Received: October 10, 2022. Revised: November 22, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

An Efficient Intrusion Detection Approach Using Ensemble Deep Learning models for IoT

Hany Mohamed[1]* Ahmed Hamza[1] Hesham Hefny[1]

[1]*Computer Science Department, Faculty of Graduate Studies for Statistical Research,*
Cairo University, Egypt
* Corresponding author's Email: enghany41@gmail.com

Abstract: The internet of things (IoT) has gained great importance due to its applicability in various daily life applications and its flexible and scalable framework. The wide and spreading use of IoT in the last few years has attracted intruders, who were able to take advantage of the vulnerabilities of any IoT framework due to the absence of robust security protocols. This discourages current and probable investors. Out-of-date intrusion detection models are mainly developed to support information technology systems using built-in, predefined patterns or highly imbalanced datasets. Over the past decade, deep learning models have outperformed traditional machine learning models in attack detection tasks. The biggest challenge in detecting zero-day attacks is determining the best deep-learning classifier. Numerous research initiatives have combined ensemble learning to improve performance, avoid overfitting, and minimize errors. In this work, to address this research gap, we propose a new enhanced meta-learning ensemble deep learning model based on stacking that combines the baseline deep learning models using two tiers of meta-classifiers. Then, conducting several experiments on two recent huge-size different IoT benchmark datasets to evaluate the performance of the proposed model. Different baseline deep learning classifiers are trained for each dataset, and their performance is compared to the proposed ensemble model. The findings show that the proposed ensemble model significantly enhances the classification accuracy of baseline deep-learning models.

Keywords: IoT intrusion detection, Ensemble deep learning approaches.

1. Introduction

Nowadays, internet of things (IoT) can connect a huge number of devices and home appliances (TVs, washing machines, air conditioning, and even more industrial machines) to the internet. Even more, in the last few years, IoT has been widely used in various aspects of life and many areas affecting human life, including healthcare, water supply, traffic monitoring, electrical grid industries, vehicle driving technology, smart cities, and lastly 5G mobile wireless technology. Fig. 1 shows the applications and main domains of IoT in real life.

The growing number of users and services in IoT networks poses a severe threat to the security of IoT systems. The combination of IoT systems and smart environments makes smart objects operational. However, IoT security flaws pose a serious threat in vital smart environments utilized in the medical and industrial sectors. Applications and services in IoT smart environments without strong security systems will be in danger. Information security in IoT systems needs more research to address these concerns because confidentiality, integrity, and availability are three key security concepts for applications and services in IoT smart environments. For instance, security and privacy issues with IoT smart homes cut across all IoT architecture layers.

Security susceptibility of IoT communication protocols is one of many angles through which researchers examine the security concerns posed by the IoT [1]. This paper focuses on security issues faced by IoT systems based on the "IEEE" definition and the overall IoT architecture. This is achieved by focusing on producing a novel intrusion detection system for the IoT paradigm, regardless of the specific protocol.

DOI: 10.22266/ijies2023.0228.31

Received: October 10, 2022. Revised: November 22, 2022.

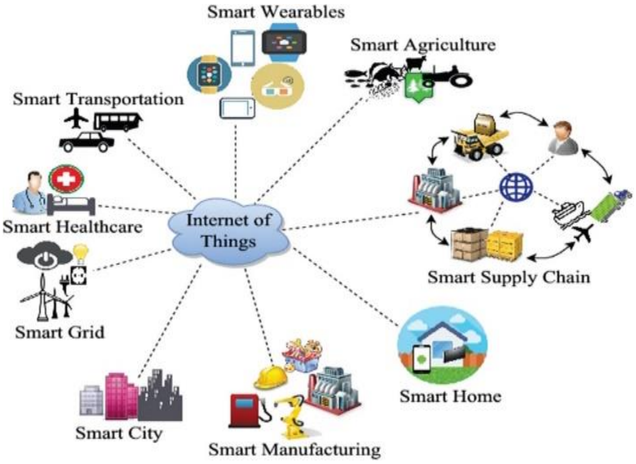


Figure. 1 Internet of things (IoT) domains and applications

IoT security concerns are a result of security problems that develop at various IoT layers. Challenges include physical harm, hardware failure, and power restrictions. On the physical layer, IoT devices are resource constrained and could be in an unprotected environment to physical damage. The network layer is responsible for information transport between application layer processes and IoT devices; denial-of-services (DoS) attacks can risk the availability of network services. The application is vulnerable to exploits of software errors, application protocol weaknesses, and permissions [2].

Within the rapid and tremendous development of the IoT and its penetration into various aspects of human life, security issues of smart systems became an important mission. Existing research exposes the danger posed by cyberattacks. In 2015, intruders remotely bargained for information about energy companies in Ukraine causing temporarily disturbing the electricity supply to Ukrainians; thus, 225 thousand people were affected because of the lower quality of the security mechanism [1].

Researchers are insistent on introducing high security in large-scale networks with good service quality. The interconnection of existing networks, and their applications, long-drawn-out, more complex networks for exchanging critical data. In addition, recent studies and reports, made by the international data corporation (IDC), published in March 2022, show that by 2025, the data generated by IoT devices will reach 73 zettabytes. A great invasion of data opens up a huge amount of possible attacks [3].

Intrusion detection systems (IDSs) are typically created to detect threats, attacks, or suspicious activities made over the network flows, by investigating network traffic flow or the exact supposed environment. Although cybersecurity researchers have proposed several IDS algorithms, most of them are based on traditional mechanisms such as encryption and authentication, procedures to protect wirelessly connected devices. Hence, the existing IDS requires a huge amount of advanced work to achieve high-performance security and authentication.

Intrusion detection techniques are mainly divided into two types: host-based intrusion detection systems (HIDS) and network-based intrusion detection systems (NIDS). HIDS is a system that monitors the most important files in the existing system, while NIDS can analyze the incoming network flows. Other researchers divide IDSs according to the approach of the system, the most known is signature-based detection, which can recognize unknown and bad patterns, and anomaly-based detection, which has the capability of detecting deviations from a model of "good" traffic and often relies on artificial intelligence [4].

One of the main security issues for IDS is dealing with malicious software deviations that lead to security network breaches. Cyber-attacks are even more complex in the detection of unknown malware attacks because of the fast development of advanced evasion approaches to pinch-critical data and information and avert IDS from detection [5-7]. Therefore, new techniques are presented for attack detection. Machine learning and deep learning techniques have lately been applied to intrusion detection of anomalous behaviors in network traffic [8-10]. Another challenge is "Mirai" which is a special sort of botnet that causes huge-scale distributed denial-of-service (DDoS) attacks by harming IoT devices [11]. The "Persirai" Botnet is one variant of the "Mirai" code that continues to grow and infects more than 1000 IP camera models [12]. Current protection components should be improved to fit the IoT ecosystem [13]. However, usage protection mechanisms are straightaway beaten when faced with fixed protection risks, with several types of attacks made by intruders to avoid existing security settings.

