Iraqi Journal of Communications and Computer Networks
Vol. (2), Iss. (1), March 2026
e-ISSN: 3105-9252, p-ISSN: 3105-9244
DOI: https://doi.org/10.58564/IJCCN.2.1.2026.14

https://ijccn.net/ **Research Article**

Robust Network Intrusion Detection Using Deep Stacking Ensembles and Hybrid Feature Selection

Shahad Aman Ibrahim

MSc. Researcher, Ministry of Higher Education and Scientific Research, Baghdad, Iraq

Corresponding Author: shahadaman2017@gmail.com

Citation: Shahad A. Ibrahim. (2026). Robust Network Intrusion Detection Using Deep Stacking Ensembles and Hybrid Feature Selection. Iraqi Journal of Communications and Computer Networks (IJCCN), 2(1), 19-26.

ARTICLE INFO

Received: Oct. 28, 2025
Revised: Jan. 30, 2025
Accepted:Mar. 14, 2026

ABSTRACT

In the contemporary networked systems, network security has emerged to be a critical issue. Intrusion Detection Systems (IDS) are imperative defense systems that can be used to detect malicious network traffic. The study provides a new IDS model, which combines various machine learning models and deep learning architectures to generate better detection accuracy on the NSL-KDD benchmark data. The researchers make four main contributions: (1) a hybrid approach to feature selection which involves Information Gain and correlation-based redundancy elimination, (2) Synthetic Minority Oversampling Technique (SMOTE) to overcome the class imbalance, (3) a stacking ensemble classifier, and (4) a Deep Neural Network (DNN) with Batch Normalization, LeakyReLU activation, and Dropout regularization. Through experiment, it has been shown that the proposed stacking ensemble has an accuracy of 99.80 and an F1-score of 0.998 on binary classification, and 99.53 and five attack categories on multi-class classification. The proposed strategy is much better than individual baseline classifiers and has strong detection ability of minority attack classes such as R2L and U2R attacks.

Keywords: Intrusion Detection System, NSL-KDD, Feature Selection, SMOTE, Stacking Ensemble, Neural Network, Network Security, Cybersecurity.

INTRODUCTION

The ever-growing number of internet-connected devices and digital communications systems has compounded the cybersecurity environment to become more complex. Cyber-attacks and network intrusions are very dangerous to organizations, governments and individual persons across the globe. Recent cybersecurity news reports indicate that the number and complexity of network attacks are still rising leading to significant financial damage and loss of sensitive data integrity.

Intrusion Detection Systems (IDS) have become the payments of network security infrastructure. An IDS is a tool that monitors the network activities and system traffic to detect possible security or policy breaches, malicious activity. The conventional IDS methods can be divided into two distinct groups; signature based detection, which is based on the known attack patterns and anomaly based detection, which detects the abnormality of a normal behavior profile.

Machine learning (ML) algorithms have proven to have incredible potential in improving the capabilities of IDS by learning intricate trends to positive network traffic data automatically. These methods have the ability to handle the changing attack strategies, and they are able to pick up attack variants that have never been seen before due to the constraints of the traditional signature-based methods. The latest developments of deep learning have added to more possibilities of efficient and more precise intrusion detection.

Although there has been tremendous improvement, there are still a number of issues that are involved in coming up with effective IDS solutions. The distribution of network traffic data in classes with normal connections far exceeding any given instance of attackers can bias classifiers to majority classes and decrease the detection rates of rare yet important attacks like Remote-to-Local (R2L) and User-to-Root (U2R) attacks. The network traffic dimensionalities are also high and may cause noise and computational load, which may worsen classification performance.

Research Objectives

This research aims to address the aforementioned challenges through the following objectives:

1. Create a hybrid Information Gain/correlation-based filtering approach to information gain that will be used to select the most discriminative and non-redundant features.
2. Apply SMOTE to address the class imbalance problem inherent in network intrusion datasets, particularly improving detection of minority attack classes.
3. Design a stacking ensemble classifier that leverages the complementary strengths of multiple base learners to achieve robust and accurate intrusion detection.
4. Implement a Deep Neural Network with modern regularization techniques for multi-class attack classification.
5. Conduct comprehensive comparative analysis between the proposed methods and existing approaches.

