Journal of Scientific & Industrial Research
Vol. 84, January 2025, pp. 48-59
DOI: 10.56042/jsir.v84i1.11616

A Genetic Algorithm based Feature Selection and CNN based Ensemble Model for Intrusion Detection in IoT Smart Environments

Hidangmayum Satyajeet Sharma* & Khundrakpam Johnson Singh

Department of Computer Science and Engineering, National Institute of Technology, Langol, Imphal 795 004, Manipur, India

Received 05 June 2024; revised 29 August 2024; accepted 22 October 2024

The growing number of Internet of Things (IoT) devices has resulted in a significant surge in network attacks, frequently causing harmful and catastrophic consequences. Malicious actors may utilize these devices to infiltrate the network infrastructure by taking advantage of hardware and software weaknesses through uninterrupted internet access. Despite significant advancements in the field of network IDS (Intrusion Detection System), there is still a lack of employing intrusion detectors in IoT environments. Hence, to address this issue, a neural network model-based intrusion detection system is introduced, which can effectively detect and classify various types of attacks on IoT devices used in intelligent applications. A feature reduction technique and a hyperparameter optimization strategy to reduce both the computing time and overhead were utilized. Important features chosen via a genetic algorithm-based feature selection model are transformed into colour images for use as input to several Convolutional Neural Network (CNN) architectures, including Xception, VGG16, and VGG19 models. The suggested ensemble model, which combines Xception, VGG16, and VGG19 classifiers using a genetic algorithm to select the most relevant features, is 98.7% accurate, which is 5% better than individual classifiers. This novel approach significantly reduces false positives while cutting computational latency when compared to existing models. By optimizing both detection speed and accuracy, the proposed system enables real-time intrusion detection, offering a scalable and efficient solution for securing IoT devices in smart environments. These advancements underscore the system's potential to set a new standard in IoT security.

Keywords: Bagging ensemble, CIC IoT 2023, Deep learning, Hyperparameter optimization, Intrusion detection system

Introduction

One of the fast progressive and adaptive domains of information technology infrastructure that is widely used in many intelligent environments is the Internet of Things (IoT). These smart infrastructures start with products that are associated with everyday lives, like smart phones, and extend to smart home facilities like smart cameras, wireless sensors, smart watches, smart TVs, robot vacuum cleaners, and smart locker systems.[1,2] In spite of their widespread adoption and extensive utility, IoT devices possess remarkably limited security features.[3] In a smart environment, data must be shared among multiple smart devices on a shared network, and a security vulnerability in a single node might jeopardize the entire network infrastructure. It is projected that the volume of data generated by these devices will exceed 73.1 ZB, and the estimated amount of IoT devices will reach 7.544 billion by 2025.(4) IoT security and privacy vulnerabilities and mitigation strategies are an increasing concern in cybersecurity.[5] Thus, improving security and protecting digital assets and network infrastructure requires implementing vital safety mechanisms, tools, and techniques at various infrastructure stages.[6] Hardware or software Intrusion Detection Systems (IDSs) detect network attacks that bypass firewalls, access control, and other security measures.[7,8] IoT systems are dynamic, so rule-based IDS may miss complex attacks. Developing an effective IDS is difficult because it requires detecting an intrusion in time to mitigate damage.

In contrast to the architecture of the conventional Internet, the IoT network has three levels: the perception, network, and application layers.[9] Routers, switches and gateways transmit data from the perception layer, which comprises detectors and physical equipments to the network layer. Various mechanisms, such as bluetooth, radio frequency identification, Wi-Fi etc., are used to protect this layer. Finally, the application layer provides application functions and services for the device's operation. Therefore, monitoring the dataflow in this layer is crucial, as it transmits the majority of device

*Author for Correspondence
E-mail: satya4hidang@gmail.com

data. Furthermore, the ever-changing strategies of the intruders constitute an important challenge. The structure of an IoT network with IDS is shown in Fig. 1. When designing and IDS, different Artificial Intelligence (AI) methods are used to find threats. These include traditional machine learning models, deep learning methods, ensemble techniques, and hybrid approaches. A lot of progress has been made in deep learning lately due to the creation of Convolutional Neural Network (CNN) models. Apart from this, employing deep learning methods can significantly improve the quality of results when working with massive datasets. However, ensemble classifiers, which are composed of multiple classifiers, produce more precise results than individual classifiers.[10,11] It has the capability to reduce bias, minimize variance, and enhance the decision-making process as a whole.[12] Various ensemble methods are available, with the primary aggregators being bagging, boosting, and stacking.[13] But in a lot of their studies, they didn't take into account the problems that come with having limited computing power and finding real-time IoT threats. Therefore, calculating the time required to detect attacks is a crucial element in designing an IDS.

In response to this knowledge vacuum, a bagging ensemble-based approach to detecting and classifying IoT attacks was presented. Xception, VGG16, and VGG19 are three CNN models that rely on deep learning to build a categorization prediction layer in the proposed ensemble model. For this purpose, the UNSW-NB15 dataset and the more current CIC IoT 2023 real-time IoT attack benchmark was employed. To improve the classification process, a feature selection method based on Genetic Algorithms (GA) successfully reduces the set of features by choosing the most relevant ones. Significant contributions of the proposed approach include:

- Develop an efficient IDS model for the IoT by employing deep learning methodology and a GA technique for optimal feature selection to ensure high detection efficiency.
- Employing an automatic hyperparameter optimization approach to fine-tune the deep learning models through a random search optimization technique and thereafter comparing the evaluation metrics achieved without employing hyperparameter tuning.
- Proposed a bagging ensemble model by combining the optimized deep learning model and comparing the performance with single classifiers.
- Use two standard datasets to assess the suggested work: the CIC IoT 2023 and UNSW-NB 15 to assess the performance metrics.
- Compare the results against the individual base learners and also with other published works.

Literature Review

This section summarizes current studies that have looked at how IDS can benefit from both traditional machine learning and deep learning approaches.