For the security of the IoT, several solutions proposed recently, [14-18] have shown that deep learning approaches can detect IoT threats more efficiently than traditional machine learning approaches. However, the cloud layer may only have the resources to run these approaches. In addition, these models are not always very effective in some concerns, such as remote surgery, since the system is designed to make decisions in real time. Previous work on IoT attacks [5, 8] has shown that machine learning approaches such as SVMs can only provide meaningful results if they are merged with a feature extraction algorithm. This combination of algorithms requires high power resources, which are not available in most IoT devices and sensors. Machine learning techniques such as decision trees, naïve Bayes, and other algorithms are very robust for applications such as offline or non-interactive predictions between small datasets. However, these models are considered weak when applied to real-time predictions. Researchers showed that the detection rate is low when using these classifiers to identify IoT attacks and detect threats [5-7].

Recently, there is a growing use of ensemble deep-learning models, due to their ability to classify multiple patterns of data. In addition, ensemble learning can exploit predictive accuracy by merging the strengths of multiple baseline classifiers. Accordingly, this research is concerned with applying ensemble deep learning models to IDS, focusing on the recognition of traffic patterns to detect threats on Internet connected devices.

To enhance the predictive accuracy of deep learning models in intrusion detection, this research has three main objectives. First, we present an enhanced meta-learning model based on ensemble stacking to improve classification performance. The proposed ensemble model relies on merging the predictions of several groups of baseline deep-learning models using two tiers of shallow meta-classifiers instead of a one-tier meta-classifier as in classical stacking. In the proposed model, we train a group of baseline deep-learning classifiers on different partitions of the training dataset in Tier-0. Then, in Tier-1, collections of shallow meta-classifiers are trained on the output prediction of Tier-0. The predictions from Tier-1 are combined with a top-level meta-classifier in the second tier to get the final prediction. The proposed ensemble model aims to extend the variety in the ensemble using variants of the training dataset, a group of trained deep learning classifiers, and variation within the merging of basic deep learning classifiers to reduce the variance and to create more robust models. Second, conducting a large variety of experiments on two different, benchmark datasets to estimate the performance of the proposed model. For each dataset, different deep learning classifiers are trained on different trained datasets, and then their best performance is compared with the proposed model. Finally, the proposed ensemble model based on our enhanced stacking approach is evaluated using two recent huge-size datasets and conduct comparative experiments on the two datasets using the well-known ensemble methods, namely, voting and classical stacking.

The paper's primary contributions can be summarized as follows:

- Proposing a novel and efficient improved stacking meta-learning model by combining the baseline deep learning classifiers with two tiers of shallow meta-learners to enhance the performance of the classification process.
- Training various deep learning models using two different benchmark datasets, one of them created based on real data from IoT devices.
- Extending the experiments by comparing the performance of various commonly used ensemble methods with the performance of our proposed ensemble model.
- Investigating the impact of prediction type within individual deep learning models on the proposed ensemble model.

This paper is organized as follows; In section 2, the authors provide a background on the ensemble methods. Section 3 presents some related work and the challenges faced by the researchers. Section 4 covers the architecture of the proposed ensemble deep learning model and the methodology used to build the model. In section 5 the authors discussed the setup of the environmental experiment and expressed the benchmark datasets used to evaluate the model. Section 6 analyses the results and compares them with other researchers' results. Finally, section 7 concludes the paper.

2. Background on ensemble methods

The concept of ensemble learning was introduced in 1979 [18]. Two or more classifiers partition the feature space using an integrated system that uses an ensemble system in a divide-and-conquer fashion. More than a decade later, another ensemble model was proposed and showed that the similar performance of neural networks could be enhanced by introducing variance reduction properties using ensembles [19]. However, the work [20] located ensemble systems at the centre of research on machine learning. This was achieved by proving that combining weak classifiers using a technique called "boosting" can produce strong classifiers, perhaps in the most correct sense.

According to how the baseline classifier interacts with others, ensemble methods could generally be categorized as dependent or independent approaches. In the dependent technique, the outcome of one classifier affects how the next classifier is created. Among examples of dependent approaches, boosting algorithms are the most well-known [21]. The independent technique generates each classifier independently from different dataset subsets and then somehow integrates the outcomes. To create an effective ensemble, classifiers should preferably be independent or negatively correlated, according to [22]. Random forest and stacking are broad examples, among many other approaches, of independent methods [22]. The general framework of all ensemble learning in the independent method uses an aggregation function G to combine a set k of baseline classifiers: c; c2;...; ck towards predicting a single output. Given a dataset of size (n) and features of dimension m, D = {(x_i, y_i)}, $1 \leq i \leq n$, $x_i \in R_m$}, the prediction of the output based on this method is given by Eq. (1).

$$y_i = \Phi\ (X_i) = G\ (c_1, c_2, . . . , c_k) \qquad (1)$$

Where $y_i \in Z$ appoints the classification. Constructing an ensemble model, using this general framework, requires taking a decision on how to train the baseline classifiers. Numerous independent ensemble approaches have been proposed in recent years for their successful enhancement in prediction accuracy and ease of parallelization in training [23]. Every ensemble approach requires a suitable merging of multiple learners to create the final predictive model. In general, fusion methods can be divided into bagging, averaging, and meta-learning techniques.

This section provides a summary of some of the most known ensemble approaches. Generally, three of the most effective and widely used approaches are described below; bagging, averaging, and meta-learning.

2.1 Bagging ensemble

Bagging is one of the most commonly used techniques to improve the prediction results of individual models. Its fundamental idea is to build more diverse prediction models by fitting the stochastic distribution of training datasets. In particular, the same learning algorithm is applied to different bootstrap data samples from the original training dataset, and the result is obtained by averaging methods. The bagging technique is extremely useful when dealing with large and high-dimensional datasets [24].

2.2 Averaging ensemble

The simplest method for combining predictions from multiple models is the mean method [25]. Averaging is a widely used technique, where every model is separately trained, and the algorithm linearly integrates the predictions of all models by averaging them to present the final prediction. This technique is easy to apply without requiring additional training for numerous individual predictions. Traditionally, voting has been the standard method for averaging the predictions of base classifiers. The final prediction result is usually determined by a majority vote over many classifier predictions, known as "hard voting". The term "hard voting" can be expressed mathematically by Eq. (2), which indicates the statistical mode of the classifiers' predictions.

$$y_i = mode\ \{c1, c2, . . . , ck\} \qquad (2)$$

However, while hard voting is easy to implement and produces better results than basic classifications, the probability of lesser predicate classes is not considered as well. Consequently, the soft prediction depends on the probability value of each classifier instead of the prediction labels of each classifier. Soft prediction can be formalized using Eq. (3).

$$y = argmax \frac{1}{n} \sum_{j=1}^{n} W_{ij} \qquad (3)$$

Where (W_{ij}) is the probability of (i^th^) class label of the (j^th^) classifier. An improved version of voting is to give weight to each classifier proportional to its accuracy performance on a validation set [24].

2.3. Meta-learning ensemble

This is a technique for learning from further classifiers. At variance with traditional learners, meta-training classifiers have two or more learning phases [26]. Start by training the baseline classifiers, and then train the meta-classifier, with the combined predictions of the baseline classifiers. During the prediction state, the baseline classifiers perform the classifications, and then the meta-classifier produces the final classification.

