Intrusion Detection For NSL-KDD Dataset

Rowan Mashaal
Department of Artificial Intelligence
University of Ottawa
Cairo, Egypt
rmash025@uottawa.ca

Hussien Hussien
Department of Artificial Intelligence
University of Ottawa
Cairo, Egypt
hhuss107@uottawa.ca

Shady Elsabow
Department of Robotics
University of Ottawa
Cairo, Egypt
selsa092@uottawa.ca

Muhammed Ali
Department of Robotics
University of Ottawa
Cairo, Egypt
mmuha095@uottawa.ca

I. *Abstract*

This research proposes a comprehensive evaluation framework for enhancing the performance of intrusion detection systems (IDS) through the application of supervised learning techniques. The primary focus is on a Voting Classifier, combining Decision Tree Classifier, Logistic Regression, and Support Vector Classifier (SVC), within a k-fold cross-validation framework. Feature selection methods such as Principal Component Analysis (PCA) and resampling techniques are employed to optimize the model's capabilities.

The study rigorously tests and evaluates the proposed approach on the NSL-KDD dataset, a valuable resource for network intrusion detection. Key evaluation metrics, including macro-avg precision, recall, F1-score, and F2-score, are used to measure the performance of the models. The Voting Classifier, after incorporating resampling and PCA feature selection, emerges as the focal point for analysis.

Furthermore, an alternative method involving clustering, termed "clustering forest," is introduced. This method utilizes k-means clustering to group training data, and each group is used to train a separate decision tree. The performance of this clustering forest is compared with a baseline Random Forest model.

Results indicate that the proposed enhancements, including PCA and a dual resampling strategy, contribute to improved IDS performance, particularly in detecting minority classes. The clustering forest method demonstrates promising results, outperforming the baseline Random Forest in terms of macro-avg F2 score and recall.

The study concludes by emphasizing the significance of macro-avg F2 score in evaluating intrusion detection systems, highlighting the effectiveness of clustering and resampling techniques in addressing class imbalances. Future work is suggested, exploring the integration of clustering forest with different models and clustering techniques, as well as investigating the impact of SVM hyperparameters on minority class detection.

Keywords—supervised learning, Feature Engineering, Machine Learning, Intrusion detection.

II. *Introduction*

In today's interconnected digital landscape, the protection of information and systems from unauthorized access and malicious activities stands as a paramount priority. Intrusion Detection Systems (IDS) constitute a pivotal element of comprehensive cybersecurity strategies. These systems play a crucial role in early threat detection and the continuous monitoring of network traffic and system content. They are designed to swiftly identify and respond to any suspicious or malicious activities in real-time. Importantly, IDS systems serve as the frontline defense that guards against potential breaches, data compromise, reduced downtime, and assists organizations in achieving compliance with regulatory requirements across diverse industries. In an ever-evolving cybersecurity landscape, IDS is indispensable in upholding the integrity, privacy, and availability of digital assets.

Two primary categories of IDS exist [1]: Network-based IDS (NIDS) and Host-based IDS (HIDS). NIDS specializes in monitoring network traffic, scrutinizing patterns, and actions that may signify attacks or anomalies. It excels at detecting both known attack signatures and emerging threats, making it invaluable in the context of extensive network environments. Conversely, HIDS concentrates on the activities of individual host systems, diligently analyzing system logs, file alterations, and user behaviors to pinpoint unauthorized access and anomalous activities. Together, NIDS and HIDS form a cohesive line of defense in the ever-changing world of cybersecurity.

Machine learning has brought about a transformative revolution in the realm of IDS, offering an array of advanced strategies, including supervised learning. In supervised intrusion detection systems (IDS), labeled datasets are utilized to train classifiers by amalgamating known malware signatures with benign network traffic data. These classifiers, once trained, become proficient in recognizing and categorizing network traffic as malicious or benign. A wide spectrum of machine learning algorithms finds application in this context, including decision trees, random forests, support vector machines (SVM), neural networks, and more. Notably, the emergence of Extreme Learning Machine (ELM) [2] has garnered significant attention

in the IDS domain due to its exceptional speed and flexibility, setting it apart from conventional methods and proving to be a valuable tool in the realm of intrusion detection. As the cybersecurity landscape continues to evolve, supervised IDS, bolstered by machine learning, stands as a robust and adaptive defense mechanism against a multitude of threats.