Nandanwar *et al.*[14] proposes a model to classify botnet attacks in an IoT environment using a hybrid CNN and bidirectional long-short-term memory (Bi-LSTM) model. They use a mode of preprocessing steps that consist of feature encoding, data standardization, and data augmentation to add some

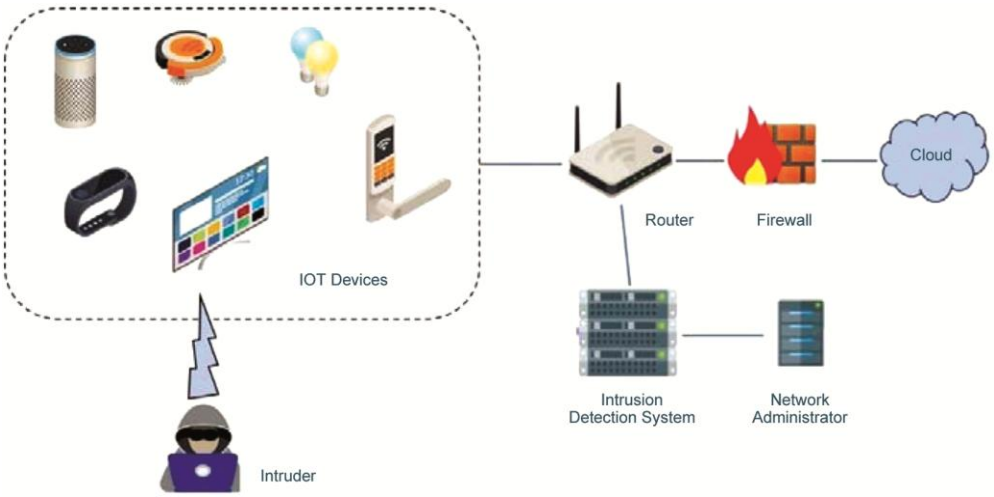

Fig. 1— Architecture of an IoT environment with IDS

noise to the dataset. They chose a hybrid deep learning model, which is part of a growing trend in IDS research that combines the strength of various deep learning models to improve detection performance.

Similarly, Selvapandian *et al*.[15] suggests an IDS model for an IoT-cloud setting that is based on deep learning. Utilizing the LeNet architecture, they emphasized on feature reduction by using two convolution layers and a pooling layer. Their model was tested using NSL-KDD. But the model's reliance on a relatively simple architecture like LeNet raises concerns about its ability to scale and defend against more complex attacks present in IoT networks.

Yongzhong *et al*.[16] introduces a layered stack ensemble method using the Fisher score for feaure selection for identifying assaults in IoT. A meta classifier with improved TPE-LightGBM predicts the final result from the output of LightGBM and Naïve Bayes as base classifiers in the stack model. Their technique is remarkable in terms of lowering training time and improving performance, but the base classifier selection could be compared with a deep learning method to assess its real-world applicability.

Mushtaq *et al*.[17] developed a stacked ensemble model with five base learners: DT, RF, bagging classifier, XgBoost, and additional trees. The multilayer perceptron functions as a meta-learner. To determine which features to submit to the meta-classifier, these five base models collaborate using a sequential forward feature selection strategy. They were able to test the model using the NSL-KDD dataset and get a high detection rate while minimizing false alarms.

Basavaraj *et al.*[18] presents an IDS to detect attacks on vehicular systems or the Internet of Vehicles (IoV). Utilizing a real-time generated CAN dataset, they implemented a DNN to illustrate the potential of the model in a real-world scenario. Although this application-oriented method is a significant improvement, it doesn't fully test the capabilities of the model because they fail to cross-validate with other datasets.

Sanju *et al*.[19] investigated an IDS model for an IoT system using a metaheuristic-based feature selection algorithm and ensemble deep learning for classification. It combines three deep learning models: a bi-directional LSTM (BiLSTM), a gated recurrent unit, and an extreme learning machine. The model was implemented using different datasets: the CIC IDS 2017, IoT 2023 and UNSW-NB15. However, the computational efficacy of their model architecture could present a challenge in real-time deployment.

Hazman *et al*.[20] put forward an IDS model to detect anomalies in the network flow of IoT-based smart environments. They present an ensemble learning model that combines the AdaBoost algorithm and the Boruta feature selection algorithm. Their findings on the Bot-IoT and NSL-KDD datasets are remarkable; however, the practicality of this ensemble method in dynamically evolving IoT environments remains an open question.

Another study by Saied *et al*.[21] provides a comparative examination of boosting techniques for IoT threat detection, demonstrating the efficacy of several boosting algorithms. They employed Adaptive, Gradient, Extreme, Light, Categorical, and Hist Gradient Boosting. This method offers an expansive view on the adaptability of boosting techniques; however, its dependence on a singular dataset, N-BaIoT, restricts the general applicability of its findings.

Kaushik *et al*.[22] proposed a method of detecting attacks in IoT by using a teaching-learning based optimization model for feature extraction, followed by random forest classification. Their model exhibited enhanced performance on the UNSW-NB15 dataset; nevertheless, the optimization approach might be further improved to minimize computational overhead.

Another approach by Kulshrestha *et al.*[23] targeted cyber-attacks on medical IoT devices using both conventional techniques and ensemble classification methods. Relevant dataset features were selected using the Extra Tree Classifier's importance score. Applications on the ToN_IoT dataset show that adaptive boosting ensemble learning outperformed other classification algorithms. However, further research is needed by comparing other boosting techniques instead of relying on a single approach.

The ensemble approach in Alotaibi *et al*.[24] used four machine learning classifiers as the basic model. They evaluated their model on the ToN-IoT dataset using both stacking and voting ensemble methods, and the results were promising. But, the robustness of the model could be further improved by incorporating advanced feature selection techniques.

In another study by Latif *et al.*[25], the authors proposed a deep learning model known as the dense random neural network that emphasize the potential of dense clustering in hidden layers to detect IoT attacks. This approach provides a novel viewpoint on

network architecture design; however, the drawbacks between model complexity and performance must be meticulously evaluated in order I facilitate practical implementation.

Hakan *et al.*[26] suggested a hybrid deep learning model for industrial IoT intrusion detection. The initial phase used min-max normalization. This hybrid model used CNN and LSTM deep learning. The model was tested for binary and multiclass classification on UNSW-NB15 and X-IIoTID. The combination model does better than single classifier models in both binary and multiclass classifiers, according to the results.