Stacking is a method of meta-learning that uses a two-stage classification structure, namely baseline classifiers tier, and top meta-classifier tier. The rationale behind this approach stems from the limitations of the simple average ensemble, in which each model, no matter how it worked, was treated equally in the ensemble prediction. On the other hand, it generates a higher-tier model for joining the predictions of every singular model. In particular, the models that make up the ensemble are all trained individually with the same training set, typically referred to Tier-0 training set. Then the combined predictions of all individual models were used to create a Tier-1 training set. To avoid meta-classifier overfitting, the data samples used to train the baseline

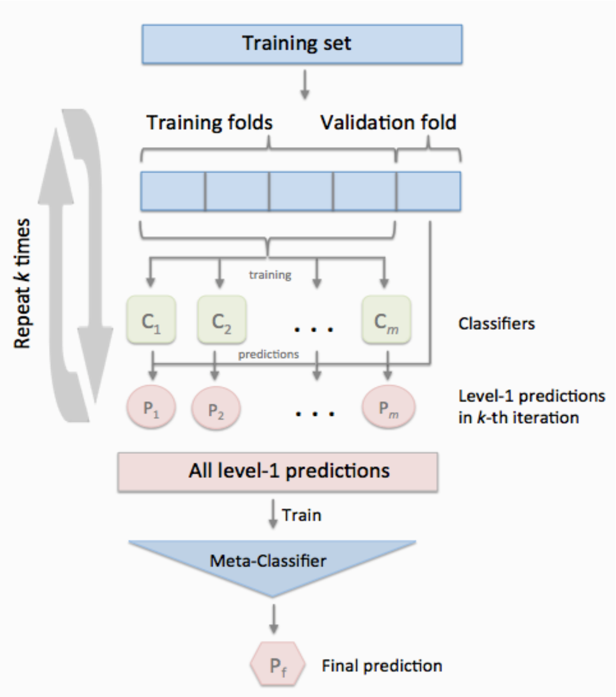


Figure. 2 Traditional ensemble stacking [26]

classifiers should be excluded during the meta-classifier training phase, as shown in Fig. 2. Consequently, the dataset must be divided into two separate parts. The first portion is used to create the base-tier classifiers, while the second portion is used to create the meta-dataset [26].

3. Related work

This section describes related works on different learning techniques used to propose an intrusion detection system using both machine and deep learning as the main classifier.

3.1 Ensemble machine learning classification

The ensemble classifier technique is one of the methods for producing an influential classifier with higher classification accuracy compared to traditional machine learning classifiers. Due to being hybrid, ensemble techniques, in general, achieve better results on most datasets. Therefore, ensemble-learning models have been developed in various research to successfully generalize machine-learning algorithms in intrusion detection. They have demonstrated that the current ensemble models perform better than baseline classifiers [26].

In [27], authors presented an intrusion detection and prevention system for the IoT based on ensemble techniques; boosted, bagged, RUSBoosted trees, and subspace discriminant. The authors evaluated their work by using the RPL-NIDDS17 dataset in a way to avoid anomalous events in IoT ecosystems, most definitely the botnet attack against MQTT, HTTP, and DNS protocols. Their model achieved an accuracy of 94.4% by boosting trees, 93.3% by bagging trees, 78.7% by Subspace discriminant, and 94.1% by RUBoosted trees. Authors believe that ensemble classifiers based on RUSBoosted trees achieve better performance than other ensemble classifiers.

Early in 2020 [28], researchers proposed an ensemble-based intrusion detection model to identify various malicious activities on IoT platforms, using the UNSW-NB15 dataset. To reduce complexity, dimensionality reduction using the ensemble classifier XGBoost algorithm was applied only over 19 features of the dataset. Results show that using the ensemble DT-XGBoost improved the accuracy of the decision tree from 88.13% to 90.85% for the binary classification. In addition, for multi-class classification, the ANN-XGBoost was the best solution and achieved an accuracy of 77.51%.

In [29], the authors suggested an intrusion detection system based on machine learning models to detect attacks in IoT networks. They evaluated their model based on data captured by IoT sensors using Node MCU ESP8266, DHT 11 sensor, and wireless router. Their model had an accuracy of 98.95% using the AdaBoost ensemble classifier. Although the results are excellent, the researchers did not mention how they dealt with the problem of an imbalanced data set.

In [30], the authors studied ten different algorithms, comparing seven supervised and three unsupervised learning algorithms, to achieve the best solution for the detection of intrusion in IoT-connected devices. The results show the XGBOOST as the best performer. The performance of all ten algorithms was tested on NSL-KDD datasets with accuracy, with the area under the curve (AUC) as evaluation metrics. The authors mentioned that the XGBoost classifier gives relatively higher results and runs approximately 10 times faster than other traditional classifiers. However, the researchers rely on only NSL-KDD without treating the imbalance issue in the dataset. In addition, the authors believe that NSL-KDD lacks public network data.

In [31], the authors presented an ensemble model based on a stacking algorithm that can detect intrusions at the level of the network. The presented model was evaluated using the N-BaIoT, CICIDS2017, IoTID20, and NSL-KDD. Their model achieved an accuracy of 98.5%. The authors mentioned that their proposed ensemble model performed better than any single machine learning classifier when evaluated with the N-BaIoT dataset.

Table 1. Recent research work in intrusion detection using ensemble-based machine learning

Ref.	Model	Dataset	Evaluation metrics
[27]	RUSBoost	RPL-NIDDS17	Acc., ROC
[28]	XGBoost	UNSW-NB15 NIMS	Acc., ROC
[29]	Adaboost	Generated dataset	All
[30]	XGBOOST	NSL-KDD	Acc., MCC, AUC
[31]	Stacking	N-BaIoT	All

They also mentioned that ensemble models deliver better results for intrusion detection systems in IoT environments. However, they did not treat the imbalance issue in any of the used datasets. Table 1 summarizes the recent research on ensemble-based machine learning used for intrusion detection.

Despite the success of ensemble models in enhancing the accuracy of baseline classifiers studied in modern intrusion detection research, there is a clear limitation in preceding studies regarding the size of the datasets and the predictive accuracy of the baseline classifiers.

3.2 Ensemble deep learning classification

Deep learning models are a promising alternative to traditional machine learning approaches. It has shown outstanding performance on large datasets in various intrusion detection systems.

In [32], the authors proposed an ensemble weighted majority algorithm to increase accuracy by employing feature elimination techniques for attack detection. They utilized three different deep learning algorithms, DNN, LSTM-RNN, and DBN to create the ensemble model. Their model consists of two levels, achieving an overall accuracy of 96.91% with the multi-class classification of the NSL-KDD dataset, and 95.81% with the multi-class classification of the CICIDS2017 dataset.

In [33], the authors presented an intrusion detection two-layer ensemble framework (I-SiamIDS) for resolving the problem of an imbalance. The first layer used binary extreme gradient boosting (bXGboost), DNN, and Siamese neural network for filtration of inputs to identify attacks; the second layer used a multi-class extreme Gradient boost classifier to classify anomaly behavior into multiple attacks based on two different datasets, NSL-KDD and CIDDS-001. Their model achieved an accuracy of 80.1% by using XGBoost over the NSL-KDD dataset while achieving 97.26% by using DNN over the CIDDS-001 dataset. In addition, the authors believe that the reason for selecting more than one classifier over two layers of ensemble models was to increase the number of correctly identified attacks.