III. RELATED WORK

In the paper [3], an innovative intrusion detection system is introduced that combines a hybrid classifier with profile enhancement techniques to effectively identify unusual user behavior. It leverages anomaly detection through supervised learning on event logs, initially creating a standard user profile from historical data, such as the KDD CUP 99 and NSL-KDD datasets, to detect deviations and potential intrusions. The hybrid classifier, merging Naïve Bayes and Support Vector Machine (SVM), boosts accuracy and reduces false positives. Additionally, it enhances user profiles with session-based features, resulting in a significant performance improvement, achieving an accuracy rate of 0.931 and a precision rate of 0.958. While this approach enhances accuracy and reduces false positives, it can be computationally intensive, relies on data quality, poses complexity challenges, risks false negatives, and may have limited generalizability to specific contexts.

In the paper [4], a machine learning-based intrusion detection methodology for cybersecurity, encompasses preprocessing, feature selection, parameter optimization, and classification phases. It effectively employs Correlation-based Based Feature Selection and machine learning algorithms such as Random Tree, AdaBoost, K-Nearest Neighbor, and Support-vector machines for intrusion classification. The evaluation on two extensive datasets, NSL-KDD and CIC-DDoS2019 shows its high detection rate and low false positive rate, enhancing network security. While the method's multi-phase nature may demand significant computational resources and domain expertise, it excels in adaptability, feature selection efficiency, and parameter optimization, although broadening the evaluation dataset and continuous adaptation to evolving threats are essential considerations.

In [5], the authors address the growing importance of intrusion detection in the face of increasing cyber threats and network security vulnerabilities. They introduce and compare two primary approaches: traditional machine learning and deep learning, specifically utilizing Convolutional Neural Networks (CNN). These methods are evaluated using the KDD-CUP99 dataset and a variety of machine learning algorithms, including KNN, Random Forest, Naive Bayes, Support Vector Machine for traditional machine learning, and CNN, ResNet, and DenseNet for deep learning. Traditional machine learning stands out due to its computational efficiency, interpretability, and resilience to noise in smaller datasets, but it requires manual feature engineering. In contrast, deep learning excels in achieving high accuracy through automatic feature extraction and has the potential to scale with larger datasets. However, it demands substantial computational resources, extensive labeled data, regularization techniques, and presents challenges in terms of model complexity and interpretability.

In [6], The research paper introduces an adaptive ensemble machine learning model that stands as a formidable force in the realm of intrusion detection. With a remarkable accuracy rate of 85.2%, this model leverages the combined power of multiple base classifiers and employs adaptive voting, endowing it with robustness in identifying an array of network intrusions. A comprehensive suite of evaluation metrics, including accuracy, precision, recall, and the F1 score, meticulously gauge its performance, affirming its effectiveness in distinguishing between benign and malicious network traffic. Despite the considerable achievements, challenges arise, notably stemming from dataset issues such as redundant records and data imbalances that may influence its performance on specific attack types. Nevertheless, the model's capacity to fortify network security through its high accuracy and adaptability is a substantial contribution to the field of cybersecurity, exemplifying the ongoing commitment to enhancing intrusion detection capabilities and reinforcing the digital realm against ever-evolving and increasingly sophisticated threats.

In paper [7], a new method for achieving more accurate detection of multiple types of attacks the method works in three distinct steps in each of these. Initially, the Extra Trees classification is used to select individual features for each attack for each Extreme Learning Machine (ELM). Then, an ELM is developed to detect each independent attack. Finally, the results from all ELMs are combined with the SoftMax level, further increasing the accuracy by reducing the results. Extensive testing was conducted on the UNSW and KDD Cup 99 data sets, and the results clearly show that the proposed method outperforms all other existing methods by a large margin. The system achieves an impressive accuracy of 98.24% and 99.76%. The strengths of the paper are working on ELM as an approach is highly reliable as it’s effective with data imbalance. Also testing the approach on 2 different datasets is a distinct evaluation approach. But the approach is highly complex which will make it unreliable in real-time detection.

Research [8] presents a hybrid machine learning approach that combines a selection approach, representing a supervised learning approach and a data reduction approach as unsupervised learning to build an appropriate selection model a relevant and important feature importance decision tree-based method with iterative feature elimination and local outlier implementation factor (LOF) method is implemented through anomaly/out-of-site data detection Experimental results show that proposed The method achieves the highest accuracy in

detecting R2L (i.e., 99.89%) and holds better for other types of attacks than most other experiments on the NSL-KDD dataset, thus making it practical more complex than others. There are several complications with the UNSW-NB15 data set of binary categories. The approach is really effective with the dataset as it has outliers. As proven by experiment it has achieved high accuracy close to 100%. However, the evaluation metrics depend only on accuracy, using the false negative alarm is better in the case of intrusion detection.