In general, machine learning models influence IDS development that improves IoT network security. Recent IDS developments show that hybrid or ensemble classifiers are used more than single classifiers. The model in Selvapandian *et al.*[15] is trained with one classifier. Also, CNNs work better with images. The authors in Nandanwar *et al.*[14] and Basavaraj *et al.*[18] ignore tabular data to image conversion. Some didn't use feature selection, which is essential before applying a machine learning model. Nandanwar *et al.*[14], Selvanpandian *et al.*[15], Latif *et al.*[25], Hakan *et al.*[26] did not use feature selection. Other state-of-the-art methods proposed by Selvanpandian *et al.*[15], Mustaq *et al.*[17], Sanju *et al.*[19] can still be improved by optimizing the model. Some validate the model with a single dataset, while others use outdated, irrelevant datasets for IoT networks. Furthermore, a significant number of models are not fully optimized, and numerous studies only verify their models using obsolete or unrelated information, therefore restricting their suitability for modern IoT deployments. This review reveals a significant gap in feature selection, image data conversion, and model optimization.

To fill these gaps, the proposed research uses a Genetic Algorithm (GA) for feature selection, image transformation for CNNs, and hyperparameter optimization. This method optimizes performance and tests the model on relevant IoT datasets, solving a major shortcoming in earlier studies. This critical study demonstrates how the suggested framework improves constraints, enabling more reliable, efficient, and real-time IDS in IoT scenarios.

Proposed Methodology

This part gives an outline of the IDS that is used to identify attacks. The primary objective of this research is to develop an ensemble bagging model that can detect hazards in an IoT setting. It has several steps, such as collecting data, preprocessing it, choosing features, and training the model using different CNN designs. The workflow of the IDS model, consisting of five stages, is depicted in Fig. 2 and is briefly elucidated below.

1. Data collection: In this step, the data from the IoT devices are collected in the form of csv data. In this case, CIC IoT 2023 and UNSW-NB15 datasets are utilized.
2. Data preprocessing: Here, the categorical data present in the dataset is encoded using label encoding. After that, an oversampling method based on SMOTE (Synthetic Minority Oversampling Technique Evaluation) was used to fix the class imbalance.
3. Feature selection and image transformation: The pre-processed dataset is used to select relevant features for classification. Tabulated features are normalized and converted to image data.
4. Data classification: This step involves training the multiple CNN models (Xception, VGG16 and VGG19) to categorize the various attacks seen in

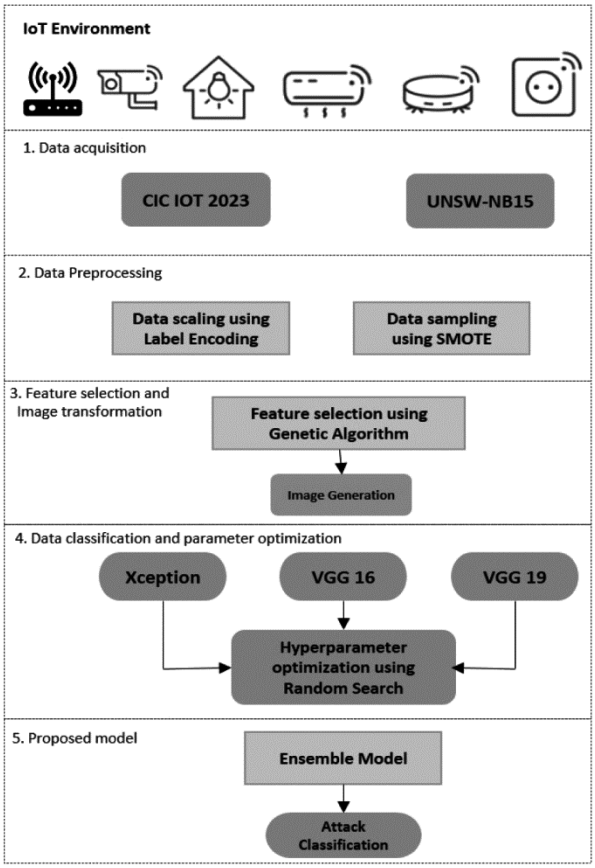

Fig. 2 — Methodology for the planned IDS in an IoT environment

the dataset. The parameters of the CNN models were optimized using a random search technique to get more effective results.

5. Proposed model: Finally, the base models are combined in order to construct an ensemble bagging model for superior outcomes.

A detailed explanation of the workflow of the IDS model is given in the following subsections.

Dataset

For the suggested model, the CIC IoT 2023 dataset and the UNSW-NB15 dataset are used, both of which are open to the public. Since both datasets are based on actual network traffic that was recorded in controlled environments, they are accurate representations of both real-world network activity and security risks.

CIC IoT 2023

One of the Canadian Institute of Cybersecurity's latest attack datasets for IoT are the CIC IoT 2023.[27] Due to the need for a lot of network equipment and human resources for infrastructure maintenance, many studies use simulated or IoT devices. A comprehensive IoT attack dataset is created with the aim of enhancing security analytics for actual IoT installations. The dataset comprises information obtained from 105 IoT equipment and 33 instances of topological attacks. It was developed utilizing real-time modeling of assaults in a large IoT network, resulting in a reliable benchmark to evaluate IDS used in a smart environment. There are seven types of attacks: DoS, DDoS, web-based, mirai, spoofing, brute force. Since the dataset is massive in size, only 5% of the total is used, with 2334300 instances and 46 unique features.

UNSW-NB 15

The UNSW-NB15 dataset is a commonly utilized and standardized labelled dataset in the domain of network intrusion detection and cybersecurity research.[28,29] It contains actual network traffic data recorded inside the dataset, which was generated in a controlled setting. It has a total of 175341 instances and 44 characteristics. Of the total, 3 are categorized and 41 are numerical. This dataset includes several forms of network attacks, including shellcode, DoS, generic, analysis, backdoor, reconnaissance, fuzzers, worms, and exploits. Aimed for assessing the efficacy of network IDS, the UNSW-NB15 dataset is a modern, genuine, complex set of data. It is useful for both academic study and real-world cybersecurity applications because it has many features and can withstand many types of attacks.

Preprocessing

The first phase in designing the proposed model is data preprocessing, which is a very crucial step before applying it to a machine learning model. The raw dataset contains null or missing values, and most of the feature set contains both categorical and numerical values. This stage involves performing various operations on unprocessed data, such as minimizing dimensions and handling missing values, in order to improve processing speed and overall performance. Both numeric and categorical data are included in the feature set. To turn the classified data into numerical data, a label encoder is employed.

The second phase of the preprocessing step is to balance the dataset. The imbalance in attack class ratios might lead to an overestimation of the majority class and a failure to detect the minority classes by the classifier. So, sampling and oversampling need to be used for balancing the dataset. Synthetic Minority Oversampling Technique Evaluation (SMOTE) is an effective imbalance class solution.[30,31] SMOTE generates synthesized instances by randomly selecting one of its k-nearest neighbours and simulating it over a line segment connecting the minority class instance to the neighbour. This process continues until the minority-majority class balance is reached.