In [34], authors proposed an ensemble CNN-LSTM deep learning model to detect attacks only on four security cameras in IoT environments. Their model has been evaluated by using the N-BaIoT dataset, achieving an accuracy of 87%, 89.23%, 89.67%, and 88.28% for each security camera.

Early in 2019 [35], authors presented an ensemble method based on Autoencoders and deep neural networks (DNN), deep belief neural networks (DBN), and an extreme learning machine (ELM). They evaluated their model on the NSL-KDD dataset. Their ensemble model experiments achieved an accuracy between 85.93% and 98.28% for all attack classes in the dataset. They believe that results for the attack classes (R2L) and (U2R) were rather low due to the imbalanced samples in the NSL-KDD dataset.

Later on 2022 [36], other authors presented an ensemble model based on autoencoders and long short-term (RNN) and temporal convolutional networks (TCN). The authors believe that the lack of availability and imbalanced datasets, combined with the narrow scope of generating an IDS based on only one classifier, add further limitations. They evaluated their model with different datasets and achieved accuracy with N-BaIoT of 80.5%, and 90.7%.

In [37], the authors presented an enhanced intrusion detection model using deep SDAE to rise the efficiency of dimensional reduction in their model. They train their model with different learning algorithms RNN-LSTM, Decision Trees, Naïve Bayes, and SVMs over the NSL-KDD dataset. The authors expressed that the proposed hybrid model using SDAE with LSTM achieved high accuracy of 86.8% while using SDAE with SVMs achieved an accuracy of 84.1%, proving that the hybrid deep learning models can achieve higher accuracy than other traditional models. The authors also mentioned that their proposed models are highly possible to be enhanced with ensemble learning models.

In [38], the authors presented a deep learning approach based on a real dataset (N-BaIoT) to detect DDoS attacks on IoT, using a combination of 3 different deep learning classifiers. The authors mentioned that the combination of BiLSTM-CNN has proven to be great combination, achieving the highest accuracy of 89.7%. The researchers implemented their model to detect DDoS attacks only, not other attacks. Table 2 summarizes the latest ensemble-based deep learning published papers.

Table 2. Intrusion detection ensemble-based deep learning published papers.

Ref.	**Model**	**Dataset**	**Evaluation metrics**
[32]	Multi-CNN	NSL-KDD CICIDS2017	Acc.
[33]	bXGBoost, Siamese NN, DNN	NSL-KDD CIDDS-001	All
[34]	CNN-LSTM	N-BaIoT	All
[35]	Autoencoder, DNN, EML	NSL-KDD	Acc., AUC
[36]	Autoencoder (LSTM-TCN)	N-BaIoT	Acc.
[37]	Deep SDAE	NSL-KDD	Acc.
[38]	BiLSTM-CNN	N-BaIoT	All

4. The proposed ensemble deep learning model

Due to the huge amount of network data gathered from IoT devices and various new attacks, traditional classifier algorithms mainly perform poorly. Ensemble learning can create improved predictions and gain better performance than any individual classifier; in addition, it can reduce the spread or dispersion of the prediction of the model's performance.

Several research initiatives have combined ensemble learning to improve classification performance, to avoid overfitting, and to minimize errors. Troika [39] is one of such attempts to improve stacking to address multi-class problems. The model is based on four layers of stacking, where the last layer contained only one model that outputs a vector of probability as a final decision. The efficiency of this solution comes with an additional computational cost due to the "extra two layers" of traditional machine models in stacking. In this regard, many traditional machine-learning tools adopted for IoT are difficult to be used in a real-time context due to their limitations in terms of classification performances or computational cost. In addition, the authors found that their model performs similarly in some cases when using traditional stacking.

Considering the above-mentioned aspects, we propose an ensemble model to detect zero-day attacks. The key idea of the proposed ensemble model is to introduce an enhancement to the classical stacking with cross-validation, as given in [26], by using two tiers of shallow meta-classifiers; instead of only one tier of shallow meta-classifier in classical stacking, to combine a group of various baseline classifier at different tiers. The first tier named Tier-0 is responsible for training a group of different deep-learning classifiers on the complete training dataset. Next added Tier to improve the traditional stacking inspired by [26], Tier-1, a group of shallow meta-classifiers will be trained using the output predictions of Tier-0. In the final tier, the prediction from Tier-1 will be combined to create meta-data. Furthermore, a top-tier meta-classifier or (Tier-2) is generated by combining the output predictions of Tier-1 or Meta-data to produce the final prediction. The initial step is to use a state-of-the-art dataset created for IoT. In addition, we performed pre-processing of the dataset, such as redundancy removal, cleansing of data, and data normalization.

4.1 Architecture of proposed model

The philosophy behind the proposed model is to improve the classical stacking [26] shown in Fig. 2. The proposed model aims to improve the classical stacking by using two stages of shallow meta-classifiers instead of one stage of meta-classifier The architecture of the proposed ensemble deep learning model consists of three levels. In the first level, Tier-0, the given dataset is divided into a training dataset and a testing dataset, then splitting the training dataset into K-folds. Each baseline classifier is trained individually utilizing (K-1) parts of the training dataset and the predictions are made for the K^{th} part. Each baseline model is fitted on the whole training dataset to calculate the performance of the test dataset. In the second level, Tier-1, each baseline-learners' outputs are integrated using various shallow classifiers or meta-classifier. Finally, in the third level, a top-Tier meta-classifier, Tier-2, is used to combine the outputs of all shallow classifiers producing the final prediction. Fig. 3 shows the architecture of the proposed ensemble model.

The algorithmic description of our proposed ensemble model is given in the following subsection.

4.2 Formal description

The steps of the proposed ensemble model are given in Algorithm 1.

The below algorithm shows the steps needed to train the proposed ensemble model. Given a dataset (***Ds***), step one randomly generates further separated into training and testing dataset $\boldsymbol{Ds} = (X_i,y_i)$. Then various baseline deep learning classifiers (T) are applying the training dataset (x_i) to develop Tier-0. Each baseline deep learning classifier (h_t) combined with the testing dataset (y_i) are used to create the Meta-data which will be used as training data for the next Tier. After creating Tier-0, each $h_i(x_i)$, is used to create Tier-1 instances which consist of meta-

```
Algorithm 1: Proposed ensemble model
Required: Training dataset $Ds$ ={$X_i$, $y_i$}, i = 1 to m
Output: An ensemble Classifier $H$

Step1: adopt cross-validation in preparing a training dataset.
split $Ds$ randomly to equal K-size subsets $Ds$={$Ds_1$, $Ds_2$, .., $D_k$}
for $K \leftarrow 1$ do
  Learn baseline classifier (Tier-0)
  for $t \leftarrow 1$ to $T$ do
    Learn baseline classifier $h_{kt}$ based on $Ds/Ds_k$
  end for

  Step2: Construct new dataset from $Ds`$
  for $x_i \in Ds_k$ do
    construct a new dataset that contains
      $Ds`$={$X`_i$, $y_i$},
      where $x`_i$ = {$h_{k1}(x_i)$, $h_{k2}(x_i)$,….., $h_{kT}(x_i)$}
  end for
end for

Step3: Learn Shallow Classifies (Tier-1)
for $t \leftarrow 1$ to $T$ do
    Learn shallow classifiers $h`_t$ based on new
     constructed dataset.
end for
Step4: Construct new dataset from $Ds``$
for $i \leftarrow 1$ to $n$ do
      construct a new dataset that contains
        $Ds``$={$X``_i$, $y_i$},
     where $x``_i$={$h`_1(x_i)$, $h`_2(x_i)$,.., $h`_T(x_i)$}
end for

Step5: Learn Top-Tier Classifier (Tier-2)
  Learn a Top-Tier classifier $h``$ based on
  the newly constructed dataset
return $H(x) = h``((h`_1(x), h`_2(x),….., h`_T(x))$
```

