Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.051539

ARTICLE

A Robust Approach for Multi Classification-Based Intrusion Detection through Stacking Deep Learning Models

Samia Allaoua Chelloug*

Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh, 11671, Saudi Arabia

*Corresponding Author: Samia Allaoua Chelloug. Email: SAChelloug@pnu.edu.sa

Received: 08 March 2024 Accepted: 29 April 2024 Published: 20 June 2024

ABSTRACT

Intrusion detection is a predominant task that monitors and protects the network infrastructure. Therefore, many datasets have been published and investigated by researchers to analyze and understand the problem of intrusion prediction and detection. In particular, the Network Security Laboratory-Knowledge Discovery in Databases (NSL-KDD) is an extensively used benchmark dataset for evaluating intrusion detection systems (IDSs) as it incorporates various network traffic attacks. It is worth mentioning that a large number of studies have tackled the problem of intrusion detection using machine learning models, but the performance of these models often decreases when evaluated on new attacks. This has led to the utilization of deep learning techniques, which have showcased significant potential for processing large datasets and therefore improving detection accuracy. For that reason, this paper focuses on the role of stacking deep learning models, including convolution neural network (CNN) and deep neural network (DNN) for improving the intrusion detection rate of the NSL-KDD dataset. Each base model is trained on the NSL-KDD dataset to extract significant features. Once the base models have been trained, the stacking process proceeds to the second stage, where a simple meta-model has been trained on the predictions generated from the proposed base models. The combination of the predictions allows the meta-model to distinguish different classes of attacks and increase the detection rate. Our experimental evaluations using the NSL-KDD dataset have shown the efficacy of stacking deep learning models for intrusion detection. The performance of the ensemble of base models, combined with the meta-model, exceeds the performance of individual models. Our stacking model has attained an accuracy of 99% and an average F1-score of 93% for the multi-classification scenario. Besides, the training time of the proposed ensemble model is lower than the training time of benchmark techniques, demonstrating its efficiency and robustness.

KEYWORDS

Intrusion detection; multi classification; deep learning; stacking; NSL-KDD

1 Introduction

Presently, information is extensively exchanged between interconnected systems. Consequently, ensuring the protection of digital assets has become a prime challenge. In particular, IDSs function as vigilant guardians for recognizing and responding to illegitimate activities within a network or

system. By analyzing network traffic, an IDS can identify signs of compromise, malicious activity, or policy violations [1]. The primary objective of an IDS is to provide early warning and swift response capabilities. By promptly detecting and alerting to threats, organizations can improve their decision making to mitigate risks, minimize damages, and restore the integrity and confidentiality of their systems and data. More specifically, an IDS acts as an additional layer of defense and contributes to ensuring security measures along with other solutions, including antivirus, firewalls, and access controls. By implementing robust IDSs, organizations can also ensure their commitment to security and meet regulatory obligations. Modern IDSs leverage artificial intelligence to provide an accurate solution for predicting attacks [2]. These techniques can learn from vast amounts of data, enabling them to identify indicative patterns of well-known and emerging threats. Among many powerful methods within deep learning, stacking has emerged as a potent approach for increasing models performance. It is a meta-learning technique that combines the predictions of multiple individual models, called base models or learners, to create a more robust and accurate ensemble model [3]. The underlying principle of stacking is to leverage the diverse strengths and weaknesses of separate models to enhance the overall predictive performance. More precisely, by combining the predictions resulting from multiple models, stacking can potentially overcome the limitations of any single model and provide a more comprehensive and reliable prediction. One of the key advantages of stacking is its ability to learn existing complex relationships and patterns within a dataset, while handling non-linearities. This versatility makes stacking particularly useful in situations where the underlying problem is challenging, the dataset is limited, or the features are complex and heterogeneous. Most related works have addressed the problem of intrusion detection as a binary problem. However, the considered classifiers cannot distinguish between different types of attacks. Furthermore, the state-of art works that deal with stacking have explored machine learning techniques that are limited to small datasets.

Thus, this paper focuses on stacking two deep learning models for multi-classification of NSL-KDD dataset attacks [4]. This approach differs from existing state-of art techniques that predominantly rely on stacking machine learning models. Therefore, the designed model includes two parallel branches. The first branch performs prediction using CNN, while the second one relies on DNN for predicting attacks. After that, a meta-learner that combines the resulting predictions is executed to generate the final prediction based on the Softmax layer that assigns attacks' labels. In order to overcome the problem of the NSL-KDD dataset imbalance, we utilize Synthetic Minority Oversampling Technique (SMOTE) [5]. The utilization of this particular technique results in improving the performance of the proposed CNN and DNN models. The contribution employed in our paper capitalizes on the advantageous effects of stacking deep learning models when applied to the multi-classification of attacks, resulting in significantly improved accuracy for recognizing all attacks. The original contributions of our study are listed below:

- The NSL-KDD dataset has been pre-processed, and SMOTE has been adopted to avoiding the models' overfitting as the number of instances of Remote2Local (R2L) and User2Root (U2R) is significantly smaller than the number of instances of normal, denial of service (DoS), and Probe attacks.
- We have designed and trained two deep learning models for handling the multi-classification of the NSL-KDD dataset. The CNN model depends on two consecutive blocks of convolution, max pooling and dropout. Then, the proposed CNN relies on a flatten and fully connected layers. The output layer of the proposed CNN investigates the Softmax function to generate the required prediction. On the other hand, the proposed DNN relies on three fully connected layers and an output layer utilizing the Softmax function.

- The results of the proposed CNN and DNN have been analyzed from many perspectives, including the confusion matrix, the classification report, and the training and validation curves for accuracy and loss.
- We noticed that each individual model has weaknesses for detecting all at-tacks. Therefore, this paper proposes stacking the proposed CNN and DNN to improve the prediction accuracy in the NSL-KDD dataset and the recognition rate for all types of attacks.

2 Related Works

An IDS involves detecting and alerting on events that may indicate potential security breaches or violations of security policies. Host-based Intrusion Detection System (HIDS) that is installed on individual hosts or servers to monitor their activities and detect any suspicious behavior or attacks. On another hand, Network-based Intrusion Detection System (NIDS) is deployed at strategic points within a network to monitor and analyze network traffic. NIDS can detect attacks targeting multiple hosts within a network and provide a broader view of network wide activities [6]. IDSs work based on several detection methods including signature-based detection that applies mapping between a database of attack signatures and network traffic [7]. A competitive approach concerns anomaly-based detection that is efficient for detecting novel attacks but may also generate false positives if network activities differ from the established baseline [7]. Another approach consists to build profiles for normal behavior for identifying attacks that deviate from normal behavior.