Feature Selection

Selecting the relevant features is an important part of making malicious network flow monitoring more accurate. Eliminating features that aren't helpful for classification is the primary goal of the feature selection approach.[32] As a result, it enhances the efficacy of the model while decreasing its computational cost.[33]

This work involves the use of Genetic algorithm (GA) inspired feature reduction technique to decrease the number of features and identify the optimum subsets.[34] Genetic algorithms conduct an extensive search across the feature space with the objective of identifying feature subsets that are globally optimal or nearly optimal. They also exhibit resistance to outliers and noise present in the data by utilizing the convergent nature of the solutions produced over multiple generations. The GA based feature selection module typically works on the following steps: (a) initialization of population, (b) fitness evaluation of the population, (c) selection of parents based on their fitness score, (d) crossover to produce new offspring, (e) mutation to increase diversity, (f) replacement of some current population and, (g) final generation selection. An iterative execution of these procedures

allows the GA feature selection model to thoroughly explore the range of possible feature subsets and identify the ones that have the most significant impact on the efficiency of the machine learning framework. The feature selection approach based on GA is implemented on two distinct datasets. A total of 10 iterations were set for the experiment. Altogether, 24 features from the IoT dataset and 29 features from UNSW-NB15 were selected.

Image Transformation

This section discusses the image transformation process after the relevant features have been selected. The features selected from the datasets are represented in the form of tabular data. Converting network traffic into image format is important because CNN architectures perform better on image-based datasets.[35,36] This requires normalizing feature values to image pixel values from 0 to 255. Normalization is crucial to preprocessing because it affects machine learning model performance. Quantile normalization normalizes data distribution across the dataset after selecting features.[37] Quantile normalization reduces technical variations in samples or datasets, such as batch effects or measurement platform differences. Thus, most parameters stay near the median, which effectively manages outliers.[38]

The normalized data samples are converted to blocks depending on the attack types. After the feature selection, the CIC IoT 2023 contains 24 features. Each block will contain 72 consecutive samples, each with 24 features, which are later transformed into a colour image of shape 24 × 24 × 3. Once the images have been transformed, they are categorized according to the attack class. For instance, a picture is labelled as "Benign" if every single sample belongs to the benign class. Nevertheless, the picture sample is labelled for a certain attack class if it is associated with that class.

Lastly, the resized photos are transformed into the ImageNet size of 224 × 224 × 3, because this specific shape is associated with improved training performance in CNN architectures. Resizing an image involves altering the pixel values to fit the new dimensions. A method known as bilinear interpolation is employed to accomplish this.[39] When deriving the new pixel values, the average weighted value of the four initial pixels nearest to the target pixel is taken into account. By modifying the pixel values to accommodate the larger scale, the visual qualities of the original are maintained in the enlarged version.

Data Classification using CNN Models

After the features have been selected using the proposed GA method, three deep learning-based CNN architectures—VGG16[40], VGG19[41], and Xception[42] are implemented on the two benchmark datasets. All three of these models are based on CNNs, which have a fully connected dense layer at the end of a series of max-pooling and convolutional layers. The hidden layers of all three designs made use of the ReLu activation function, while the last dense layer made use of Softmax. With a total of sixteen levels, the VGG16 architecture includes thirteen convolutional-pooling layers for feature extraction and three fully linked dense layers for classification. The VGG19 model incorporates three dense layers after 16 convolution-pooling layers, building upon the VGG16. Depth-wise separable convolution is employed by the Xception model. It effectively resolves model efficiency and performance, rendering it highly suitable for environments with limited resources and complex image processing tasks.

Random Hyperparameter Optimization

Hyperparameter optimization is essential in machine learning to improve model performance. Hyperparameters determine deep learning model architecture, generalization, and optimization. Also, properly tuned hyperparameters reduce model training time and computational resources. Machine learning and deep learning algorithms are enhanced by automated hyperparameter tuning.[43] Dropout, frozen layers, learning rate, patience, and epochs are adjusted in the models.

In Random search Hyperparameter Optimization (RHO), the hyperparameters used to train the model are chosen at random from a predefined search space.[44] The model's efficacy is then tested using cross-validation. This procedure is repeated with random hyperparameter combinations. After a certain number of iterations, the hyperparameter configuration with the best validation set performance is chosen as the optimal set.

The Proposed Ensemble Model

The type of ensemble classifier used for this model is the bagging ensemble model. It has achieved extraordinary results despite being one of the earliest ensemble classifiers and being relatively simple to implement.[17] Furthermore, they are known for their high computing efficiency and their capacity to operate optimally on imbalanced data.[45] The suggested ensemble bagging model that is utilized in

this work is illustrated in Fig. 3. The bagging ensemble takes the predictions from many instances of a basic learning algorithm that have been trained on different datasets and combines that to produce the final prediction.

The proposed model utilizes three optimized pre-trained CNN models: VGG16, VGG19 and Xception as base learners to make predictions on the batch of test images. A voting mechanism is employed to aggregate the predictions of all three models, and a final result is obtained. Bagging mitigates the issue of overfitting by calculating the mean of the variance provided through the use of different subsets of the data and base learners. Mathematically, the proposed model can be described as:

Each batch of image B_1, B_2, B_3, containing a subset of the images is passed through a different deep learning model. Let $f_1(B)$, $f_2(B)$, $f_3(B)$ represent the output predictions of the Vgg16, vgg19 and xception model respectively, where $B = B_1 + B_2 + B_3$. The predictions of the models are aggreagated using a voting mechanism. models can use a voting mechanism to choose the most probable class. The class that receives the majority of votes is then selected as the final prediction. The final prediction y for each image b_i, is given in Eq. (1):

$$y(b_i) = \text{mode}\left(\{\text{argmax}(f_1(b_i)), \text{argmax}(f_2(b_i)), \text{argmax}(f_3(b_i))\}\right) \quad \ldots (1)$$

The final output of the model is a set of predictions denoted by Y, as given in Eq. (2):

$$Y = \{y(b_1), y(b_2), \ldots, y(b_n)\} \quad \ldots (2)$$

where, $y(b_i)$ represents the predicted label for the corresponding input image b_i.

