qatawneh, and O. Adwan, "Fusion-based anomaly detection system using modified isolation forest for internet of things," Journal of Ambient Intelligence and Humanized Computing, vol. 14, no. 1, pp. 131-145, 2023/01/01 2023.

[14] S. Hajiheidari, K. Wakil, M. Badri, and N. J. Navimipour, "Intrusion detection systems in the Internet of things: A comprehensive investigation," Computer Networks, vol. 160, pp. 165-191, 2019/09/04/ 2019.

[15] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in Cloud,"

Journal of Network and Computer Applications, vol. 36, no. 1, pp. 42-57, 2013/01/01/ 2013.
[16] R. G. Bace and P. Mell, "NIST Special Publication on Intrusion Detection Systems," 2001.
[17] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," Journal of Network and Computer Applications, vol. 98, pp. 27-42, 2017/11/15/ 2017.
[18] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. 2018.
[19] Q. Niyaz, W. Sun, and A. Y. Javaid, "A Deep Learning Based DDoS Detection System in Software-Defined Networking (SDN)," EAI Endorsed Trans. Security Safety, vol. 4, p. e2, 2017.
[20] S. Rahman, S. A. Mamun, M. U. Ahmed, and M. S. Kaiser, "PHY/MAC layer attack detection system using neuro-fuzzy algorithm for IoT network," in 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016, pp. 2531-2536.
[21] E. Hodo et al., "Threat analysis of IoT networks using artificial neural network intrusion detection system," in 2016 International Symposium on Networks, Computers and Communications (ISNCC), 2016, pp. 1-6.
[22] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Conditional Variational Autoencoder for Prediction and Feature Recovery Applied to Intrusion Detection in IoT," Sensors, vol. 17, no. 9, p. 1967, 2017.
[23] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," Future Generation Computer Systems, vol. 82, pp. 761-768, 2018/05/01/ 2018.
[24] M. E. Pamukov, V. K. Poulkov, and V. A. Shterev, "Negative Selection and Neural Network Based Algorithm for Intrusion Detection in IoT," in 2018 41st International Conference on Telecommunications and Signal Processing (TSP), 2018, pp. 1-5.
[25] Y. Zhong et al., "HELAD: A novel network anomaly detection model based on heterogeneous ensemble learning," Computer Networks, vol. 169, p. 107049, 2020/03/14/ 2020.
[26] C. A. d. Souza, C. B. Westphall, R. B. Machado, J. B. M. Sobral, and G. d. S. Vieira, "Hybrid approach to intrusion detection in fog-based IoT environments," Comput. Networks, vol. 180, p. 107417, 2020.
[27] K. Atefi, H. Hashim, and T. Khodadadi, A Hybrid Anomaly Classification with Deep Learning (DL) and Binary Algorithms (BA) as Optimizer in the Intrusion Detection System (IDS). 2020, pp. 29-34.
[28] Y. Wu, W. Lee, X. Gong, and H. Wang, "A Hybrid Intrusion Detection Model Combining SAE with Kernel Approximation in Internet of Things," Sensors, vol. 20, p. 5710, 10/08 2020.
[29] T. Wisanwanichthan and M. Thammawichai, "A Double-Layered Hybrid Approach for Network Intrusion Detection System Using Combined Naive Bayes and SVM," IEEE Access, vol. PP, pp. 1-1, 10/06 2021.
[30] B. A. Tama, L. Nkenyereye, S. M. R. Islam, and K. S. Kwak, "An Enhanced Anomaly Detection in Web Traffic Using a Stack of Classifier Ensemble," IEEE Access, vol. 8, pp. 24120-24134, 2020.
[31] C. Tang, N. Luktarhan, and Y. Zhao, "SAAE-DNN: Deep Learning Method on Intrusion Detection," Symmetry, vol. 12, no. 10, p. 1695, 2020.
[32] B. Adhi Tama, M. Comuzzi, and K. H. Rhee, "TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-based Intrusion Detection System," IEEE Access, vol. 7, 07/11 2019.
[33] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," Computer Networks, vol. 174, p. 107247, 2020/06/19/ 2020.
[34] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Attack classification of an intrusion detection system using deep learning and hyperparameter optimization," Journal of Information Security and Applications, vol. 58, p. 102804, 2021/05/01/ 2021.
[35] Y. Tang, L. Gu, and L. Wang, "Deep Stacking Network for Intrusion Detection," Sensors, vol. 22, p. 25, 12/22 2021.
[36] L. Liu, B. Xu, X. Zhang, and X. Wu, "An intrusion detection method for internet of things based on suppressed fuzzy clustering," EURASIP Journal on Wireless Communications and Networking, vol. 2018, 05/09 2018.
[37] Y. Li et al., "Robust detection for network intrusion of industrial IoT based on multi-CNN fusion," Measurement, vol. 154, p. 107450, 2020/03/15/ 2020.
[38] S. Chatterjee and M. K. Hanawal, "Federated Learning for Intrusion Detection in IoT

Security: A Hybrid Ensemble Approach,"p. arXiv:2106.15349Accessed on: June 01, 2021Available: https://ui.adsabs.harvard.edu/abs/2021arXiv210615349C

[39] S. D. Bay, D. F. Kibler, M. J. Pazzani, and P. Smyth, "The UCI KDD archive of large data sets for data mining research and experimentation," SIGKDD Explor., vol. 2, pp. 81-85, 2000.

[40] S. Naseer et al., "Enhanced Network Anomaly Detection Based on Deep Neural Networks," IEEE Access, vol. 6, pp. 48231-48246, 2018.

[41] T. Hwang, T.-J. Lee, and Y.-J. Lee, A three-tier IDS via data mining approach. 2007, pp. 1-6.

[42] H. Yao, Q. Wang, L. Wang, P. Zhang, M. Li, and Y. Liu, "An Intrusion Detection Framework Based on Hybrid Multi-Level Data Mining," International Journal of Parallel Programming, vol. 47, no. 4, pp. 740-758, 2019/08/01 2019.

[43] S. J. Farlow, "The GMDH Algorithm of Ivakhnenko," The American Statistician, vol. 35, no. 4, pp. 210-215, 1981.