The authors of paper [8] have analyzed three techniques including decision tree, support vector machine (SVM) and backpropagation neural networks. Their results demonstrate that the decision tree has attained the best accuracy for detecting intrusion followed by backpropagation neural network that generated more steady results than SVM while considering 60% and 70% of training ratios of total instances. However, SVM achieved higher accuracy than backpropagation when considering 60% of training ratio.

In [9], a hybrid approach that provides more promising performance for multi-classification based-intrusion detection has been proposed. Hence, the proposed double layered hybrid approach (DLHA) has been designed to accurately detect all types of attacks, especially unfrequented attacks including R2L and U2R. More specifically, the proposed approach relies on Naive Bayes classifier in the first layer to detect DoS and Probe, SVM is investigated in the second layer to differentiate R2L and U2R from normal instances. Compared to the benchmark techniques, the approach described in [9] has achieved an intrusion detection rate of 96.67% and 100% for R2L and U2R, respectively.

In paper [10], a robust IDS that deals with modern attacks has been developed. The latter balances CICIDS 2017 dataset using SMOTE. Besides, four machine learning algorithms have been implemented and evaluated. The results discussed in [10] show that random forest (RF) classifier outperforms other machine learning model by achieving a detection rate of 97% demonstrating its performance to detect different attacks.

The research presented in [11] has focused on intrusion detection in IoT. Thus, data pre-processing and reduction of dimensionality have been performed before applying a set of machine learning techniques. The authors of [11] have concluded that PCA-XgBoost method that adopts principal component analysis (PCA) for dimensionality reduction and XgBoost for classification has provided the highest accuracy in terms of accuracy, precision, F1, and Mathew Correlation Coefficient (MCC).

The authors of [12] have proposed an idea for evaluating unsupervised machine learning methods on two IoT datasets, while performing inter-dataset evaluation strategy. In particular, the autoencoder,

SVM, and isolation forest have been tested for two similar IoT datasets that have been split into training, testing and validation sets. In addition, the instances of the training set from the first dataset have been trained, and the validation set has been employed to select the best model with its corresponding hyperparameters. According to the results presented in [12], all models demonstrated high classification scores on an individual dataset, but failed to directly transfer those to a second unseen but related dataset.

In recent years, there has been a growing interest in developing effective techniques for intrusion detection in computer networks. Two popular approaches that have been extensively studied and combined are metaheuristics and deep learning. Metaheuristics are optimization algorithms that can be used to search for optimal or near optimal solutions in complex problem spaces, while deep learning is a sub field of machine learning that focuses on learning hierarchical representations of data. Researchers have explored various ways to integrate metaheuristics and deep learning to improve the accuracy and efficiency of IDSs:

- Hybrid Models: One common approach is to develop hybrid models that combine the strengths of both metaheuristics and deep learning. For example, some researchers have proposed to combine metaheuristics with deep neural networks [13–15]. These metaheuristics are used to optimize the architecture or parameters of the neural network to improve its performance in detecting network intrusions. Metaheuristics have been employed to select the optimal subset of features for training deep learning models, reducing the dimensionality and improving the overall performance of the IDS.
- Transfer Learning: Involves leveraging knowledge learned from one task to improve the performance on another related task. Researchers of papers [16,17] have explored the use of transfer learning in the context of intrusion detection, where metaheuristics are used to transfer knowledge between different deep learning models. This approach aims to enhance the generalization capability of the models, particularly in scenarios where labeled training data is limited.
- Ensemble Methods: Combine multiple models to make collective decisions, often resulting in improved performance compared to individual models. Metaheuristics have been employed to optimize the ensemble of deep learning models for intrusion detection. This optimization process involves selecting the best combination of models, determining their weights, or adapting their architectures to achieve better accuracy, robustness, or efficiency.

It is worth mentioning that the use of stacked models for intrusion detection involves training several different models and then combining their predictions to make a final decision. This approach aims to improve the overall performance and robustness of the IDS by leveraging the strengths of different models. There have been several research works that have used stacked models for the NSL-KDD dataset. These works typically involve training a diverse set of base models, such as decision trees, SVM, RF, or neural networks, and then combining their predictions using techniques like majority voting or weighted averaging.

The idea proposed in [18] consists to combine different machine learning models to optimize the classification performance of NSL-KDD dataset. More precisely, the predictions obtained by gradient boosting machine (GBM) and RF algorithm have been stacked and fed as an input for the meta-classifier that relies on K-fold cross validation to minimize the error. Compared to bagging and boosting, the proposed stacked model achieves better performance. One more advantage of the proposed stacked model is that the detection rate of each type of attack is higher than the detection rate obtained through ANN, CNN, and recurrent neural network (RNN).

In paper [19], the authors have designed and implemented a stacked model that incorporates the decision of 10 cross-validation of KNN, decision tree, RF and provides a final decision through a simple DNN consisting of two fully connected layers and Softmax function. The proposed model has been evaluated for NSL-KDD. Compared to single classifiers, the developed deep stacked model improved the overall classification accuracy and also enhances the multi-classification accuracy for detecting all attacks.

The authors of [20] have designed an ensemble stacking method to effectively detect cyberattacks in the IoT with high performance. The proposed stacked ensemble classifier outperformed individual base model classifiers when tested on three different datasets: Credit card fraud, NSL-KDD, and UNSW datasets. The experimental results presented in [20] demonstrated the performance of the proposed stacked ensemble models for cyberattack and fraud detection. The findings indicate that the ensemble stacking method outperformed individual base models, achieving high accuracy in detecting cyberattacks and credit card fraud.

Paper [21] presented a study on the application of ensemble learning techniques for detecting low footprint network intrusions. The authors have emphasized the importance of minimizing false positives in network intrusion detection and have compared the performance of bagging, boosting, and stacking techniques. They demonstrated that multi-layer stacking ensemble models outperform other ensemble techniques and non-ensemble models in detecting low footprint network intrusions, achieving an F1-score of 0.99 and a false positive rate of 0.001. The authors of [21] have also discussed the drawbacks of deep learning models in the context of intrusion detection, such as the need for large datasets, complex hyperparameters, and opaque results.