Experimental Evaluations and Discussion

This part presents an analysis of the suggested ensemble model, as well as a thorough examination of the base learners. Experimental trials were carried out on the CIC IoT 2023 and UNSW-NB15 datasets, allocating 70% of the data for training purposes and the remaining 30% for testing.

Performance Metrics

A range of performance indicators, including recall, accuracy, precision, F1-score, detection rate, and time complexity, was utilized to evaluate the effectiveness of the IDS model. Four inputs that include true positive (TT), true negative (TF), false positive (FF), and false negative (FT) are used to determine these measures. The overall performance of the model is positively affected by both TP and TN, which are important in assessing the reliability of the model in correctly classifying events into positive and negative categories. By correctly identifying negative cases, TN improves the model's dependability, while FN reveals the limitations of positively detecting instances. The performance measures are determined by these four metrics, as illustrated below:

Accuracy: As stated in Eq. (3), the number of correctly described instances relative to the overall number of instances determines the accuracy of a model.

$$Accuracy = \frac{TT + TF}{TT + TF + FF + FT} \quad \ldots (3)$$

Precision: Indicated in Eq. (4), it is the percentage of accurate positive projections among all the positive instances generated by the method.

$$Precision = \frac{TT}{TT + FF} \quad \ldots (4)$$

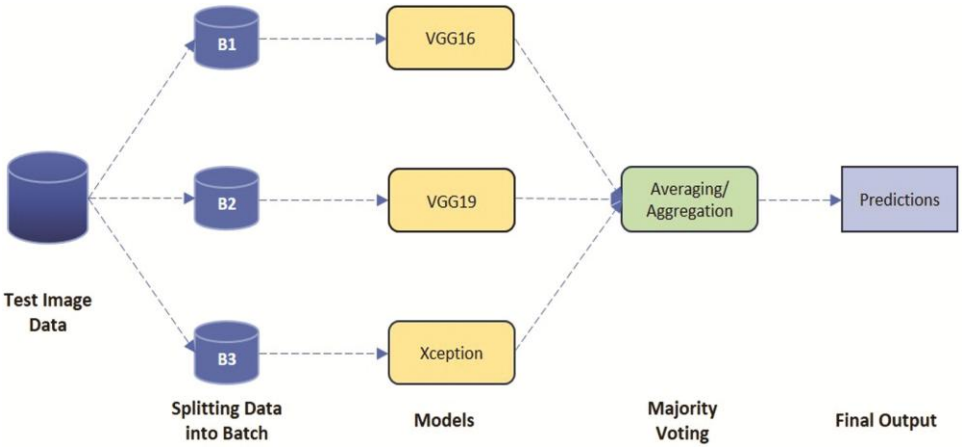

Fig. 3 — Layout of the proposed ensemble model

Recall: Indicated in Eq. (5), it measures the fraction of positive outcomes in the dataset that were accurately detected as predictions.

$$Recall = \frac{\text{TT}}{\text{TT}+\text{FT}} \quad \ldots (5)$$

F1-Score: As demonstrated in Equation (6), it is the harmonic mean of recall and precision that aids in achieving a balance between the precision and recall.

$$F1 - Score = \frac{2 \times \text{Precision} \times \text{Recall}}{Precision + Recall} \quad \ldots (6)$$

Detection rate: Indicated in Eq. (7), it takes into account all positive cases and finds the proportion of positive occurrences that a classification model effectively identifies.

$$Detection\ Rate = \frac{\text{TT}}{\text{TT}+\text{FT}} \quad \ldots (7)$$

Time complexity: It is related to the computation resources as well as hyperparameter tuning that are necessary for the training and deployment of models. This evaluation is essential as it establishes the feasibility of using a certain algorithm on a dataset of a specified dimension and complexity in a resource constrained environment.

Evaluation in CIC IoT 2023

The ensemble approach was evaluated by employing deep learning models as base classifiers on the CIC IoT 2023 dataset, which is based on real-time offensive data. A total of 24 features were selected while applying the GA-based feature selection method. To begin, standard hyperparameter configurations of VGG16, VGG19, and Xception are used to evaluate the performance of the proposed model in detecting attacks. Next, using pre-trained base classifiers, an ensemble model is constructed and subsequently evaluated. An accuracy of 99.18%, precision of 99.19%, recall of 99.18%, F1 score of 98.92%, and detection rate of 84.49% can be attained by the ensemble model. Moreover, the time complexity of the ensemble model outperforms that of other base classifiers, resulting in an execution time of 30.34 seconds. The main hyperparameters of the base models were optimized by the random optimization approach to build an optimal model. For every base model, Table 1 lists the ideal hyperparameters that were found. The hyperparameters are trained by modifying the default values to the parameters that optimise performance. The table clearly shows that the tuned models outperform the default model. Moreover, the suggested ensemble method could reach accuracy of 99.18%, precision of 99.74%, recall of 99.72%, F1-score of 99.72%, detection rate of 94.87%, and execution time of 30.36 seconds. A comparative analysis of the proposed ensemble bagging model and the base models used in the proposed work is given in Table 2. The comparison of model's accuracy is depicted in Fig. 4. It demonstrates that hyperparameter optimization may be used to improve the model performance.

Evaluation in UNSW-NB15

A completely different dataset known as the UNSW-NB15 was utilized to evaluate and enhance the capabilities of the proposed model. Experiments

Table 1— Optimal settings for the hyperparameters of the base classifiers in CIC IoT 2023

Parameters	Classifier	Optimum Value
Epochs	Vgg16	20
	Vgg19	10
	Xception	10
Dropout percent	VGG16	0.4
	Vgg19	0.4
	Xception	0.6
Learning rate	Vgg16	0.004
	Vgg19	0.006
	Xception	0.003
Frozen	Vgg16	17
	Vgg19	17
	Xception	16
Patience value	Vgg16	4
	Vgg19	2
	Xception	2

Table 2 — Performance analysis of the proposed model against other base classifiers used in this work on CIC IoT 23