datasets (***Ds`***), where $1 \leq i \leq n$, generated from the output predictions on test dataset of Tier-0.

Every generated (***Ds`***) in Tier-1 have (*K+1*) features, whose values were the predictions of the baseline classifiers, whereas $1 \leq i \leq K$, and an additional feature represents the target class ($y_i$). The meta-data (***Ds*ᵢ`**) will also be separated into ($X`_i$,$y_i$). After the Meta-data of Tier-1 is created, various shallow meta-classifiers will be utilized to create the Tier-1 models ($h`_t$). Each training ($X`_i$) is applied on every shallow meta-classifiers generating the Meta-data (***Ds``***). After creating Tier-1 models, the test data ($y_i$) are used to create the Top-Tier Meta-data named Tier-2 Meta-data, which will be created within two steps. The first step, the (n) predictions of Tier-1 models on ***X``*** with an additional feature representing the target feature (y_i) are used to generate (***Ds``***). Secondly, mount all (***Ds``***) to perform the final Meta-data. Then a Top-Tier meta-classifier **(*H*)** is used to learn the Tier-2 Meta-data.

5. Simulation experiments

To evaluate the performance of the proposed model, simulation experiments have been conducted based on a group of benchmark datasets. According to [36-37], datasets such as KDD99, NSL-KDD, RPL-NIDDS17, and ISCX, realized a limited number of attacks, which are also outdated. Therefore, we have adopted N-BaIoT and UNSW-NB15 as a recent powerful benchmark datasets for evaluating our proposed model.

5.1 N-BaIoT and UNSW-NB15 datasets

The existing datasets do not represent the comprehensive representation of the modern orientation of network traffic and attack scenarios. These reasons have investigated a serious challenge for cyber security researchers. In this work, we relied on two different state-of-the-art and recent datasets.

The N-BaIoT dataset reports the lack of public botnet datasets, especially for IoT. It proposes real traffic data, collected from nine IoT commercial devices [36]. It consists of (11) classes; such as, benign and two types of the latest IoT malware, "Mirai" and "Bashlite" with a total of (6.5) million instances. The benign data were captured immediately after setting up the network to ensure that the data was benign. For two types of packet sizes, packet counts, and packet jitters, the times between packet advent were extracted for each statistical value. A total of 23 features were extracted for each of the (5) time windows (100ms, 500ms, 1.5s, 10s, and 1 min), creating 115 features [36].

The UNSW-NB15 was created by the IXIA PerfectStorm tool. It has information about the latest IoT attacks. It could be considered an intrusion network dataset that contains nine different attacks. It includes 10 different classes, such as DoS, Fuzzers, Backdoors, and worms. It has more than (175) thousand records as a training dataset and (82) thousand records as a testing dataset from the different types of attacks. The completely saved records are two million and 540 thousand [38].

5.2 pre-processing phase

The pre-processing was made to enhance the

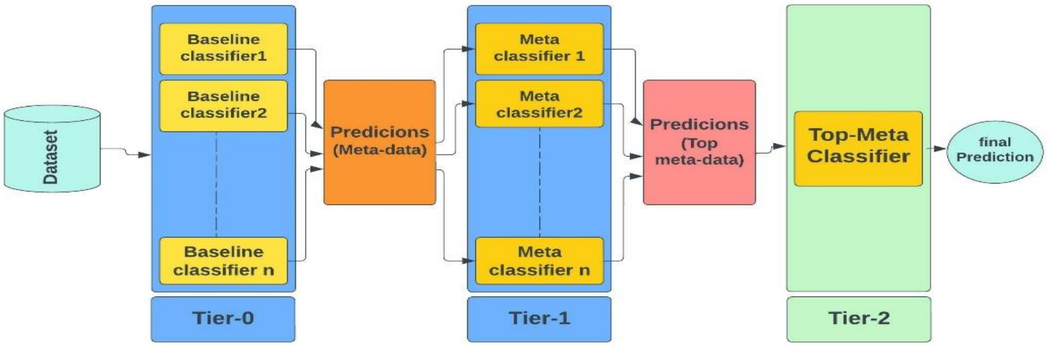


Figure. 3 The architecture of the proposed ensemble model

Table 3. Practical settings of varied proposed models

	B.size	Epoch	Lr. rate	Drop out	optimizer
DNN	400 50	50 100	0.01 0.001	0.4 0.2	Adamsgd RMS-Prop
LSTM RNN	400 200 100	50 100	0.01 0.001	0.4 0.2	
ResNet	400 200 100	50 100	0.01 0.001	0.4 0.2	

effectiveness and the performance of the proposed model, where the first step in the pre-processing was the removal of the redundant and checking for the missing values. In addition, for more enhancement, MinMaxScaler was applied to normalize the dataset and One-hot-Encoder on the target labels for the preparation of the deep learning algorithm. Then SMOTE (Synthetic Minority Oversampling Technique) was applied to the dataset to solve the problem of the imbalanced dataset. SMOTE is an oversampling method where synthetic samples are generated for the minority class. It aids in resolving the overfitting issue posed by random oversampling and emphasizes the feature space to produce novel instances with the help of exclamation between the positive instances that lie together [33]. Next, each baseline classifier is trained with various hyper-parameters during the training of the baseline models, as shown in Table 3.

5.3 Generating baseline learning models

To evaluate the proposed ensemble model, we built a group of deep classification algorithms constituting the baseline deep learning classifiers. Then trained on a different size of the benchmark dataset with different optimizers such as "adam", "RMSProp" and "sgd" were tested with various learning rates, batch sizes, and epochs. As shown in Table 3. The baseline classifiers used to evaluate the proposed model were LSTM-RNN, DNN, and Residual Networks.

5.3.1. Recurrent neural network (RNN)

RNN is considered the state-of-the-art algorithm for sequential data; it is a robust type of neural network and one of the most promising algorithms as it is the only algorithm with internal memory. The rise in computing power, the sheer amount of data we now have to process, and the invention of long-short-term memory (LSTM) in the 1990s made RNNs come to prominence. Internal memory allows the RNN to remember important things about the input; it receives and predicts what comes next with great accuracy. This makes it an ideal algorithm for sequential data such as anomaly detection. LSTM-RNN can understand sequences and their context more deeply than other algorithms and classifies network traffic with high accuracy in detecting anomaly behavior. In addition, it shows extensive potential in enhancing IoT system security [36].