IV. METHODOLOGY

In our pursuit to enhance the performance of intrusion detection systems and optimize their capabilities, we propose a comprehensive evaluation design that compares various supervised learning techniques. Our approach includes employing a Voting Classifier, combining base models such as Decision Tree Classifier, Logistic Regression, and Support Vector Classifier (SVC). This ensemble approach aims to leverage the strengths of different models for improved classification of network traffic, particularly in multiclass intrusion detection scenarios.

To ensure that our model incorporates the most relevant features, we employ different feature selection methods. We utilize principal component analysis (PCA) to fine-tune the input features. Additionally, we leverage feature importance from ensemble models to enhance the discriminatory power of our models.

To further refine our experimental framework, we incorporate k-fold cross-validation in the base models both before and after implementing feature selection and resampling techniques. Initially, we apply k-fold cross-validation before feature selection and resampling to assess the robustness of the base models. Subsequently, we utilize k-fold cross-validation after the application of resampling and feature selection. It's noteworthy that the resampling procedure is specifically applied to the training data.

Our rigorous testing and evaluation are conducted on the NSL-KDD dataset, which encompasses a diverse range of network traffic data, including both normal and malicious instances with distinct attack types. Key evaluation metrics such as macro-avg(precision, recall, F1-score, F2-score), and the confusion matrix are used to measure the performance of our models.

The focal point of our study is the Voting Classifier, combining Decision Tree Classifier, Logistic Regression, and SVC, and its performance after the incorporation of resampling and PCA feature selection within a k-fold cross-validation framework. This approach enables us to identify not only the top-performing model but also the combination of models and feature selection methods that yield the most robust intrusion detection system.

We anticipate that the results of this study will provide valuable insights for enhancing network security practices and contribute to the development of more effective intrusion detection systems.

In method 2 clustering using k-means will be used to group the training data into smaller groups, number of groups will be a parameter that can be tuned later. After this the trained k-means clustering model will be kept and used later on. Then each group of data will be used to train a separate decision-tree.

V. Results and Analysis

Dataset

The NSL-KDD dataset, known as the "NSL-KDD Network Intrusion Detection Dataset," is a valuable resource for the evaluation of intrusion detection systems and machine learning models in the realm of network security. With approximately 42 features available for each network connection record, it offers a dataset of practical size, comprising about 11849 training instances and 22,544 test instances. This manageable dataset covers a diverse range of features, describing network traffic and system activities. It is categorized into multiple classes, including normal network traffic, Denial of Service (DoS) attacks, probes, unauthorized access from remote machines (R2L), and unauthorized access to the root (U2R).

Evaluation*:*

Since the problem is to get the type of the attack accurately as much as possible, The Macro average recall is more critical than the macro average precision.

Macro average Precision: Evaluates the proportion of true positive predictions out of all predicted positives, useful for minimizing false positives.

Macro avg Recall (Sensitivity): Calculates the proportion of true positive predictions out of all actual positives, important for identifying as many positive instances as possible.

Macro avg F2 score gives more weight for recall.

Method_1

Baseline model

In our baseline model, we utilized a Voting Classifier integrating Decision Tree Classifier, Logistic Regression, and Support Vector Classifier (SVC). To ensure a comprehensive assessment, we employed k-fold cross-validation with 5 folds. This approach forms the foundation for evaluating the collective performance of our model across various subsets of the dataset, providing a robust basis for subsequent analyses and comparisons.

Improvements

We implemented several enhancements to our baseline model.

First improvement

we incorporated Principal Component Analysis (PCA) into the baseline model, aiming to improve its feature representation and overall performance.

Second improvement

we introduced resampling techniques, initially employing oversampling and subsequently combining it with undersampling. This dual resampling strategy was applied to address potential imbalances in the dataset and enhance the model's robustness.

Third improvement

We extended the use of PCA to the resampled data, further refining the feature selection process in the context of the modified dataset.

Results

For baseline model

In the evaluation of our baseline model, we achieved a macro-averaged F2_score of 0.5, with consistent results observed across each fold.

	precision	recall	f1-score
accuracy -	0.97	0.97	0.97
macro avg -	0.55	0.50	0.51
weighted avg -	0.97	0.97	0.97

Fig (1) Classification report for baseline model

For first improvement

incorporating PCA into the baseline model, we observed a modest enhancement, resulting in a macro-averaged F2_score of 0.515.