Paper [22] discussed an ensemble-based approach for efficient intrusion detection in network traffic, emphasizing the use of machine learning techniques for improved security measures. The study introduces an IDS based on stacking ensemble learning, which integrates multiple machine learning algorithms to enhance classification efficiency. The proposed system has been tested for UNSW-NB15, and has achieved high accuracy (96.16% in training and 97.95% in testing) and improved the performance compared to other IDS, as evidenced by precision, recall, and other measurement criteria.

The research presented in [23] has discussed the development of an IDS using ensemble-based machine learning techniques, with a focus on the RF ensemble method. The research has addressed the limitations of traditional IDSs and has presented a novel approach using various ensemble strategies, including RF, Adaboost, Gradient Boosting, and Gradient XGBoost. The system described in [23] was tested on multiple public datasets and consistently demonstrated accuracy exceeding 99%.

Recently, Dynamic multi-scale topological representation (DMTR) approach [24] has been published. The main features of DMTR concern its ability to handle dataset imbalance, while enhancing intrusion detection by building a dynamic topological representation of network traffic. Hence, the robustness of DMTR technique in handling class-imbalance and extremely dynamic network traffic have been demonstrated through experiments performed on four publicly accessible network traffic datasets.

Moreover, paper [25] has reviewed existing techniques that represent network traffic as graphs, such that graph neural networks (GNNs) can successfully capture the relationships encapsulated within it for intrusion detection. The survey [25] has identified the state-of-art GNN-based methods and has outlined interesting research directions for investigating GNN for solving intrusion detection problems.

As shown in Table 1, the research works that focused on ensemble learning for classifying attacks in NSL/KDD have explored machine learning techniques that provided limited accuracy.

Table 1: Comparison between ensemble learning approaches

Reference	Type of classification	Classifiers	Datasets	Accuracy	Recall	F1-score	Time
[18]	Multi-classification	Gradient descent and RF	NSL-KDD	91.96%	DoS: 81.85% Probe: 96.11% R2L: 97.75% U2R: 98.47%	–	
[19]	Multi-classification	Decision tree, k-nearest neighbors, deep neural network and RF	NSL-KDD	90.41%	78.82%	86.83%	1652 s
[20]	Multi-classification	Decision tree, XGBoost, and RF classifiers	NSL-KDD, and UNSW datasets	UNSW: 95.15% NSL-KDD: 81.28%	–	Unsw: 96.13% NSL-KDD: 77.26%	UNSW: 690.82 s NSL-KDD: 1669.04 s
[21]	Multi-classification	Stacking: Stochastic gradient descent (SGD), decision tree (Tree), naive bayes (NB), and logistic regression (LR)	Mawilab dataset CIC-IDS2017 dataset	99%	–	99%	–
[22]	Multi-classification	RF, decision tree, and k-nearest-neighbors	UNSW-NB15	96.16% in the training phase and 97.95%	96.62	97.20	–
[23]	Multi-classification	RF, gradient boosting, adaboost, gradient XGBoost	SIMARGL21 dataset	99%	99%	99%	–

3 Proposed Methodology

Our main framework is illustrated in Fig. 1. Thus, a pre-processing stage is performed before training separate models. The predictions from the separate deep learning models are then integrated and weighted to provide the final classification decision.

3.1 Standardization

Standardization has been utilized to change numeric features, such that the mean of each feature will have a mean of 0 and a standard deviation of 1 [26,27]. This brings all features to a similar scale, making it easier for machine learning and deep algorithms to learn and converge faster. Standardization requires the followings:

1. For each numeric feature, calculate the mean (μ) and standard deviation (σ).

2. Use Eq. (1) for each data point in the feature, to subtract the mean (μ) and divide by the standard deviation (σ).

$$z = (x - \mu)/\sigma \tag{1}$$

where:

- z is the standardized value.
- x is the original value.
- μ is the mean of the feature.
- σ is the standard deviation of the feature.

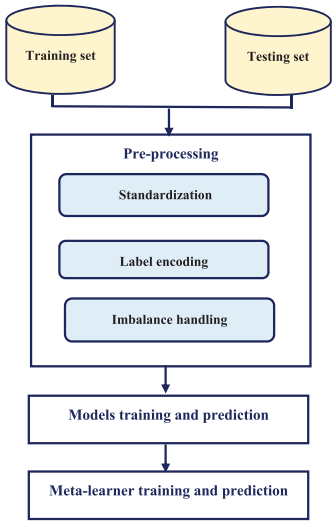

Figure 1: Proposed design for stacking deep learning models

3.2 Label Encoding

In the context of data science, label encoding refers to the process of converting categorical labels into numerical representations. Label encoding is necessary because most machine learning algorithms require numerical inputs. We mention that one hot [28] is a pre-processing technique that outputs binary vectors for categorical variables. We have adopted one hot encoding for representing categorical features of NSL-KDD dataset as this technique preserves the categorical information contained in the original variable. Instead of assigning arbitrary numerical values to categories, it models each categorical feature as a binary vector, which facilitates the task for machine and deep learning algorithms to understand and interpret the data.

It also considers each category as independent, avoiding the assumption of ordinality that may not exist in the data. In addition, one hot encoding reduces bias. Assigning arbitrary numerical values

to categories may introduce unintended relationships or orderings. By using binary indicators, one hot encoding avoids these biases and ensures that each category is treated equally. Another motivation for applying one hot encoding is that it supplies interpretable representation. The resulting binary vectors explicitly indicate the presence or absence of each category, allowing humans to easily understand and interpret the encoded features.

3.3 Imbalance Handling

Imbalance handling in the NSL-KDD dataset refers to addressing the issue of class imbalance within the dataset when building a deep learning model for network intrusion detection. The NSL-KDD dataset contains network connection records labeled as either normal or belonging to a specific type of attack. Class imbalance is present in NSL-KDD as the number of instances in one class (e.g., normal) is significantly larger than the number of instances in another class [27]. To handle class imbalance in the NSL-KDD dataset, several techniques can be employed. We have adopted SMOTE. This involves increasing the number of instances in the minority class by synthesizing new instances. We indicate that different imbalance handling techniques are available and may impact the performance of the network intrusion detection model. Further, the choice of technique may depend on the specific requirements of the application and the characteristics of the dataset. More importantly, SMOTE is particularly useful when you have imbalanced datasets in which the minority class is underrepresented and you want to balance the class distribution. It can be applied form multi-class classification. In this case, SMOTE generates synthetic examples for each minority class, making the dataset more balanced and improving the performance of the classifier. Thus, table shows the distribution of NSL-KDD samples before and after applying SMOTE.