5.3.2. Deep neural network (DNN)

DNN is a feedforward multilayer neural network. It can model complex non-linear relationships. DNNs are widely used in supervised and reinforcement learning. In addition, it can produce better results than traditional machine learning models [35].

5.3.3. Residual networks (ResNet)

A residual network (ResNet) stacks residual blocks on the pinnacle of each different to shape a network. It is an innovative neural network model, introduced in [38], in their research "Deep residual learning for image recognition". Residual learning is realized by establishing a direct connection between the input and output. CNN based on residual learning has achieved outstanding results in image recognition. In intrusion detection, it is also vital to build deeper networks to improve the detection capabilities of IDSs [40, 41].

5.4 Shallow Meta-classifiers combiner

To merge the baseline trained models' predictions, several meta-classifiers were used as Top meta-learner, such as NB, XGBoosing, LR, SVMs, and RF. Generally, any shallow classifier could merge the Tier-1 predictions.

5.5 Evaluations

Basic evaluation metrics such as accuracy, recall, precision, and F1-score values are derived from confusion metrics that Consist of TP, TN, FP, and FN to represent the Confusion matrix [42].

6. Results and discussion

The bassline classifiers were implemented using TensorFlow and Keras. In addition, dropout and batch normalization are two approaches used to reduce the overfitting and long training time with the LSTM and Res.Net [43]. Scikit-learn is another library, used to implement shallow classifiers [44].

To evaluate the influence of the proposed ensemble model, several experiments on the datasets were done to assess the performance of each baseline deep learning classifier individually. In addition, we evaluated the proposed model using predictions generated from baseline deep learning models. Lastly, we summarize the experimental results and compare the proposed model with other known ensemble methods, such as stacking and voting. Both benchmark datasets were divided into training, and testing datasets, with a ratio of 70% as the training dataset, and the other 30% as the testing dataset. To train the baseline classifiers in Tier-0, the training dataset was partitioned using one of the partitioning methods. In this experiment, 10-fold cross-validation was used on the predictions of the bassline classifiers. Additionally, for the final predictions, various shallow classifiers show greater performance. The next section provides a practical analysis of the proposed ensemble deep learning model and then compares the performance with the baseline models.

Using cross-validation on the predictions of the baseline classifiers, random forest (RF) is shown to be the best combinatory for merging the predictions at Tier-1. However, for the final prediction, different shallow top-meta classifiers show better performance.

Table 4. Results of using classifiers in N-BaIoT

Classifiers	Acc. %	Pr.	Rec.	F1
LSTM-RNN	92.3	90	89	92
DNN	92.7	89	87	91
ResNet.	92.85	91.8	92.2	92

Table 5. Accuracy of the proposed ensemble model using various meta-classifiers over N-BaIoT

	XGB	NB	LG	SVM	RF
Acc. %	99.8	93	99.3	99.4	99.2

Table 6. Results of using classifiers in UNSW-NB15

Classifiers	Acc.	Pr.	Rec.	F1
LSTM-RNN	94.2	88	90	91
DNN	95.2	95	93	92.2
ResNet.	95.9	95.2	95.7	95

Table 7. Accuracy of the proposed ensemble model using various meta-classifiers over UNSW-NB15

	XGB	NB	LR	SVM	RF
Acc. %	99.82	92.8	99.1	99	98.9

In the next section, we present an experiential analysis of the proposed ensemble model and compare the performance with the better baseline models.

6.1 Results on N-BaIoT

Here, we trained various deep learning classifiers individually as baseline classifiers. Table 4. summarizes the obtained accuracy of the trained models. The evaluation of baseline classifiers shows that the accuracy of Res.Net outperforms all other baseline classifiers with an accuracy of 92.85%. The best hyper-parameters used were: (0.001) as learning rate, 512 batches, and "adam" as an optimizer.

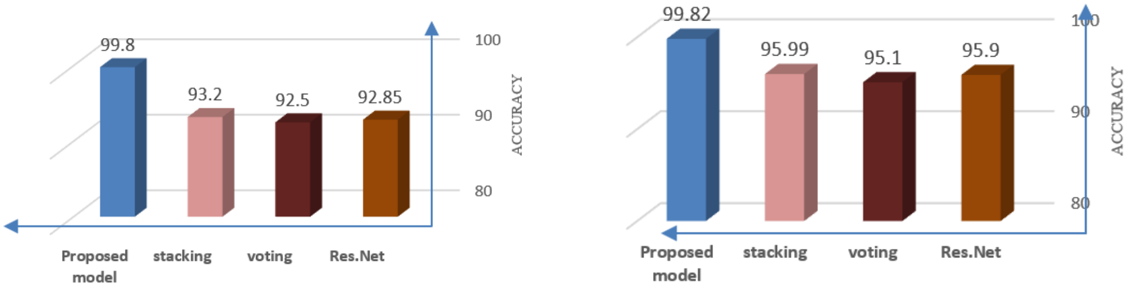
Then we combined the three baseline models' output predictions by using various meta-classifiers. The results of the proposed ensemble model outperformed the best-performed model of the baseline models in all conducted shallow Meta-classifies. According to the experimental results in [31, 34, 36, 38], the best-achieved accuracy was 98.5%. Here, the experimental results, in Table 5, show that the ensemble with XGBoosting as a meta-learner exceeds the results of other meta-classifiers with an accuracy of 99.8%. Compared to the best individual deep learning classifier, the accuracy of the proposed model was increased by 6.95%. In addition, the proposed model exceeds the best-achieved model in [31] by 1.3%.

Table 8. Results of all models

Dataset	Baseline models	Voting	Classical stacking	Proposed model

N-BaIoT	LSTM-RNN= 92.3%
	DNN= 92.7%
	Res.Net= 92.85%
		92.5%	93.2%	99.8

UNSW-NB15	LSTM-RNN= 94.2%
	DNN= 95.2%
	Res.Net= 95.9%
		95.1%	95.99%	99.82



(a) (b)

Figure. 4 Comparison of the proposed model with known ensemble methods and best bassline models using: (a) N-BaIot dataset and (b) UNSW-NB15 dataset