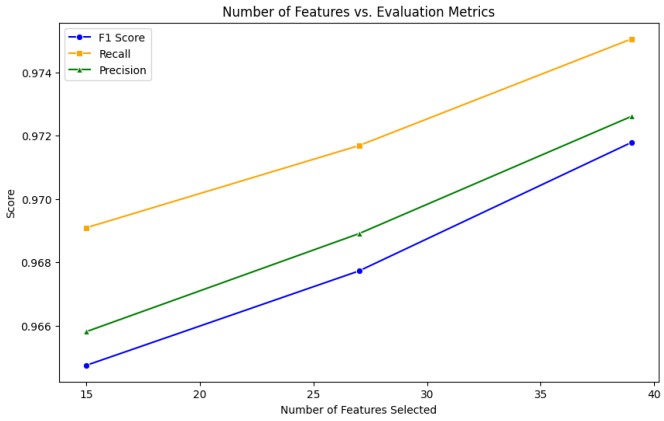

Fig (2) Number of features vs score

As we shown in figure if number of features increases, recall increases.

For second improvement

Introducing a dual resampling strategy with oversampling and subsequent undersampling, yielded a further improvement, with the macro-averaged F2_score reaching 0.528.

	precision	recall	f1-score
accuracy -	0.88	0.88	0.88
macro avg -	0.40	0.68	0.46
weighted avg -	0.96	0.88	0.91

Fig (3) Classification report for second improvement

For third improvement

involving the application of PCA to the resampled data, showcased the most substantial enhancement, resulting in a macro-averaged F2_score of 0.587. This indicates a notable advancement in the model's overall performance, highlighting the effectiveness of the combined feature selection and resampling techniques.

	precision	recall	f1-score
accuracy -	0.89	0.89	0.89
macro avg -	0.38	0.64	0.43
weighted avg -	0.96	0.89	0.91

Fig (4) Classification report for third improvement

Analysis of results

The papers discussed how ensembling techniques performed in intrusion detection while changing the methodology of the voting.

For baseline model

For the base-line models it is shown that the voting classifier performed well in detecting whether the data sample represents an attack or not, but it got only a macro avg f2 score around 50 percent. And around 50 percent for the macro avg recall Which means that nearly 97% of the attacks have been caught but only 50% of their types have been correctly recognized.

For first improvement

For the base-line models after using PCA, it is shown that the voting classifier performed well in detecting whether the data sample represents an attack or not, but it got only a macro avg f2 score around 51.5 percent. And around 51.5 percent for the macro avg recall Which means that nearly 97% of the attacks have been caught but only 56% of their types have been correctly recognized.

For second improvement

For the base-line models after using PCA, it is shown that the Voting classifier performed well in detecting whether the data sample represents an attack or not, but it got only a macro avg f2 score around 52.8 percent. And around 52.8 percent for the macro avg recall Which means that nearly 91% of the attacks have been caught but only 52.8% of their types have been correctly recognized.

For third improvement

For the base-line models after using PCA, it is shown that the Voting Classifier performed well in detecting whether the data sample represents an attack or not, but it got only a macro avg f2 score around 58.7 percent. And around 58.7 percent for the macro avg recall Which means that nearly 91% of the attacks have been caught but only 58.7% of their types have been correctly recognized.

Summary of results

In the baseline model, class 17 exhibited a recall of 67. After implementing feature selection and sampling techniques, specifically PCA and a dual resampling strategy, the recall for class 17 significantly improved to 94. This underscores the effectiveness of the applied enhancements in enhancing the model's ability to correctly identify instances belonging to class 17, reflecting a notable boost in sensitivity for this class.

Method2 - clustering forest:

Base line model:

The papers discussed how ensembling techniques performed in intrusion detection with changing the methodology of the voting. In this case A random forest model will be used as a baseline model. And the score will be stored for later discussion.

The suggested improvement:

The name of the method is chosen by the team. In this method clustering using k-means will be used to group the training data into smaller groups, number of groups will be a parameter that can be tuned later. After this the trained k-means clustering model will be kept and used later on. Then each group of data will be used to train a separate decision-tree.