Paper Organization

The rest of this paper will be structured in the following way. Section 2 presents a literature review on related work on intrusion detection based on machine learning. Section 3 offers a description of the NSL-KDD dataset and its properties. Section 4 presents the suggested methodology comprised of feature selection, data balancing, ensemble learning and deep learning strategies. In section 5, one can find the experimental setup and results. The findings and implications are discussed in Section 6. Lastly, Section 7 brings the paper to a conclusion and proposes future research directions.

RELATED WORK

The field of intrusion detection has been actively developed since the pioneer publication of Denning in 1986 as the model of intrusion detection was introduced. Since that time, many detection methods have been invented to enhance the detection accuracy and efficiency. The paper examines the most topical research on the subject.

Traditional Machine Learning Approaches

Various scholars have used classical machine learning algorithms in detection of network intrusion. The use of decision trees has been popular because it is easy to interpret and efficient. In a number of studies on IDS, Random Forest, being an ensemble of decision trees has shown better performance compared to other rivals such as Maximum Entertainment by minimizing overfitting by means of bagging along with random selection of features. It has been discovered that on binary classification problems, Random Forest accuracy rates are above 99 percent on the NSL-KDD data[1].

SVM has been also used in IDS especially in the context of binary classification. Although SVMs provide great theoretical basis and excellent generalization, they require considerable complexity in their computation of large-scale data, which is often prohibitive. Another non-parametric method studied is K-Nearest Neighbors (KNN) which is affected by feature scale and high dimensionality, restricting its applicability[2].

Gradient Boosting, such as XGBoost and CatBoost, and LightGBM, are the methods that had been a major point of attention over the last few years. These algorithms construct sequential decision trees, which are aimed at fixing past errors and achieve state-of-the-art results on a range of benchmarks. XGBoost specifically has proved to be highly accurate and computationally efficient in intrusion detection with certain experiments reporting a high accuracy of over 99.9 percent on the training sample[3].

Deep Learning Approaches

There has been an increase in the use of deep learning techniques on intrusion detection problems. Deep Neural Networks (DNNs) that have multiple hidden layers are able to learn hierarchical feature representations on raw network traffic information automatically. CNNs that have been modified to be used in IDS are considering network traffic features to be spatial patterns. RNNs and Long Short-Term Memory (LSTM) networks have been used to learn temporal relationships between network traffic data in a sequence[4].

Autoencoders have been applied to intrusion detection using anomalies with the model trained to replicate typical traffic patterns and detect anomalies based on reconstruction error. GANs explored have also been used to augment data and generate synthetic attacks to enhance training of classifiers[5].

Feature Selection Methods

The selection of features is a significant issue in the performance of IDS. Only Information Gain, Chi-squared, and Correlation-based Feature Selection (CFS) are mostly used filter methods. Recursive Feature Elimination (RFE) and wrapper algorithms are more precise at more cost in terms of computation. Other studies have taken two or more technique to choose features and used them in combination so that the weaknesses are used to their advantage[6].

Class Imbalance Handling

One of the issues that are familiar with network intrusion detection is class imbalance. Normal traffic usually makes up the largest part of the network data, and there are some forms of attacks that are not common. SMOTE and its versions (Borderline-SMOTE, ADASYN) have been extensively used as the means of creating synthetic minorities classes samples. Other scholars have used SMOTE alongside the under-sampling methods or K-means clustering to create more balanced and representative training data sets[7].

Ensemble Methods

Ensemble learning techniques represent a combination of a set of classifiers in order to enhance both accuracy and strength of prediction. The three most appropriate ensemble strategies are bagging (Bootstrap Aggregating), Boosting, and Stacking. In specific, stacking has demonstrated good approach to IDS by enhancing the combination of different base learners via a meta-learner to learn the best combination of base prediction. The recent research has shown that it is possible to stack ensembles of Random Forest, XGBoost, and Extra Trees with Logistic Regression as a meta-learner and achieve classification accuracy more than 99.5%[8].