Model	Accuracy	Precision	Recall	F1 Score	Detection rate	Time
Xception	99.32	99.40	99.31	99.29	91.94	2 min 38 Sec
Vgg16	98.32	97.79	98.31	97.73	74.60	2 min 18 Sec
Vgg19	98.63	97.88	98.63	98.25	78.23	2 min 19 Sec
Ensemble	99.18	99.19	99.18	98.92	84.49	30.34 sec
Xception + RHO	99.50	99.53	99.49	99.47	92.14	2 min 42 sec
Vgg16 + RHO	99.86	99.88	99.86	99.86	97.32	2 min 19 sec
Vgg19 + RHO	99.59	99.58	99.59	99.55	92.63	2 min 21 sec
Proposed	**99.73**	**99.74**	**99.72**	**99.71**	**94.87**	**30.36 sec**

were done with and without hyperparameter tuning. The ensemble model without the use of an optimization algorithm could obtain an accuracy of 93.61%, 94.33% precision, 93.61% recall, 93.30% f1-score, 82.63% detection rate and an execution time of 15.91 seconds. Later, the random optimization technique is applied to the base classifiers prior to the implementation of the ensemble model. The list of optimal hyperparameters that were obtained using the RHO is indicated in Table 3. The adoption of a RHO algorithm can enhance the performance of the model, as exhibited in Table 4. The suggested ensemble model outperforms the baseline model in terms of overall accuracy of 95.33%, precision of 94.9%, recall of 95.2%, f1-score of 95.2%, detection rate of 88.88%, and execution time of 15.26 seconds. A graphical representation of accuracy comparison with various models is depicted in Fig. 5.

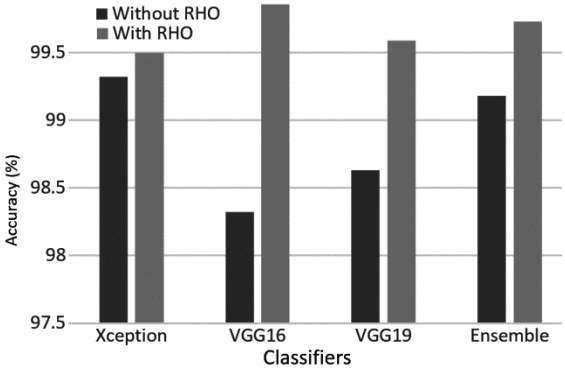

Fig. 4 — Accuracy comparison on CIC IoT 2023 dataset

Table 3 — Optimal settings for the hyperparameters of the base classifiers in UNSW-NB15

Parameters	Model	Optimal Value
Epochs	VGG16	15
	Vgg19	10
	Xception	15
Dropout percent	VGG16	0.4
	Vgg19	0.5
	Xception	0.5
Learning rate	VGG16	0.003
	Vgg19	0.006
	Xception	0.005
Frozen	VGG16	16
	Vgg19	17
	Xception	17
Patience value	VGG16	4
	Vgg19	2
	Xception	3

Performance Analysis

In this study, various classifiers including deep learning methods were assessed using an optimizer. GA-based feature selection helped remove irrelevant and redundant instances that could affect classifier performance. Additionally, feature reduction speeds up tabular data-to-image conversion. Implementation results show that random search hyperparameter optimization improves the performance of VGG16, VGG19, and Xception base CNN classifiers. The random search hyperparameter optimization (RHO) technique efficiently traverses a large search space, sampling a wide range of solutions and increasing the likelihood of finding global optima or high-performance solutions. The ensemble model with three base classifiers performs well in both datasets. It predicts attack types quickly and accurately. As depicted in Table 2 and Table 4, the ensemble model executes faster than the other classifiers in both the CIC IoT 2023 and UNSW-NB15 datasets. The

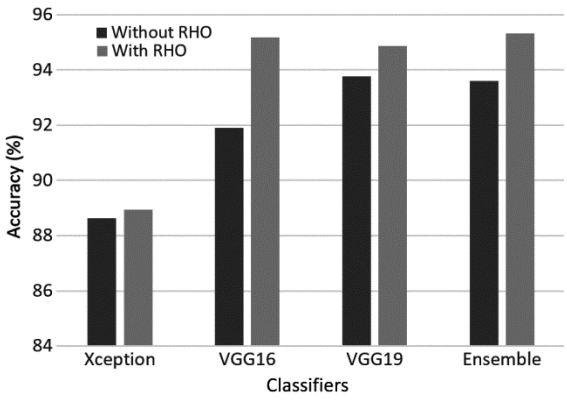

Fig. 5 — Accuracy comparison on UNSW-NB15 dataset

Table 4 — Performance analysis of the proposed model against the base classifiers used in this work using UNSW-NB15

Model	Accuracy	Precision	Recall	F1 Score	Detection rate	Time
Xception	88.63	89.48	88.63	88.29	76.58	49.7 sec
Vgg16	91.90	90.93	91.90	91.06	76.55	41.1 sec
Vgg19	93.77	94.77	93.76	93.92	88.52	40.3 sec
Bagging Ensemble	93.61	94.33	93.61	93.30	82.63	15.91 sec
Xception + RHO	88.94	89.96	88.94	88.62	78.05	48.7 sec
Vgg16 + RHO	95.17	95.44	95.17	95.22	90.40	41 sec
Vgg19 + RHO	94.86	95.26	94.85	94.81	88.71	41 sec
Proposed	**95.33**	**95.97**	**95.32**	**95.32**	**88.88**	**15.26 sec**

Table 5 — Comparison of the proposed ensemble model with existing models in several IDS utilized in the IoT environment

Reference	Method	Dataset	Feature Selection	HPO	Accuracy
Nandanwar et al.[14]	CNN + BiLSTM	N_BaIoT	—	—	99.52%
Selvapandian et al. [15]	LeNet	NSL-KDD	—	—	96.28%
Mustaq et al. [17]	Stacked Ensemble	NSL-KDD	Sequential Forward Feature Selection	—	88.10 %
Sanju et al.[19]	Ensemble	IoT 2023	HHO-EFDM	—	98.12%
Hazman et al.[20]	Ensemble blending technique	IoT 2023	Mean Decrease in Impurity	Manually set	99.51%
Alotaibi et al.[24]	Ensemble model	ToN-IoT	Coefficient Correlation, Mutual Information, K-best feature	—	98.64%
Latif et al.[25]	Dense Random Neural Network	ToN_IoT	—	Manually set	99.05%
Hakan et al.[26]	CNN with LSTM	UNSW-NB15	—	Manually set	92.90
Maghrabi et al.[46]	Random Forest	UNSW-NB15	Pearson's correlation coefficient	—	90.17%
Proposed Model	**Ensemble Model**	**CIC IoT 2023**	**GA**	**RHO**	**99.73%**
	Ensemble Model	**UNSW-NB15**	**GA**	**RHO**	**95.33%**

detection rate for each attack class in both datasets is significant. The proposed strategy is contrasted in Table 5 with various modern IDS models for IoT systems. From the analysis, it can be inferred that the suggested work outperforms other comparable models. An ensemble model can improve accuracy and robustness by reducing overfitting and variance through the combination of predictions from multiple base learners. Moreover, the use of the RHO technique helps in maximizing the efficacy of the deep learning models by identifying the optimal hyperparameters of each of the base learners.