3.4 Stacked Model Architecture

The proposed architecture of multi-classification of attacks. It includes two parallel branches to process data. The two branches have been constructed through experiments. The first branch relies on CNN model, while the second one relies on deep learning model. Another characteristic of the proposed developed stacked architecture is that the meta-learner focuses on a DNN to generate the final prediction.

The designed CNN typically consists of multiple layers, each one of them serves a specific purpose in the feature extraction and classification process. Hence, the role of the input layer is to receive the raw input data as one-dimensional array, and passes it for processing by the subsequent layers. Next, the convolution layer that consists of 32 filters performs a convolution operation. Then, the activation layer applies ReLu activation function element-wise to the output of the convolutional layer. The aim of the activation layer is to introduce non-linearity into the network, enabling it to learn complex patterns in the data. Furthermore, the pooling layers down-samples the spatial dimensions (width and height) of the feature maps while preserving the most salient information. We have adopted Max pooling where the maximum value within a local neighborhood is selected as the representative value. The dropout rate is 0.4 and it has been fixed by experiment. After that, a regularization is performed to prevent overfitting. This helps in making the network independent from specific activations, and therefore supports network's training. Subsequently, the proposed CNN executes convolution using 64 filters followed by ReLU activation function, max pooling and dropout. Then, the proposed CNN relies on a flatten layer that takes the output from the previous layers and flatten them into a vector. Finally, the proposed CNN learns high-level representation of the features through two fully connected layers, and produces the final multi-classification decision using a fully connected layer that contains 5 neurons and depends upon Softmax function as illustrated in Fig. 2.

On another hand, the architecture of the proposed DNN is simple as it incorporates an input layer and two fully connected layers. The number of neurons of fully connected layers is 25 and 10, respectively. At the end, the proposed DNN establishes multi-classification via an output layer including 5 neurons and relying on Softmax function.

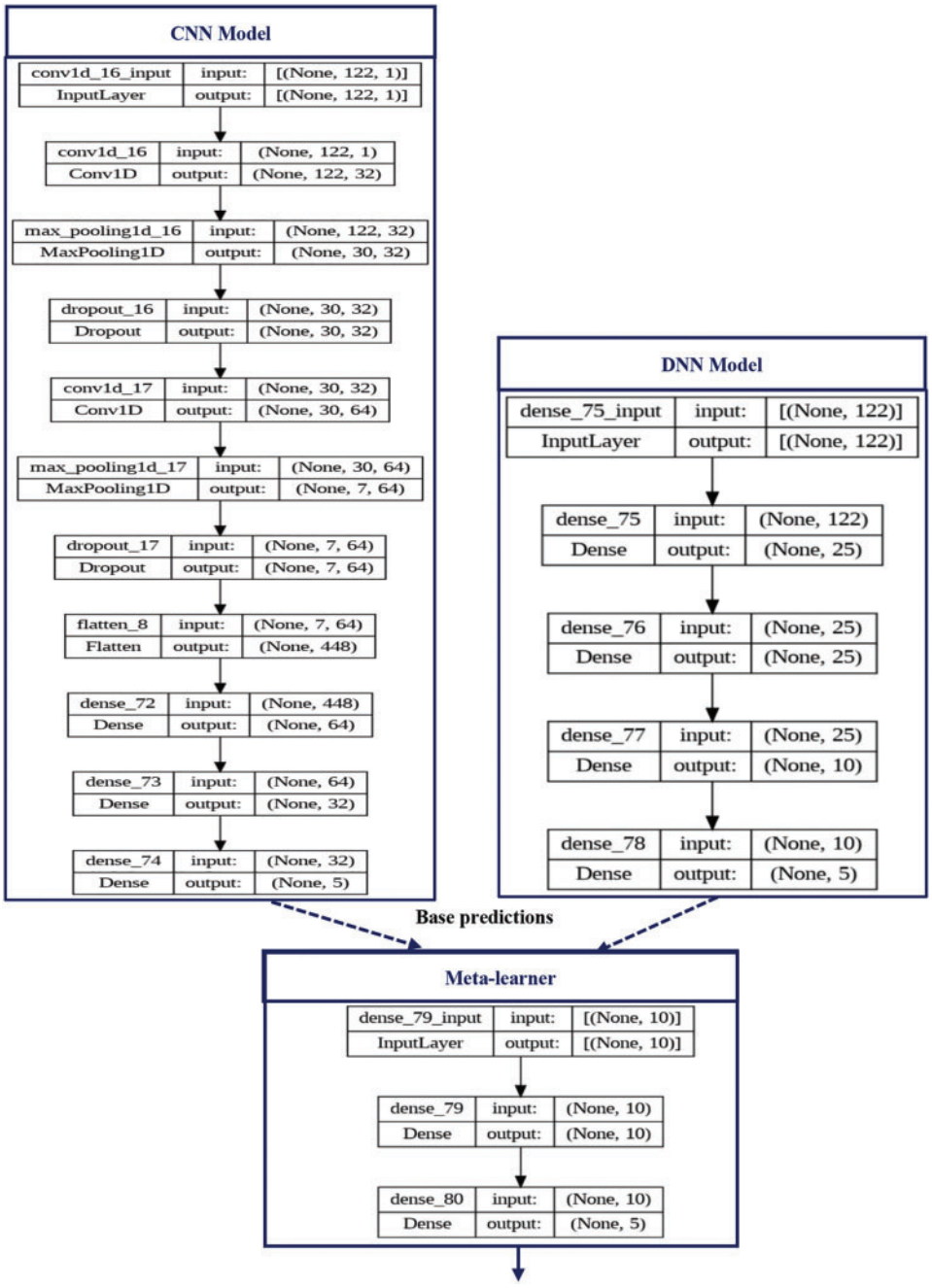

Figure 2: Detailed architecture for the proposed stacked model

The most important feature of the proposed model consists to combine the predictions of the designed CNN and DNN, which is a form of ensemble learning, where the goal is to improve the overall performance of the model by leveraging the strengths of individual models. Actually, we have adopted stacking for ensemble learning by training individually the proposed CNN and DNN on the same dataset, and then use the predictions of those models as input features for a meta-model that is trained to make the final prediction based on the predictions of the individual models. The architecture of the meta-learner includes an input layer, a ReLU activation layer and an output layer that outputs the final prediction based on Softmax function.