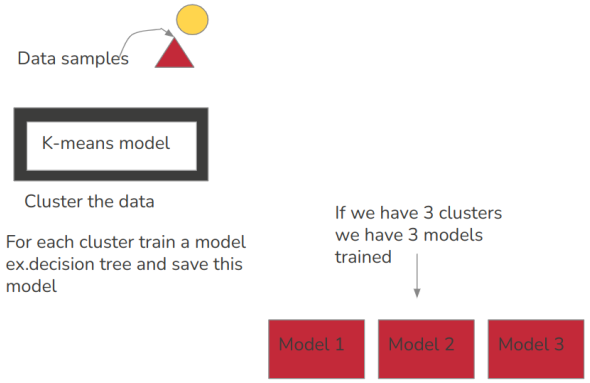

Fig (5): how clustering forest training process works.

As shown in figure below the trained k-means model will be used now to predict the cluster of each testing data sample, for example, cluster 3. Each cluster of the data has its own unique decision tree that has already been trained on. So, the decision tree that was already trained on cluster 3 should predict the class of this sample from the testing data. Also, there is a difference between this method and the random forest method, that this method does not have voting because each classifier is the only classifier dedicated for a group of data.

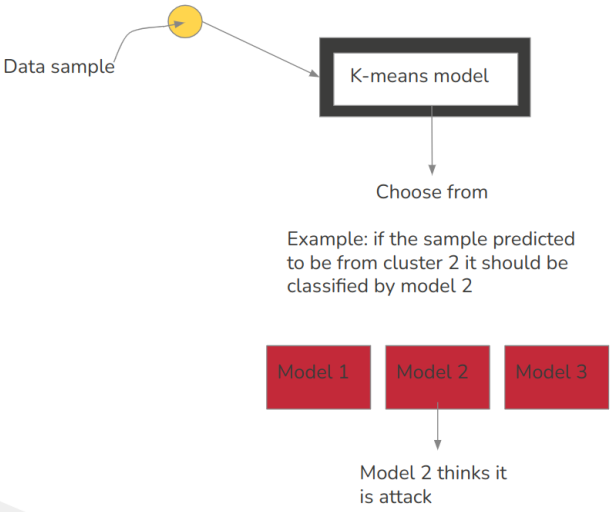

Fig (6): how clustering forest testing process works.

Hyperparameters: the hyper parameter here is the k for the k-means model. Which represents the number of groups and models that will be trained. Since each cluster will train a single model.

Results:

The following results will show the performance of the base-line model which is Random Forest.

30	0.73	0.53	0.62	15
31	0.98	0.94	0.96	267
32	0.33	0.25	0.29	4
33	0.00	0.00	0.00	1
34	0.00	0.00	0.00	8
35	1.00	1.00	1.00	13761
36	0.97	0.93	0.95	40
37	1.00	1.00	1.00	993
38	0.00	0.00	0.00	1
39	0.72	0.94	0.82	99
accuracy			0.99	44555
macro avg	0.61	0.56	0.57	44555
weighted avg	0.99	0.99	0.99	44555

Fig (7): The base-line model classification report - random forest

	precision	recall	f1-score	support
0	0.99	0.99	0.99	17436
1	1.00	1.00	1.00	27119
accuracy			1.00	44555
macro avg	1.00	1.00	1.00	44555
weighted avg	1.00	1.00	1.00	44555

Fig (8): the base-line model classification report that in case of binary classification Attack vs -non Attack

The coming results will show the performance of the chosen method - clustering forest:

32	0.29	0.50	0.36	4
33	0.50	1.00	0.67	1
34	0.00	0.00	0.00	8
35	1.00	1.00	1.00	13761
36	0.95	0.95	0.95	40
37	1.00	1.00	1.00	993
38	0.33	1.00	0.50	1
39	0.97	0.99	0.98	99
accuracy			0.99	44555
macro avg	0.72	0.72	0.70	44555
weighted avg	0.99	0.99	0.99	44555

Fig (9): The clustering forest method classification report.

	precision	recall	f1-score	support
1	1.00	1.00	1.00	44555
accuracy			1.00	44555
macro avg	1.00	1.00	1.00	44555
weighted avg	1.00	1.00	1.00	44555

Fig (10): the clustering forest classification report in case of binary classification Attack vs -non Attack

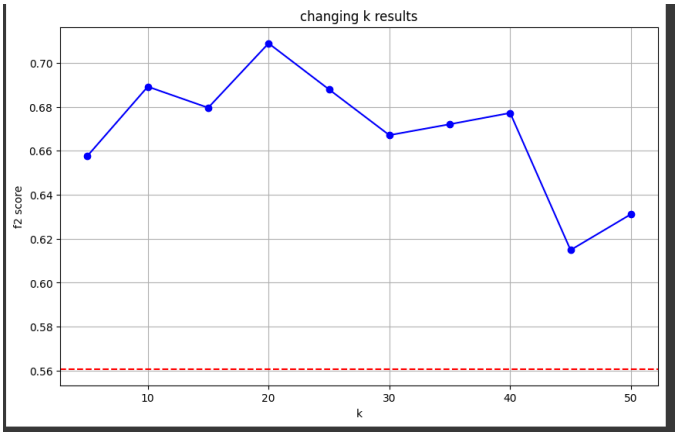

Fig (11): Macro avg F2 score comparison between different values of parameter k in blue and the base- line model in red.

Analysis of the Method:
For the base-line models it is shown that the random forest performed well in detecting whether the data sample represents an attack or not, but it got only a macro avg f2 score around 56.00 percent. And around 56 percent for the macro avg recall Which means that nearly 99% of the attacks have been caught but only 56% of their types have been correctly recognized.
For the improved model -clustering forest- it performed will in detecting whether the data sample represents an attack or not, and it got only a macro avg f2 score around 72 percent for k =20 . And around 72% percent for the macro avg recall Which means that nearly 99% of the attacks have been caught but only 72% of their types have been correctly recognized. This improvement happened because the clustering technique provides a certain group for outliers, also the -minority classes- that have a small amount of data records will be grouped with less number of majority classes, This will make the model less biased to the majority classes.

VI. Conclusion

Previous results showed how clustering gives extra knowledge about the nature of intrusion detection systems. Also the macro average F2 score is an effective measurement for intrusion detections because it is not affected by the majority classes since all classes have the same weight and it focuses more on the recall since our main problem is to get the type of the attack. Not only detecting whether it is an attack or not. We learned also that the data resampling is effective in improving the minority classes detection.

VII. Future work:

In future work the clustering forest can be used with different models like SVM and another clustering technique like DBSCAN. This can provide more insights about the efficiency of different clustering techniques and how it can increase the ability of detecting minority classes correctly. Also will show

the effect of SVM hyperparameters on separating the minority classes. Also the merging between resampling data and clustering forest.

VIII. REFERENCES

[1] Khraisat, A., Gondal, I., Vamplew, P. et al. Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecur 2, 20 (2019). https://doi.org/10.1186/s42400-019-0038-7

[2] W. Yang, S. Wang, and M. Johnstone, "A Comparative Study of ML-ELM and DNN for Intrusion Detection," in Proceedings of the 2021 Australasian Computer Science Week Multiconference (ACSW '21), February 2021, pp. 1-7. DOI: 10.1145/3437378.3437390.

[3] P. Pokharel, R. Pokhrel and S. Sigdel, "Intrusion Detection System based on Hybrid Classifier and User Profile Enhancement Techniques," 2020 International Workshop on Big Data and Information Security (IWBIS), Depok, Indonesia, 2020, pp. 137-144, doi: 10.1109/IWBIS50925.2020.9255578.

[4] A. A. Yilmaz, "Intrusion Detection in Computer Networks using Optimized Machine Learning Algorithms," 2022 3rd International Informatics and Software Engineering Conference (IISEC), Ankara, Turkey, 2022, pp. 1-5, doi: 10.1109/IISEC56263.2022.9998258.

[5] R. Liu, "Multivariate Network Intrusion Detection Methods Based on Machine Learning," 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 2023, pp. 148-155, doi: 10.1109/EEBDA56825.2023.10090554.

[6] X. Gao, C. Shan, C. Hu, Z. Niu, & Z. Liu, "An Adaptive Ensemble Machine Learning Model for Intrusion Detection," 2019 IEEE Special Section on Artificial Intelligence in Cybersecurity, Changchun, China, pp. 9-16, doi: 10.1109/ACCESS.2019.2923640.

[7] J. Sharma, C. Giri, O.-C. Granmo, and M. Goodwin, "Multi-layer intrusion detection system with ExtraTrees feature selection, extreme learning machine ensemble, and softmax aggregation," EURASIP Journal on Information Security, vol. 2019, no. 1, 2019. [Online]. Available: https://doi.org/10.1186/s13635-019-0098-y.

[8] Megantara, A.A., Ahmad, T. A hybrid machine learning method for increasing the performance of network intrusion detection systems. *J Big Data* 8, 142 (2021). https://doi.org/10.1186/s40537-021-00531-w