3. Dataset Description

NSL-KDD Dataset

The NSL-KDD data is a refined version of the original KDD Cup 1999 data which was created by the Canadian Institute of Cybersecurity at the University of New Brunswick[9]. It solves a number of intrinsic issues of the KDD'99 data such as the removal of duplicate entries that might give preference to popular patterns by the classifiers. The dataset has emerged as one of the most popular standards of assessing the performance of IDS[10].

The NSLKDD dataset of connections records has 41 features in every record of the connection and a label defining the type of connection[11]. The features may be classified into four namely: basic features of TCP connections (features 1-9), content features within a connection (features 10-22), time-based traffic features (features 23-31), and host-based traffic features (features 32-41). Three of the attributes (protocol type, service and flag) are categorical and the rest 38 are numeric[12] [13].

Dataset Statistics

Table 1: NSL-KDD Dataset Statistics

roperty	Training Set	Test Set
Total Records	125,973	22,544
Normal	67,343 (53.46%)	9,711 (43.08%)
DoS	45,927 (36.46%)	7,458 (33.08%)
Probe	11,656 (9.25%)	2,421 (10.74%)
R2L	995 (0.79%)	2,754 (12.22%)
U2R	52 (0.04%)	200 (0.89%)
Features	41	41
Attack Types	24	38

Attack Categories

The NSL-KDD dataset has four major categories of attacks. Denial of Service (DoS) attacks are used to render network resources inaccessible to unattackable users by using methods like SYN flooding and Smurf attacks[14]. Probe attacks include surveillance and port scanning to come up with information on possible vulnerabilities.

Remote-to-Local (R2L) attacks- are attacks by a remote machine that is unauthorized to access such as password guessing[15]. The User-to-Root (U2R) attacks are used to exploit the local vulnerabilities to acquire superuser privileges, like buffer overflow attacks[16] [17] [18].

PROPOSED METHODOLOGY

The four principal stages that make up the proposed IDS framework are as follows: preprocessing of data, including hybrid features selection, imbalance of classes management through the use of SMOTE, model training, which includes a stacking ensemble model and an individual one, as well as the evaluation of these models through the application of comprehensive performance measures. Each of the stages is explained in the following subsections.

Data Preprocessing

The preprocessing stage involves several steps to prepare the raw NSL-KDD data for model training. First, the difficulty level column is removed as it is not part of the original network traffic features. Categorical features (protocol_type, service, and flag) are encoded using Label Encoding to convert them to numerical representations. The attack labels are mapped to five categories (Normal, DoS, Probe, R2L, U2R) for multi-class classification, and to two categories (Normal, Attack) for binary classification.

Contribution 1: Hybrid Feature Selection

The proposed hybrid feature selection method operates in two phases to identify the most relevant and non-redundant features for intrusion detection.

Phase 1: Information Gain Selection

In the first phase, Mutual Information (Information Gain) is computed for each feature with respect to the target class. Mutual Information quantifies the amount of information that a feature provides about the class label, making it effective for identifying individually discriminative features. Features with Mutual Information scores above a threshold of 0.01 are retained, eliminating features with negligible information content.

The Mutual Information $I(X;Y)$ between feature X and class Y is defined as: $I(X;Y) = H(Y) - H(Y|X)$, where $H(Y)$ is the entropy of the class distribution and $H(Y|X)$ is the conditional entropy of Y given X. This measure features both linear and non-linear relationships between features and their target variable.

Phase 2: Correlation-based Redundancy Removal

During the second step, Pearson correlation coefficient is calculated on all pairs of features of those features that are selected in Phase 1. Where the absolute correlation between two features has a score above 0.95 then the feature whose Information Gain score is less is deleted. This would remove duplicate features that may give redundant information which lowers the complexity of the model and possible multicollinearity without loss of discriminative power.

Contribution 2: SMOTE for Class Imbalance

To counteract the extreme imbalance of classes of the NSL-KDD training set, the Synthetic Minority Oversampling Technique (SMOTE) is used. SMOTE produces artificial examples of minority classes through interpolation between existing instance of minority classes and their closest neighbors. Each of the minority samples then identifies k nearest neighbors to create new synthetic samples along the line segments between the sample and the neighbors.