4 Results and Discussion

We have assessed the performance of the proposed CNN, DNN, and the proposed stacked model using Dell Inspiron 3880 with a 12-GB RAM configuration and having an Intel Core i7 CPU. Our implementation is based on Python that is a high-level programming language which is easy to learn and comprehend. We have also investigated Keras library which is a Python deep learning package that is available for free. Keras is user friendly, modular, and provides extendable design. Consequently, deep neural network implementation is becoming more and more popular using Keras library that supports creating and training different kinds of neural networks.

4.1 Confusion Matrix

The confusion matrix for multi-classification is a generalization for the confusion matrix for binary classification. It is an important metric as it summarizes the model's predictions across all classes. It is considered as a square matrix, where the number of rows and columns equal the number of classes. More specifically, the rows represent the true classes, while the columns represent the predicted classes. Each element of the confusion matrix indicates the number of instances that are assigned to a particular true/predicted class [28]. Using the confusion matrix, various evaluation metrics can be derived to assess the model's performance in multi-classification. These metrics are based on true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) as shown in Eqs. (2)–(5).

$$Accuracy\,(Acc) = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \tag{2}$$

$$Precision\,(P) = \frac{TP}{TP + FP} \times 100\% \tag{3}$$

$$Recall\,(R) = \frac{TP}{TP + FN} \times 100\% \tag{4}$$

$$F1 - score\,(F1) = 2 \times \frac{P \times R}{P + R} \tag{5}$$

4.2 Classification Report

The classification report provides textual analysis for each class. It illustrates the precision, recall, F1-score, and support for each class as well as the model's accuracy [28]. It is a useful evaluation tool for multi-classification as it shows the model's performance for each class, highlighting any variation in performance across multiple classes. In conjunction to the obtained accuracy, the classification report indicates the macro avg F1-score, which is an important metric that may be used as the dataset is balanced. The aim is to achieve a high value for the macro avg F1-score as it represents the average of

the precision and recall across all classes. In the context of intrusion detection, the macro avg F1-score illustrates the ability of the model to recognize all types of attacks.

4.3 Discussion

The confusion matrix of the proposed CNN, DNN, and stacked model are respectively shown in Figs. 3–5. The proposed stacked deep learning model's confusion matrix demonstrates its great accuracy since the majority of the integers are diagonal. This indicates that most of the attacks were accurately predicted by our designed stacked model. However, the proposed CNN and DNN models could only identify certain types of attacks.

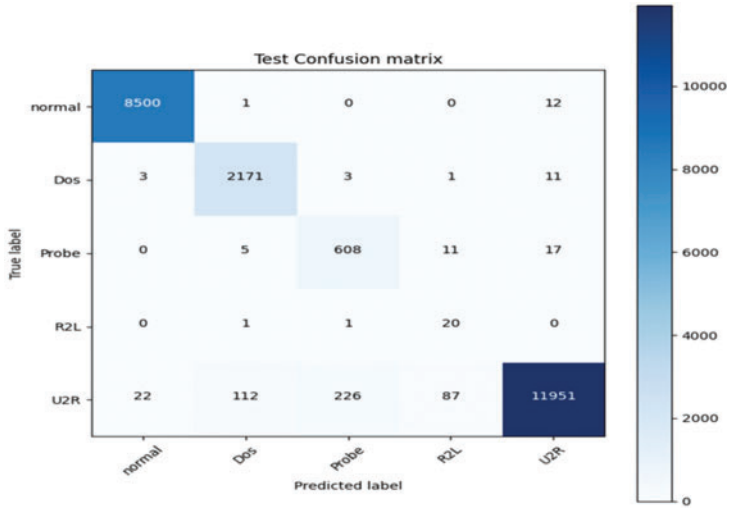

Figure 3: Confusion matrix for the proposed CNN

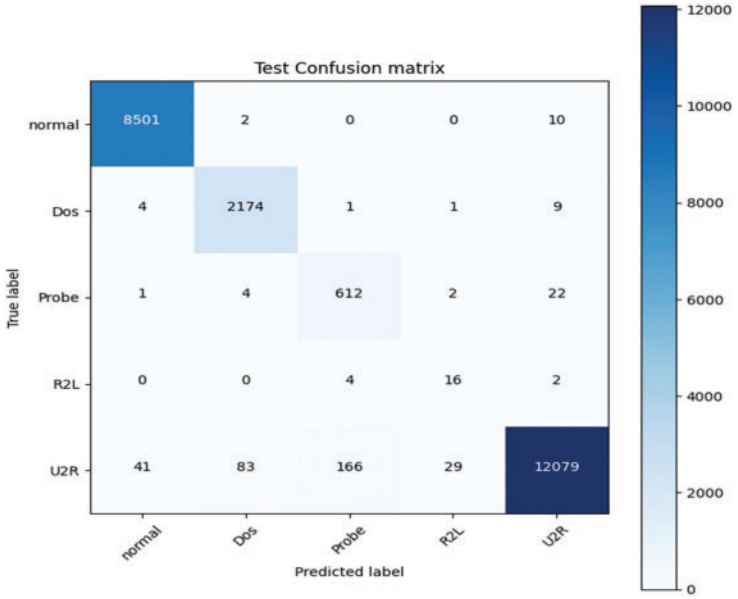

Figure 4: Confusion matrix for the proposed DNN

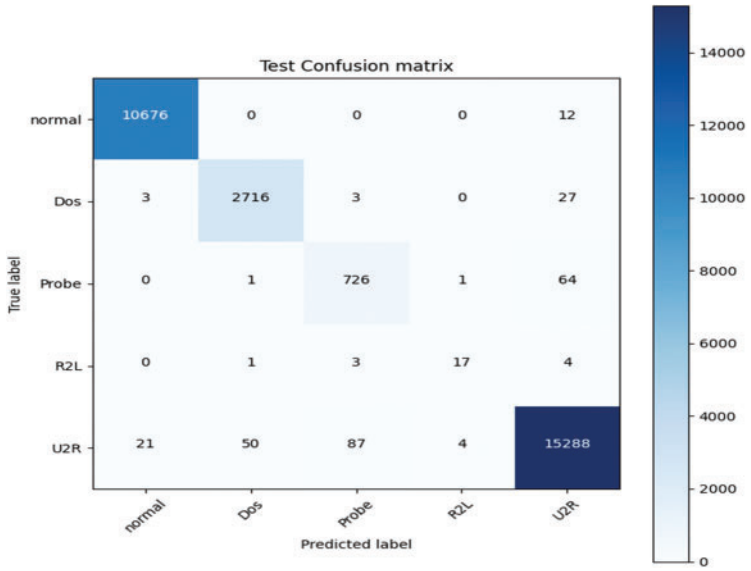

Figure 5: Confusion matrix for the proposed stacked model

The plots of the accuracy *vs.* epoch show the performance of the models on the training dataset over different iterations of the training process. Thus, Figs. 6–8 represent the plots of the accuracy for the proposed CNN, DNN, and stacked model, respectively. In each plot, the x-axis represents the epochs or iterations, which are the number of times the model has been trained on the entire training dataset. The y-axis represents the accuracy of the model, which is the proportion of correctly predicted instances on the training data.

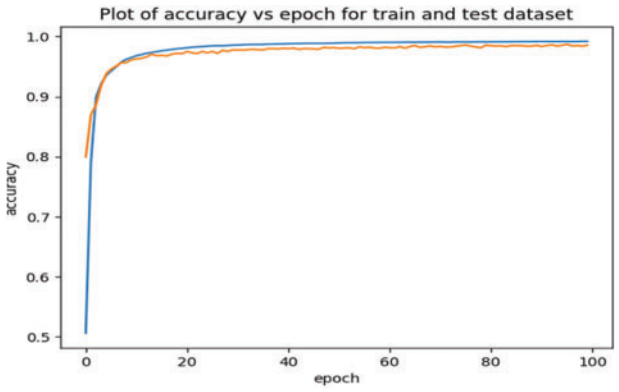

Figure 6: Accuracy plots for the proposed CNN

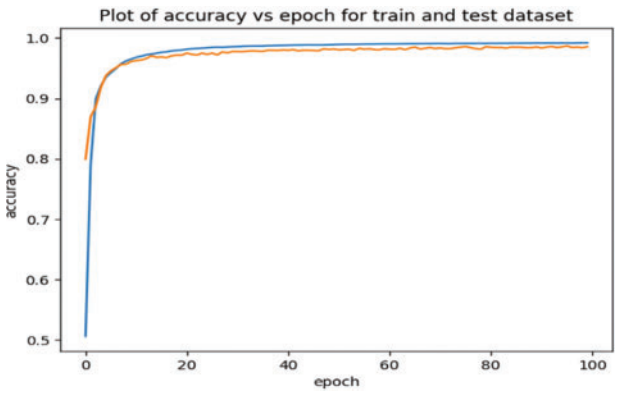

Figure 7: Accuracy plots for the proposed DNN

Figs. 6–8 prove that the accuracy consistently increases with each epoch for the three proposed models that have the ability to learn and improve their performance over time. This confirms that the three proposed models have converged towards better performance and have captured effectively the patterns in the training dataset. The graphs for both the training and validation sets illustrate the models' capability for intrusion detection. As the training progressed, the attained accuracies consistently and steadily improved. The high level of validation accuracy indicates the models' strength and proficiency for classifying attacks. Importantly, the curves remained stable after a

certain number of epochs, with no significant changes observed. It is noteworthy, the proposed stacked deep learning model converges quickly compared to the other two proposed models. Additionally, the training and prediction performance of the three proposed models have been evaluated as shown in Figs. 9–11 that illustrate the variation of the loss *vs.* epoch for the training and testing datasets.