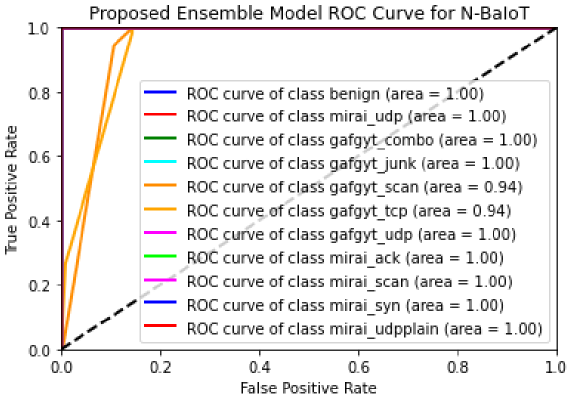


Figure. 5 ROC Curve of the proposed model using N-BaIoT dataset

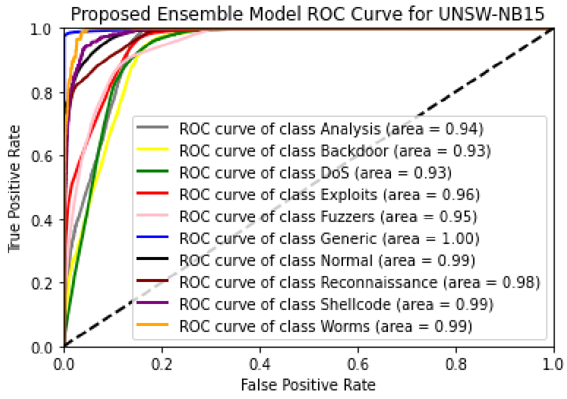


Figure. 6 ROC Curve of the proposed model using the UNSW-NB15 dataset

6.2 Results on UNSW-NB15

The experimental results show that the accuracy of Res.Net outperforms all other baseline classifiers with an accuracy of 95.9%. As shown in Table 6. Table 7. Summarizes the accuracy of the proposed ensemble model using the merging of top-tier meta-learners. The results of the proposed model show expressively performed better than the best-performed model of the baseline deep learning models in all accompanied meta-learners. In addition, the merging of baseline deep learning models on soft prediction advance increased the accuracy. The best hyper-parameters used are (0.001) as the learning rate, 1024 batches, and "adam" as an optimizer.

According to the experimental results in [28] the best-achieved accuracy we 96.91%. Here, the experimental results, in Table 7, indicate that the ensemble with XGbooting as a meta-learner achieves the best accuracy. Therefore, the proposed ensemble exceeds the best-achieved model [28] by 3.92%.

6.3 Discussion and summary of results

The above results obtained on the two different benchmark datasets show that the proposed model enhances the accuracy of the baseline deep learning models. Along with the results of the previous experiments, we compared the performance of the proposed model with that of two well-known effective ensemble techniques on the same generated baseline models, namely: voting and classical stacking. The summary of the obtained accuracy with the benchmark datasets is shown in Table 8 and illustrated in Figs. 4 (a) and (b).

We showed a severe evaluation based on multiple limitations to validate the performance of the proposed detection model. In addition, we carried out the 10-fold cross-validation shown. The ROC curves in Figs. 5 and 6 grant the overall performance of our proposed ensemble model with the N-BaIoT and UNSW-NB15 datasets. The proposed model improves the detection rate as shown in Table 8.

As shown in Figs. 5 and 6, a model with low prediction values of FPR, FNR, is considered an efficient model. In addition, according to the experimental results in [27, 29, 30, 32-33, 35, 37] achieved low FPR and FNR. Here, it is clear the proposed ensemble model achieved the lowest FPR, and FNR of 0.004%, and 0.003% respectively.

7. Conclusion

The increasing number of IoT devices is spurring research examining the highly sophisticated security threats associated with them. Current literature proves that IoT devices are vulnerable to various botnet attacks. Additionally, botnet attacks can wreak havoc across IoT networks. As a result, there is an urgent need for efficient, adaptable, cost-effective, and highly scalable solutions capable of detecting botnet attacks with the ability to identify zero-day attacks. In recent years, experimental research conducted by the machine learning community has shown that combining the outputs of different classifiers reduces the generalization error and can cope with large variances of individual classifiers. Ensembles are therefore an elegant solution for the handling of high variance of individual classifiers while minimizing general errors. The idea of combining different models to create a predictive model has been studied for a long time. The key idea behind ensemble strategies is based on the principle of different models and combining their predictions to improve performance.

In this paper, we presented an improved stacking ensemble strategy that combines a board of baseline classifiers using two stages of meta-classifiers instead of one meta-classifier. The key idea of the proposed ensemble relies on increasing the diversity of classifiers to improve performance. To test the efficiency of the proposed ensemble approach, we performed several experiments on two public benchmark datasets for IoT. A group of robust baseline classifiers is trained on each benchmark dataset and their best models are compared to the proposed ensemble method. Specifically, we trained three deep learning models with various iterations of hyper-parameter tuning and conducted a comparative study using five different shallow meta-classifiers to integrate these models. Furthermore, we evaluated the accuracy of the proposed ensemble method in comparison with further deep-learning ensemble models widely used in the literature on the same-trained base model. The results disclosed that the proposed improved stacking considerably increased the performance of the baseline deep learning classifiers on the used benchmark datasets and outperformed the classical stacking and weighted ensemble voting.

Conflicts of interest

The authors declare no conflict of interest on this study, authorship, and publishing of this manuscript.

Author contributions

Conceptualization, methodology, formal analysis, investigation, resources, data curation, writing original draft and preparation, Hany Mohamed; writing review and editing, Ahmed Hamza; visualization, validation, and supervision, Hesham Hefny.

References

[1] R. Lee, M. J. Assante, and T. Conway, "Analysis of the Cyber Attack on the Ukrainian Power Grid. Defense Use Case", *Information and Security*, Vol. 39, pp. 265-274, 2018.

[2] H. Mohamed, H. Hefny, and A. Alsawy, "Intrusion Detection System Using Machine Learning Approaches", *Egyptian Computer Science Journal*, Vol. 42, No.3, 2018.

[3] H. Abdulghani, N. Alexander, A. Collen, and D. Konstantas, "A Study on Security and Privacy Guidelines, Countermeasures, Threats: IoT Data at Rest Perspective", *Symmetry*, Vol. 11, No. 6, p. 774, 2019.

[4] R. A. Ramadan and K. Yadav, "A novel hybrid intrusion detection system (IDS) for the detection of Internet of Things (IoT) network attacks", *Annals of Emerging Technologies in Computing*, Vol. 4, No. 5, pp. 61-74, 2020.

[5] D. Hemavathi and H. Srimathi, "Effective feature selection technique in an integrated environment using enhanced principal component analysis", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 12, No. 11, pp. 3679-3688, 2021.

[6] F. Salo, A. B. Nassif, and A. Essex, "Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection", *Computer Networks*, Vol. 148, pp. 164-175, 2019.

[7] S. Hosseini and B. M. H. Zade, "New hybrid method for attack detection using combination of evolutionary algorithms, SVM, and ANN", *Computer Networks*, Vol. 173, p. 107168, 2020.

[8] G. Giacinto, F. Roli, and L. Bruzzone, "Combination of neural and statistical algorithms for supervised classification of remote-sensing images", *Pattern Recognition Letters*, Vol. 21, pp. 385-397, 2000.

[9] A. Bansal and S. Mahapatra, "A Comparative Analysis of Machine Learning Techniques for Botnet Detection", In: *Proc. of the 10th International Conference on Security of Information and Networks SIN '17*, NY, USA, pp. 91-98, 2017.

[10] A. N. Jaber, and S.U. Rehman, "FCM–SVM-based intrusion detection system for cloud computing environment", *Cluster Computing*, Vol. 23, pp. 3221-3231, 2020.

[11] P. Radanliev, D. Roure, R. Nicolescu, M. Huth, R. Montalvo, S. Cannady, and P. Burnap, "Future developments in cyber risk assessment for the internet of things", *Computers in Industry*, Vol. 102, pp. 14-22, 2018.

[12] E. Bertino and N. Islam, "Botnets and Internet of Things Security", *Computer*, Vol. 50, No. 2, pp. 76-79, 2017.

[13] I. Abunadi, A. A. Albraikan, J. S. Alzahrani, M. Eltahir, A. Hilal, M. Eldesouki, A. Motawakel, and I. Yaseen, "An automated glowworm swarm optimization with an inception based deep convolutional neural network for COVID-19 diagnosis and classification", *Healthcare*, Vol. 10, No. 4, p. 697, 2022.

[14] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and Other Botnets", *Computer*, Vol. 50, No. 7, pp. 80-84, 2017.

[15] J. Yang, Y. Sheng, and J. Wang, "A GBDT-paralleled quadratic ensemble learning for intrusion detection system", *IEEE Access*, Vol. 8, pp. 175467-175482, 2020.

[16] W. Liu, X. Liu, X. Di, and H. Qi, "A novel network intrusion detection algorithm based on Fast Fourier Transformation", In: *Proc. of the 1st International Conference on Industrial Artificial Intelligence (IAI), IEEE*, Shenyang, China, pp. 1-6, 2019.

[17] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naive Bayes vs. decision trees in intrusion detection systems", In: *Proc. of the 2004 ACM Symposium on Applied Computing, Nicosia*, Cyprus, pp. 420-424, 2004.

[18] B. V. Dasarathy and B. V. Sheela, "Composite classifier system design: concepts and methodology", In: *Proc. of the IEEE*, Vol. 67, No. 5, pp. 708- 713, 1979.

[19] L. K. Hansen, and P. Salamon, "Neural network ensembles", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 10, pp. 993-1001, 1990.

[20] R. E. Schapire, "The Strength of Weak Learnability", *Machine Learning*, Vol. 5, No. 2, pp. 197-227, 1990.

[21] A. Mayr, H. Binder, O. Gefeller, and M. Schmid, "The evolution of boosting algorithms-from machine learning to statistical modelling", *Methods of Information in Medicine*, Vol. 53, No. 6, pp. 1403-1452, 2014.

[22] F. Haghighi and H. Omranpour, "Stacking ensemble model of deep learning and its application to Persian/Arabic handwritten digits recognition", *Knowledge-Based Systems*, Vol. 220, No. 14, 2021.

[23] L. Rokach, "Ensemble learning: pattern classification using ensemble method", *2nd Edition, World Scientific Publishing*, p.85, 2019.

[24] Z. H. Zhou, "Ensemble Methods: Foundations and Algorithms", *1st Edition, Chapman & Hall / CRC Press*, FL, USA, 2012.

[25] I. E. Livieris, L. Iliadis, and P. Pintelas, "On Ensemble techniques of weight constrained neural networks", *Evolving Systems*, Vol. 12, pp. 155-167, 2021.

[26] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning", *Artificial Intelligence Review*, Vol. 18, No. 2, pp. 77-95, 2002.

[27] A. Verma and V. Ranga, "Elnids: ensemble learning based network intrusion detection system for RPL based Internet of Things", In: *Proc. of fourth International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, India, pp. 1–6, 2019.

[28] S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset", *Journal of Big Data*, Vol. 7, 2020.

[29] U. Amin, A. S. Ahanger, F. Masoodi and A. Bamhdi, "Ensemble based Effective Intrusion Detection System for Cloud Environment over UNSW-NB15 Dataset", In: *Proc. of Intelligent Systems, Computing & Intelligent Systems, SCRS, India IEEE Access 8*, pp. 483- 494, 2022.

[30] S. S. Dhaliwal, A. Nahid and R. Abbas, “Effective intrusion detection system using XGboost”, *Information*, Vol. 9, No. 7, p. 149, 2018.

[31] A. Bansal and S. Kaur, “Extreme gradient boosting based tuning for classification in intrusion detection systems”, In: *Proc. of International Conference on Advances in Computing and Data Sciences, Springer,* pp. 372-380, 2018.

[32] W. Elmasry, A. Akbulut, and A. H. Zaim, “Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic”, *Computer Networks,* Vol. 168, p. 107042, 2020.

[33] P. Bedi, N. Gupta, and V. Jindal, “I-SiamIDS: an improved Siam-IDS for handling class imbalance in network-based intrusion detection systems”, *Applied Intelligence,* Vol. 51, pp. 1131-1151, 2021.

[34] M. Y. Alzahrani, and A. M. Bamhdi, “Hybrid Deep Learning model to detect Botnet attacks over Internet of Things environments”, *Soft Computing*, Vol. 26, pp. 7721–7735, 2022.

[35] S. A. Ludwig, “Applying a Neural Network Ensemble to Intrusion Detection”, *Journal of Artificial Intelligence and Soft Computing Research*, Vol. 9, No. 3, pp. 177-188, 2019.

[36] R. Ahmad, I. Alsmadi, W. Alhamdani, and L. Tawalbeh, “A comprehensive deep learning benchmark for IoT IDS”, *Computers & Security*, Vol. 114, No. 1, p. 102588, 2022.

[37] A. Sunyoto and Hanafi, “Enhance Intrusion Detection (IDS) System Using Deep SDAE to Increase Effectiveness of Dimensional Reduction in Machine Learning and Deep Learning”, *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, pp. 125-141, 2022, doi: 10.22266/ijies2022.0831.13.

[38] A. R. Sonule, M. Kalla, A. Jain, and D. S. Chouhan, “Unsw-Nb15 Dataset and Machine Learning Based Intrusion Detection Systems”, *International Journal of Engineering and Advanced Technology (IJEAT),* Vol. 9, No. 3, pp. 2638–2648, 2020.

[39] E. Menahem, L. Rokach, and Y. Elovici, “Troika An improved stacking schema for classification tasks”, *Information Sciences*, Vol. 179, No. 24, pp. 4097- 4122, 2009.

[40] Y. Pei, T. Huang, W. Ipenburg, and M. Pechenizkiy, “ResGCN: attention-based deep residual modelling for anomaly detection on attributed networks”, *Machine Learning*, Vol. 111, pp. 519-541, 2021.

[41] J. Man and G. Sun, “A Residual Learning-Based Network Intrusion Detection System”, *Security and Communication Networks,* Vol. 2021, No. 11, pp. 1-9, 2021.

[42] A. Attia, M. Faezipour, and A. Abuzneid, “Network Intrusion Detection with XGBoost and Deep Learning Algorithms: An Evaluation Study”, In: *Proc. of International Conference on Computational Science and Computational Intelligence (CSCI),* pp. 138-143, 2020.

[43] C. Garbin, X. Zhu, and O. Marques, “Dropout vs. batch normalization: an empirical study of their impact to deep learning”, *Multimedia Tools and Applications,* Vol. 79, pp. 12777-12815, 2020.

[44] A. Gulli and S. Pal “Deep learning with Keras”, *Packt Publishing Ltd*, pp. 12-15, 2017.