Such a strategy is especially critical to the R2L and U2R types of attacks, which comprise less than 1% of the training data. Classifiers without oversampling are normally biased against the majority classes (Normal and DoS) and are therefore not good at detecting rarer but critical attacks. The implementation of SMOTE involves the use of 3 nearest neighbors to strike a balance between the cost and quality of the generated samples.

Contribution 3: Stacking Ensemble Classifier

The proposed stacking ensemble consists of four different base classifiers, and a meta-learner, to obtain a strong classification performance. Random Forest, XGBoost, Extra Trees and Gradient Boosting are the base estimators. These classifiers have been chosen due to their complementary learning plans: Random Forest and Extra Trees are based on bagging, random feature sampling, and XGBoost and Gradient Boosting are based on sequential boosting.

In the stacking framework, the base classifiers are trained using 5-fold cross-validation on the training data, and their out-of-fold predictions form a new feature matrix. A Logistic Regression meta-learner is then trained on this feature matrix to learn the optimal combination of base classifier predictions. This two-level architecture enables the ensemble to leverage the strengths of each base classifier while mitigating their individual weaknesses.

Contribution 4: Deep Neural Network

A Deep Neural Network is designed for multi-class intrusion detection with the following architecture. The network consists of four fully connected hidden layers with 256, 128, 64, and 32 neurons respectively. Each hidden layer is followed by Batch Normalization for stable training, LeakyReLU activation (alpha=0.1) to prevent the dying neuron problem, and Dropout for regularization. The Dropout rates decrease progressively from 0.3 in the first layers to 0.09 in the deeper layers, allowing the network to retain more information as features become more abstract.

The network is trained using the Adam optimizer with an initial learning rate of 0.001, categorical cross-entropy loss, and a batch size of 256. Two callback mechanisms are employed: Early Stopping with a patience of 10 epochs to prevent overfitting, and ReduceLROnPlateau to dynamically reduce the learning rate when validation loss plateaus.

EXPERIMENTAL RESULTS

Experimental Setup

All experiments were conducted using Python 3.10 with scikit-learn 1.3, XGBoost 2.0, imbalanced-learn 0.11, and TensorFlow 2.15. The experiments were performed on a system with an Intel Core i7 processor and 16GB RAM. All models were trained on the SMOTE-balanced training set and evaluated on the original (unmodified) test set to simulate realistic deployment conditions.

Feature Selection Results

The hybrid feature selection method reduced the original 41 features to 28 features (31.7% reductionPhase 1 (Information Gain) dropped 6 features that had insignificant mutual information scores. Phase 2 (Correlation-based filtering): It eliminated another 7 features whose inter-feature correlation was greater than 0.95. The Information gain rankings of the top features are service, flag, srcbytes, dstbytes, dsthostsrvcount, samesrvrate, and loggedin, which are also in line with the known domain information about network intrusion profiles.

Binary Classification Results

Table 2: Binary Classification Results (Normal vs Attack)

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	MCC	Time (s)
Decision Tree	98.45	98.50	98.45	98.47	0.969	1.2
Random Forest	99.62	99.63	99.62	99.62	0.992	3.8
Extra Trees	99.58	99.60	99.58	99.59	0.991	2.9
XGBoost	99.71	99.72	99.71	99.71	0.994	5.1
KNN	97.85	97.90	97.85	97.87	0.957	8.5
Naive Bayes	89.12	90.35	89.12	89.50	0.783	0.3
AdaBoost	98.90	98.92	98.90	98.91	0.978	12.4
Gradient Boosting	99.35	99.37	99.35	99.36	0.987	15.2
Stacking Ensemble	**99.80**	**99.81**	**99.80**	**99.80**	**0.996**	**45.3**

The results of the binary classification for all the evaluated models are presented in Table 2. The obtained Stacking Ensemble has the best performance in all the metrics with 99.80% accuracy, 99.80% F1-score, and 0.996 Matthews Correlation Coefficient (MCC). XGBoost has the second highest accuracy with 99.71%, Random forest comes a close second with 99.62% and Extra trees with 99.58%. The ensemble approach gives a consistent improvement of 0.09-0.18% over the best classifier.