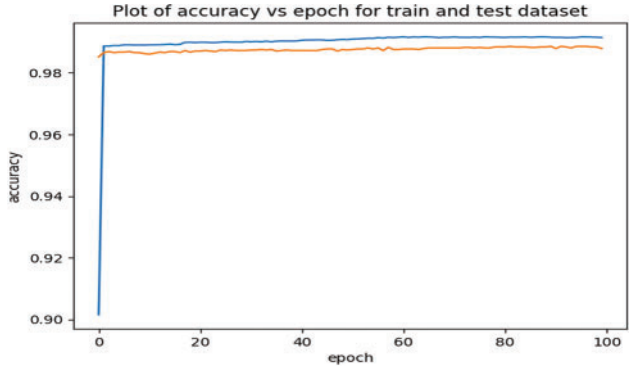

Figure 8: Accuracy plots for the proposed stacked model

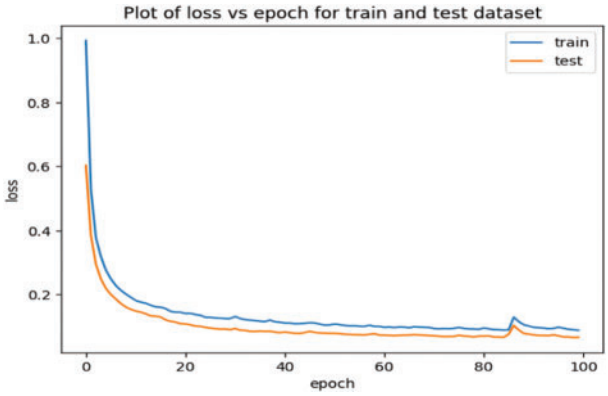

Figure 9: Loss plots for the proposed CNN

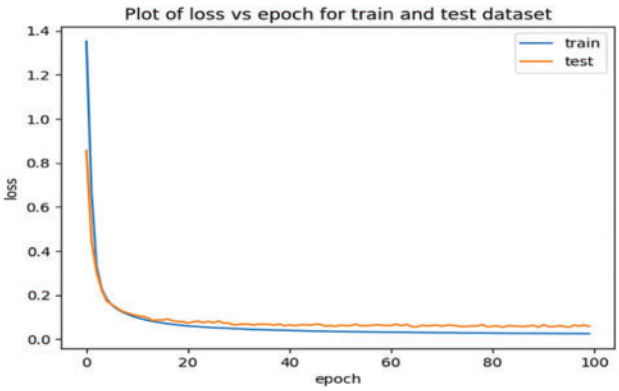

Figure 10: Loss plots for the proposed DNN

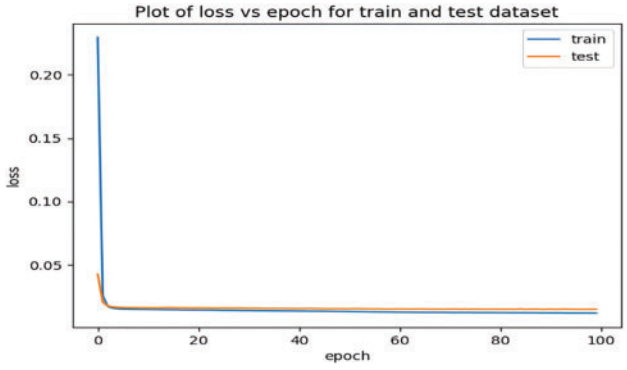

Figure 11: Loss plots for the proposed stacked model

It is clear from Fig. 11 that the proposed stacked deep learning model exhibits the lowest loss compared to the other two models. Moreover, the classification reports of the proposed CNN and DNN are shown in Figs. 12 and 13, respectively. With a macro average F1-score of 93%, the proposed stacked deep learning demonstrates its performance according to the classification report shown in Fig. 14. The average of each class's F1-score is used to compute the macro average. As a result, putting the model's performance into a single figure is a useful assessment. Examining the categorization reports reveals that the macro average F1-score for the other models is substantially lower than the average F1-score of the proposed stacked deep learning model, indicating its superiority for predicting various attacks.

	precision	recall	f1-score	support
normal	1.00	1.00	1.00	8513
Dos	0.95	0.99	0.97	2189
Probe	0.73	0.95	0.82	641
R2L	0.17	0.91	0.28	22
U2R	1.00	0.96	0.98	12398
accuracy			0.98	23763
macro avg	0.77	0.96	0.81	23763
weighted avg	0.98	0.98	0.98	23763

Figure 12: Classification report for the proposed CNN

	precision	recall	f1-score	support
normal	0.99	1.00	1.00	8513
Dos	0.96	0.99	0.98	2189
Probe	0.78	0.95	0.86	641
R2L	0.33	0.73	0.46	22
U2R	1.00	0.97	0.99	12398
accuracy			0.98	23763
macro avg	0.81	0.93	0.86	23763
weighted avg	0.99	0.98	0.98	23763

Figure 13: Classification report for the proposed DNN

	precision	recall	f1-score	support
normal	1.00	1.00	1.00	8513
Dos	0.98	0.99	0.99	2189
Probe	0.93	0.88	0.90	641
R2L	0.74	0.77	0.76	22
U2R	0.99	0.99	0.99	12398
accuracy			0.99	23763
macro avg	0.93	0.93	0.93	23763
weighted avg	0.99	0.99	0.99	23763

Figure 14: Classification report for the proposed stacked model

The proposed deep learning stacked model outperforms the other models, according to the findings of a variety of criteria used to evaluate the various algorithms. Thus, our results show that stacking deep learning neural networks is an effective way to address the imbalance issue and subsequently predict attacks when combined with SMOTE technique. To evaluate the performance of our staking model, its results have been compared to recently published papers that applied stacking for intrusion detection. We mention that some papers have focused on specific evaluation metrics and did not provide details regarding multi-classification results. Table 2 shows that that our proposed technique achieves high accuracy similar to the results obtained in [21,23]. In particular, the training time of the proposed stacking model is lower than the training time of benchmark techniques. This result demonstrates the efficiency and robustness of our proposed model.

Table 2: Performance comparison with state-of-art techniques

Reference	Accuracy	Time
[18]	91.96%	–
[19]	90.41%	1652 s

(Continued)

Table 2 (continued)

Reference	Accuracy	Time
[20]	UNSW: 95.15%, NSL-KDD:81.28%	UNSW: 690.82 s, NSL-KDD: 1669.04 s
[21]	99%	–
[22]	96.16% in the training phase and 97.95%	–
[23]	99%	–
Our proposed technique	99%	158 s

5 Conclusion

In conclusion, the utilization of stacking deep learning models for multi-classification in the NSL-KDD dataset offers several significant advantages. Combining SMOTE and stacking leverages the collective wisdom of CNN and DNN, enabling improved predictive performance compared to individual models. By combining the strengths of the designed deep learning architectures, stacking can effectively increase the detection rate for multiple types of attacks.

Overall, the use of stacking deep learning models for multi-classification in the NSL-KDD dataset represents a powerful approach that can significantly enhance the accuracy, and the robustness of the classification process. By harnessing the collective intelligence of diverse models, stacking offers a promising avenue for advancing the field of network intrusion detection and related domains. As a future work, we will examine the role of transfer learning for intrusion detection.

Acknowledgement: The author extends her appreciation to Princess Nourah bint Abdulrahman University for supporting scientific research.

Funding Statement: This research received no specific grant.

Author Contributions: The author confirms contribution to the paper as follows: Study conception and design: S.A. Chelloug; data collection: S.A. Chelloug; analysis and interpretation of results: S.A. Chelloug; draft manuscript preparation: S.A. Chelloug. The author reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available at https://www.kaggle.com/datasets/hassan06/nslkdd.

Conflicts of Interest: The author declares that she has no conflicts of interest to report regarding the present study.

References

[1] A. A. Ghorbani, W. Lu, and M. Tavallaee, "Detection approaches," in *Network Intrusion Detection and Prevention*, New York, USA: Springer, 2010, vol. 2, pp. 27–53.

[2] K. Kim, M. E. Aminanto, and H. C.Tanuwidjaja, *Network Intrusion Detection Using Deep Learning: A Feature Learning Approach*, New York, USA: Springer, 2018, vol. 5, pp. 35–44.