Conclusions

The ensemble bagging IDS model for IoT smart environments demonstrates significant potential for enhancing detection speed and accuracy. In contrast to conventional models, this innovative methodology significantly lowers false positives and minimizes execution time. The proposed system facilitates real-time intrusion detection by optimizing both detection speed and accuracy, providing a scalable and efficient solution for the security of IoT devices in smart environments. However, limitations include high computational resource demands, which may hinder deployment on resource-constrained IoT devices, and the need for further validation in dynamic or novel attack scenarios.

Furthermore, the IDS model has the potential to be applied to the security of IoT gateways, smart city infrastructures, healthcare devices, and industrial IoT systems. It has the potential to significantly enhance the security of the increasingly interconnected IoT networks by playing a critical role in real-time detection and prevention of intrusions with further advancements.

Conflict of Interest

The authors have no conflict of interest to declare.

References

1 Sisinni E, Saifullah A, Han S, Jennehag U & Gidlund M, Industrial internet of things: Challenges, opportunities, and directions, *IEEE Trans Industr Inform*, **14** (2018) 4724–4734, doi: 10.1109/TII.2018.2852491.

2 Vinoth R, Deborah L J, Vijayakumar P & Gupta B B, An anonymous pre-authentication and post-authentication scheme assisted by cloud for medical IoT environments, *IEEE Trans Netw Sci Eng*, **9** (2022) 3633–3642, doi: 10.1109/TNSE.2022.3176407.

3 Pour M S, Mangino A, Friday K, Rathbun M, Bou-Harb E, Iqbal F, Samtani S, Crichigno J & Ghani N, On data-driven curation, learning, and analysis for inferring evolving internet-of-Things (IoT) botnets in the wild, *COSE*, **91** (2020), https://doi.org/10.1016/j.cose.2019.101707.

4 Zhou W, Jia Y, Peng A, Zhang Y & Liu P, The effect of IoT new features on security and privacy: New threats, existing solutions, and challenges yet to be solved, *IEEE Internet Things J*, **6** (2019) 1606–1616, doi: 10.1109/JIOT.2018.2847733.

5 Abdel-Basset M, Chang V, Hawash H, Chakrabortty R K & Ryan M, Deep-IFS: Intrusion detection approach for industrial internet of things traffic in fog environment, *IEEE Trans Industr Inform*, **17** (2021) 7704–7715, doi: 10.1109/TII.2020.3025755.

6 Kannari P R, Chowdary N S & Biradar R L, An anomaly-based intrusion detection system using recursive feature elimination technique for improved attack detection, *Theor Comput Sci*, **931** (2022) 56–64, https://doi.org/10.1016/j.tcs.2022.07.030.