Multi-class Classification Results

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Random Forest	99.35	99.38	99.35	99.36
XGBoost	99.53	99.55	99.53	99.54
Extra Trees	99.28	99.30	99.28	99.29
DNN	98.87	98.92	98.87	98.89

Table 3: Multi-class Classification Results (5 categories)

The results of multi-class classification for five categories of attacks are presented in Table 3. XGBoost has the highest performance as it yields 99.53% accuracy and F1-score 99.54%. The DNN is able to achieve competitive results with 98.87% accuracy showing the effectiveness of the proposed deep learning architecture. All models greatly benefit from the SMote balanced training data, as shown by their increased detection rates for minority classes compared to models trained with imbalanced data.

Per-class Detection Performance

Table 4: Per-class Detection Performance (XGBoost with SMOTE)

Attack Category	Precision (%)	Recall (%)	F1-Score (%)	Support
Normal	99.75	99.82	99.78	9,711
DoS	99.90	99.85	99.87	7,458
Probe	99.42	99.38	99.40	2,421
R2L	97.85	96.92	97.38	2,754
U2R	92.50	88.50	90.45	200

The table 4 shows the per-class performance of the XGBoost model that has achieved the best performance. The model attains great detection of the normal, DoS, and Probe traffic with the F1-scores above 99. It is interesting to note that R2L detection F1-score is 97.38% and U2R is 90.45% which is almost twice that of the models trained without SMOTE (usually around 60% in U2R). This success is a direct indication of the suitability of the suggested method of class balancing.

Impact of Feature Selection

Table 5: Impact of Feature Selection on Random Forest Performance

Configuration	Features	Accuracy (%)	Training Time (s)
All Features	41	99.45	7.2
MI Selection Only	35	99.58	5.8
Hybrid Selection	28	99.62	3.8

The effect of the hybrid feature selection method is shown in Table 5. The hybrid way is not only more accurate with a 99.45% reduction to 99.62% through the elimination of noisy and redundant features, but also shorter training time by 47.2 (7.2 seconds to 3.8 seconds). This validates the fact that in Phase 2, elimination of correlated features is a further added advantage to the Information Gain filtering of Phase 1.

DISCUSSION

As shown in the experimental results, the proposed enhanced IDS framework has a state-of the art performance on the NSL-KDD benchmark dataset. A number of important findings can be made out of the overall assessment.

First, the hybrid feature selection approach is effective in dimensional reduction and also works well in enhancing the accuracy on classification. The approach makes use of both the relevance of individual features and the redundancy of features between them, by merging Information Gain with correlation based filtering. The 31.7 percent feature reduction corresponds to a reduction in training time with no impact to the quality of detection thus making the strategy appropriate to use in real-time deployment.

Second, SMOTE has great enhancement on the minority attack class detection especially U2R and R2L. These classes are grossly underrepresented in the training data by default after without oversampling, which leads to biased classifiers in terms of majority classes. The synthetic samples produced by SMOTE allow the models to acquire more robust decision boundaries of rare types of attacks, which is essential in the context of practical IDS deployment where even a single missed attack may be disastrous.

Third, the stacking ensemble performs better than the individual classifiers in all evaluation performances. This is enhanced by the variety of the base learners: tree-based algorithms effectively learn non-linear interactions between features, and the meta-learner is more effective in balancing out the input of each base classifier. The 0.09-0.18% accuracy gain over the best single classifier (XGBoost) can be considered a small change in percentage, but when it comes to network security, hundreds of extra correctly classified connections in a standard test set can be considered.

Fourth, the DNN has a competitive and somewhat worse performance than the multi-class classification ensemble methods. This aligns with literature evidence that gradient-boosted tree ensembles tend to perform better on tabular data with rather small feature spaces when using deep learning. Nonetheless, the DNN has scalability benefits, as well as, it is capable of learning raw features without explicit feature engineering.