[3] S. A. N. Alexandropoulos, C. K. Aridas, S. B. Kotsiantis, and M. N. Vrahatis, "Stacking strong ensembles of classifiers," in *Artificial Intelligence Applications and Innovations*, New York, USA: Springer, 2019, vol. 559, pp. 545–556.
[4] M. H. Zaib, NSL-KDD. Accessed: Feb. 4, 2024. [Online]. Available: https://www.kaggle.com/datasets/hassan06/nslkdd
[5] A. Saad Hussein, T. Li, C. W. Yohannese, and K. Bashir, "A-SMOTE: A new preprocessing approach for highly imbalanced datasets by improving SMOTE," *Int. J. Comput. Intell. Syst.*, vol. 12, no. 2, pp. 1412–1422, 2019. doi: 10.2991/ijcis.d.191114.002.
[6] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 2–22, 2019. doi: 10.1186/s42400-019-0038-7.
[7] S. M. Othman, F. M. BaAlwi, N. T. Alsohybe, and A. Y. Al-Hashida, "Intrusion detection model using machine learning algorithm on big data environment," *J. Big Data*, vol. 5, no. 43, pp. 12, 2018. doi: 10.1186/s40537-018-0145-4.
[8] A. R. Khan, M. Kashif, R. H. Jhaveri, R. Raut, T. Saba and S. A. Bahaj, "Deep learning for intrusion detection and security of internet of things (IoT): Current analysis, challenges, and possible solutions," *Secur. Commun. Netw.*, vol. 2022, pp. 4016073, 2022. doi: 10.1155/2022/4016073.
[9] T. Wisanwanichthan and M. Thammawichai, "A double-layered hybrid approach for network intrusion detection system using combined naive bayes and SVM," *IEEE Access*, vol. 9, pp. 138432–138450, 2021. doi: 10.1109/ACCESS.2021.3118573.
[10] M. Al Lail, A. Garcia, and S. Olivo, "Machine learning for network intrusion detection—A comparative study," *Future Internet*, vol. 15, no. 7, pp. 243, 2023. doi: 10.3390/fi15070243.
[11] Y. Kayode Saheed, A. Idris Abiodun, S. Misra, M. Kristiansen Holone, and R. Colomo-Palacios, "A machine learning-based intrusion detection for detecting internet of things network attacks," *Alex. Eng. J.*, vol. 61, no. 12, pp. 9395–9409, 2022. doi: 10.1016/j.aej.2022.02.063.
[12] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert, and F. de Turck, "Towards model generalization for intrusion detection: Unsupervised machine learning techniques," *J. Netw. Syst. Manage.*, vol. 30, no. 1, pp. 25, 2022. doi: 10.1007/s10922-021-09615-7.
[13] A. Dahou *et al.*, "Intrusion detection system for IoT based on deep learning and modified reptile search algorithm," *Comput. Intell. Neurosci.*, vol. 2022, no. 4, pp. 1–15, 2022. doi: 10.1155/2022/6473507.
[14] A. Fatani, M. A. Elaziz, A. Dahou, M. A. A. Al-Qaness, and S. Lu, "IoT intrusion detection system using deep learning and enhanced transient search optimization," *IEEE Access*, vol. 9, pp. 123448–123464, 2021. doi: 10.1109/ACCESS.2021.3109081.
[15] A. Sagu, N. S. Gill, P. Gulia, P. K. Singh, and W. C. Hong, "Design of metaheuristic optimization algorithms for deep learning model for secure IoT environment," *Sustainability*, vol. 15, no. 3, pp. 21, 2023. doi: 10.3390/su15032204.
[16] H. M. Chuang and L. J. Ye, "Applying transfer learning approaches for intrusion detection in software-defined networking," *Sustainability*, vol. 15, no. 12, pp. 24, 2023. doi: 10.3390/su15129395.
[17] S. T. Mehedi, A. Anwar, Z. Rahman, and K. Ahmed, "Deep transfer learning based intrusion detection system for electric vehicular networks," *Sensors*, vol. 21, no. 14, pp. 23, 2021. doi: 10.3390/s21144736.
[18] H. Rajadurai and U. D. Gandhi, "A stacked ensemble learning model for intrusion detection in wireless network," *Neural Comput. Appl.*, vol. 34, no. 18, pp. 15387–15395, 2022. doi: 10.1007/s00521-020-04986-5.
[19] Y. Tang, L. Gu, and L. Wang, "Deep stacking network for intrusion detection," *Sensors*, vol. 22, no. 1, pp. 17–202, 2022. doi: 10.3390/s22010025.
[20] R. Soleymanzadeh, M. Aljasim, M.W. Qadeer, and R. Kashef, "Cyberattack and fraud detection using ensemble stacking," *AI*, vol. 3, no. 1, pp. 22–36, 2022. doi: 10.3390/ai3010002.
[21] S. Shafieian and M. Zulkernine, "Multi-layer stacking ensemble learners for low footprint network intrusion detection," *Complex Intell. Syst.*, vol. 9, no. 4, pp. 3787–3799, 2023. doi: 10.1007/s40747-022-00809-3.
[22] A. Almomani *et al.*, "Ensemble-based approach for efficient intrusion detection in network traffic," *Intell. Autom. Soft Comput.*, vol. 37, no. 2, pp. 2499–2517, 2023. doi: 10.32604/iasc.2023.039687.

[23] M. A. Hossain and M. S. Islam, "Ensuring network security with a robust intrusion detection system using ensemble-based machine learning," *Array*, vol. 19, no. 2, pp. 14, 2023. doi: 10.1016/j.array.2023.100306.
[24] M. Zhong, M. Lin, and Z. He, "Dynamic multi-scale topological representation for enhancing network intrusion detection," *Comput. Secur.*, vol. 135, pp. 103516, 2023. doi: 10.1016/j.cose.2023.103516.
[25] M. Zhong, M. Lin, C. Zhang, and Z. Xu, "A survey on graph neural networks for intrusion detection systems: Methods, trends and challenges," *Comput. Secur.*, vol. 141, no. 3, pp. 103821, 2024. doi: 10.1016/j.cose.2024.103821.
[26] A. Abdelkhalek and M. Mashaly, "Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning," *J. Supercomput.*, vol. 79, no. 10, pp. 10611–10644, 2023. doi: 10.1007/s11227-023-05073-x.
[27] Z. Tauscher, Y. Jiang, K. Zhang, J. Wang, and H. Song, "Learning to detect: A data-driven approach for network intrusion detection," in *Proc. 2021 IEEE Inter. Perfor., Comput., Commun. Conf. (IPCCC)*, Austin, USA, 2021, pp. 1–6.
[28] D. Bowes Wagner, T. Hall, and D. Gray, "Comparing the performance of fault prediction models which report multiple performance measures: Recomputing the confusion matrix," in *PROMISE'12: Proc. 8th Inter. Conf. Predictive Models Softw. Eng.*, San Francisco, USA, 2012, pp. 109–118. doi: 10.1145/2365324.2365338.