7 Ashfaq R A R, Wang X, Huang J Z, Abbas H & He Y, Fuzziness based semi-supervised learning approach for

intrusion detection system, *Inf Sci*, **378** (2017) 484–497, https://doi.org/10.1016/j.ins.2016.04.019.
8 Kim G, Lee S & Kim S, A novel hybrid intrusion detection method integrating anomaly detection with misuse detection, *Expert Syst Appl*, **41** (2014) 1690–1700, https://doi.org/10.1016/j.eswa.2013.08.066.
9 Sharma B, Sharma L, Lal C & Roy S, Anomaly based network intrusion detection for IoT attacks using deep learning technique, *Comput Electr Eng*, **107** (2023), https://doi.org/10.1016/j.compeleceng.2023.108626.
10 Aburomman A A & Reaz M B, A survey of intrusion detection systems based on ensemble and hybrid classifiers, *COSE*, **65** (2017) 135–152, https://doi.org/10.1016/j.cose.2016.11.004.
11 Zhou Z H, *Ensemble methods: Foundations and algorithms* (CRC Press) 2012, 236 https://doi.org/10.1201/b12207.
12 Lazzarini R, Tianfield H & Charissis V, A stacking ensemble of deep learning models for IoT intrusion detection, *Knowl-Based Syst*, **279** (2023) 110941 https://doi.org/10.1016/j.knosys.2023.110941.
13 Ren Y, Zhang L & Suganthan P N, Ensemble classification and regression-recent developments, applications and future directions, *IEEE Comput Intell Mag*, **11** (2016) 41–53, doi: 10.1109/MCI.2015.2471235.
14 Nandanwar H & Katarya R, TL-BILSTM IoT: transfer learning model for prediction of intrusion detection system in IoT environment, *Int J Inf Secur*, **23** (2024) 1251–1277, https://doi.org/10.1007/s10207-023-00787-8.
15 Selvapandian D & Santhosh R, Deep learning approach for intrusion detection in IoT-multi cloud environment, *Autom Softw Eng*, **28(2)** (2021) 19, https://doi.org/10.1007/s10515-021-00298-7.
16 Cao Y, Wang Z, Ding H, Zhang J & Li B, An intrusion detection system based on stacked ensemble learning for IoT network, *Comput Electr Eng*, **110** (2023) 108836, https://doi.org/10.1016/j.compeleceng.2023.108836.
17 Mushtaq E, Zameer A & Khan A, A two-stage stacked ensemble intrusion detection system using five base classifiers and MLP with optimal feature selection, *Microprocess Microsyst*, **94** (2022) 104660, https://doi.org/10.1016/j.micpro.2022.104660.
18 Basavaraj D & Tayeb S, Towards a lightweight intrusion detection framework for in-vehicle networks, *J Sens Actuator Netw*, ***11*** (2022) 6, https://doi.org/10.3390/jsan11010006.
19 Sanju P, Enhancing intrusion detection in IoT systems: A hybrid metaheuristics-deep learning approach with ensemble of recurrent neural networks, *J Eng Res*, **11** (2023) 356–361, https://doi.org/10.1016/j.jer.2023.100122.
20 Hazman C, Guezzaz A, Benkirane S & Azrour M, Toward an intrusion detection model for IoT-based smart environments. *Multimed Tools Appl*, **83** (2023) 62159–62180, https://doi.org/10.1007/s11042-023-16436-0.
21 Saied M, Guirguis S & Madbouly M, A comparative study of using boosting-based machine learning algorithms for IoT network intrusion detection, *Int J Comput Intell Syst*, **16(1)** (2023) 177, https://doi.org/10.1007/s44196-023-00355-x.
22 Kaushik A & Al-Raweshidy H, A novel intrusion detection system for internet of things devices and data, *Wireless Netw*, **30** (2024) 285–294, https://doi.org/10.1007/s11276-023-03435-0.
23 Kulshrestha P & Vijay Kumar T V, Machine learning based intrusion detection system for IoMT, *Int J Syst Assur Eng Manag*, **15** (2023) 1802–1814, https://doi.org/10.1007/s13198-023-02119-4.
24 Alotaibi Y & Ilyas M, Ensemble-learning framework for intrusion detection to enhance internet of things' devices security, *Sens*, ***23*** (2023**)** 5568, https://doi.org/10.3390/s23125568.
25 Latif S, Huma Z, Jamal S S, Ahmed F, Ahmad J, Zahid A, Dashtipour K, Aftab M U, Ahmad M & Abbasi Q H, Intrusion detection framework for the internet of things using a dense random neural network, *IEEE Trans Industr Inform*, **18** (2022) 6435–6444, doi: 10.1109/TII.2021.3130248.
26 Hakan C A & Zafer A, A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks, *Eng Sci Tech Int J*, **38** (2023) 101322, https://doi.org/10.1016/j.jestch.2022.101322.
27 Neto E C P, Dadkhah S, Ferreira R, Zohourian A, Lu R & Ghorbani A A, CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment, *Sens*, ***23*** (2023) 5941, https://doi.org/10.3390/s23135941.
28 Moustafa N & Slay J, The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set, *Inf Secur J A Glob Perspect*, **25(1-3)** (2016) 18–31, https://doi.org/10.1080/19393555.2015.1125974.
29 Kasongo S M & Sun Y, Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset, *J Big Data*, 7(1) (2020) 105, https://doi.org/10.1186/s40537-020-00379-6.
30 Srinivasan M & Senthilkumar N C, Class imbalance data handling with optimal deep learning-based intrusion detection in IoT environment, *Soft Comput*, **28** (2024) 4519–4529, https://doi.org/10.1007/s00500-023-09610-x.
31 Zakariah M, AlQahtani S A & Al-Rakhami M S, Machine learning-based adaptive synthetic sampling technique for intrusion detection, *Appl Sci*, ***13*** (2023) 6504, https://doi.org/10.3390/app13116504.
32 Sharma H S & Singh K J, A feed forward deep neural network model using feature selection for cloud intrusion detection system, *Concurr Comput Pract Exp*, **36(9)** (2024) e8001, doi: 10.1002/cpe.800.
33 Sharma B, Sharma L & Lal C, Feature selection and deep learning technique for intrusion detection system in IoT, *Int Conf Comput Intell, Algo Intell Sys* (Springer) 2022, 252–261 https://doi.org/10.1007/978-981-16-3802-2_21.
34 Halim Z, Yousaf M N, Waqas M, Sulaiman M, Abbas G, Hussain M, Ahmad I & Hanif M, An effective genetic algorithm-based feature selection method for intrusion detection systems, *COSE*, **110** (2021) 102448, https://doi.org/10.1016/j.cose.2021.102448.
35 Hussain F, Abbas S G, Husnain M, Fayyaz U U, Shahzad F & Shah G A, IoT DoS and DDoS attack detection using ResNet, *2020 IEEE 23rd INMIC*, Bahawalpur, Pakistan (IEEE) 2020, 1–6, doi: 10.1109/INMIC50486.2020.9318216.
36 Sharma H S & Singh K J, Intrusion detection system: A deep neural network-based concatenated approach, *J Supercomput*ing, **80** (2024) 13918–13948, https://doi.org/10.1007/s11227-024-05994-1.
37 Latif S, Boulila W, Koubaa A, Zou Z & Ahmad J, DTL-IDS: An optimized Intrusion Detection Framework using deep transfer learning and genetic algorithm, *J Netw Comput Appl*, **221** (2024) 103784, https://doi.org/10.1016/j. jnca. 2023.103784.
38 Lokman S F, Othman A T, Bakar M H A & Musa S, The Impact of different feature scaling methods on intrusion detection for in-vehicle controller area network (CAN),

ACeS 2019 *Commun Comput Inf Sci*, **1132** (2020) 195–205, https://doi.org/10.1007/978-981-15-2693-0_14.

39 Ye H, Su K & Huang S, Image enhancement method based on bilinear interpolating and wavelet transform, *IEEE 5th Adv Inf Tech, IAEAC* (IEEE) 2021, 1147–1150, doi: 10.1109/IAEAC50856.2021.9390624.

40 Simonyan K & Zisserman A, Very deep convolutional networks for large-scale image recognition, (2015), https://doi.org/10.48550/arXiv.1409.1556.

41 Bansal M, Kumar M, Sachdeva M & Mittal A, Transfer learning for image classification using VGG19: Caltech-101 image data set, *J Ambient Intell Hum Comput*, **14** (2023) 3609–3620, https://doi.org/10.1007/s12652-021-03488-z.

42 Chollet F, Xception: Deep learning with depthwise separable convolutions, Proceedings of the *IEEE conference on computer vision and pattern recognition* (IEEE) 2017, 1251-1258.doi: 10.1109/CVPR.2017.195.

43 Feurer M & Hutter F, Hyperparameter optimization, *Automated Machine Learning, The Springer Series on Challenges in Machine Learning*, (Springer) 2019, 3–33, https://doi.org/10.1007/978-3-030-05318-5_1.

44 Kunang Y N, Nurmaini S, Stiawan D & Suprapto B Y, Attack classification of an intrusion detection system using deep learning and hyperparameter optimization, *J Inf Secur Appl*, **58** (2021) 102804, https://doi.org/10.1016/j.jisa.2021.102804.

45 Zareapoor M & Shamsolmoali P, Application of credit card fraud detection: based on bagging ensemble classifier, *Procedia Comput Sci*, **48** (2015) 679–685, https://doi.org/10.1016/j.procs.2015.04.201.

46 Maghrabi L A, Automated network intrusion detection for internet of things: Security enhancements, *IEEE Access*, **12** (2024) 30839–30851, doi: 10.1109/ACCESS.2024.3369237.