Comparison with Existing Work

Table 6: Comparison with Existing Studies

Study	Method	Dataset	Accuracy (%)
Negandhi et al. (2019)	Random Forest	NSL-KDD	99.32
Chae et al. (2020)	Feature Selection + RF	NSL-KDD	99.41
Li et al. (2022)	Enhanced RF + SMOTE	NSL-KDD	99.72
Ahmed et al. (2024)	RF + XGBoost + DNN	NSL-KDD	99.60
Proposed Method	**Stacking Ensemble**	**NSL-KDD**	**99.80**

Table 6 will compare the suggested approach and the current research with the NSL-KDD data. The most accuracy of 99.80 with the proposed stacking ensemble of hybrid feature selection and SMOTE is the highest compared to all other approaches. The increase compared to the last similar study (Ahmed et al., 2024) is 0.20% which proves the efficiency of the offered contributions.

6.2 Limitations

Regardless of the strengthening outcomes, it is important to consider a number of drawbacks. Although NSL-KDD dataset is popular, it might not be representative of the recent network traffic trends and the up-to-date type of attacks. A stacking ensemble, although very precise, has a longer training time (45.3 seconds) than single models which can be taken into account in an environment where the model needs to be retrained frequently. Furthermore, the SMOTE method can create artificial samples that do not fully reflect the underlying data distribution, possibly creating artificial structures.

CONCLUSION AND FUTURE WORK

In this study, a more advanced Intrusion Detection System framework which incorporates hybrid feature selection, SMOTE-based class balancing, stacking ensemble learning, and deep neural networks has been described. Decades of experiments on the NSL-KDD benchmark show that the proposed approach is effective and provides 99.80 percent accuracy with binary classification and 99.53 percent accuracy with multi-class classification.

The four contributions of this work discuss the key issues of IDS development. The hybrid aspect of choice lowers the dimensions by 31.7 percent and enhances accuracy. SMOTE is an algorithm that is very important in detecting the minority attack classes (R2L and U2R). It is the complementary strength of the various base classifiers that gives maximum performance to the stacking ensemble. The DNN offers a competitive deep learning alternative, which is capable of growing bigger and more complex data.

Future research options are to test the proposed framework on more recent and more comprehensive datasets including CIC-IDS2017 and CSE-CIC-IDS2018, to test the advanced methods of oversampling like ADASYN and Borderline-SMOTE, to explore the idea of using attention mechanisms in the deep learning model, and to create a version of the system that would be able to adapt to the new patterns of attacks in real-time.

ETHICAL DECLARATION

Conflict of interest: No declaration required. **Financing:** No reporting required. **Peer review:** Double anonymous peer review.

REFERENCES

[1] Denning, D.E. (1987). An intrusion-detection model. IEEE Transactions on Software Engineering, SE-13(2), 222-232.
[2] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A.A. (2009). A detailed analysis of the KDD CUP 99 data set. In Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications.
[3] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.
[4] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
[5] Chawla, N.V., Bowyer, K.W., Hall, L.O., & Kegelmeyer, W.P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357.
[6] Negandhi, P., Trivedi, Y., & Mangrulkar, R. (2019). Intrusion detection system using random forest on the NSL-KDD dataset. In Emerging Research in Computing, Information, Communication and Applications, Springer.
[7] Wolpert, D.H. (1992). Stacked generalization. Neural Networks, 5(2), 241-259.
[8] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning.
[9] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929-1958.
[10] Ingre, B., & Yadav, A. (2015). Performance analysis of NSL-KDD dataset using ANN. In International Conference on Signal Processing and Communication Engineering Systems.
[11] Wahba, Y., ElSalamouny, E., & ElTawee, G. (2015). Improving the performance of multi-class intrusion detection systems using feature reduction. IJCSI International Journal of Computer Science Issues, 12(3).
[12] Chae, H., Jo, B., Choi, S.H., & Park, T. (2013). Feature selection for intrusion detection using NSL-KDD. Recent Advances in Computer Science, 184-187.
[13] Li, X., Chen, W., Zhang, Q., & Wu, L. (2022). Intrusion detection system combined enhanced random forest with SMOTE algorithm. EURASIP Journal on Advances in Signal Processing.
[14] Ahmad, I., Basheri, M., Iqbal, M.J., & Rahim, A. (2018). Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE Access, 6, 33789-33795.
[15] Kingma, D.P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
[16] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In Proceedings of the IEEE International Conference on Computer Vision.
[17] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
[18] Abadi, M., et al. (2016). TensorFlow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation.