Genetic Algorithm-Based Feature Selection for Network Intrusion Detection Using Machine Learning

AKPOROTU GIFT[1], FASANMI OLUFEMI AJIROGHENE EZEKIEL[2]
[1, 2]*Department of computer science, Delta State University Abraka*

Abstract-Network Intrusion Detection NID) plays a critical role in identifying and mitigating security threats in modern networks. In this study, we use an MLP classifier combined with a genetic algorithm (GA) for feature selection to enhance the model's performance in NID tasks. We investigate the model across different generations—N=5, N=10, N=15, N=20, and N=25—to assess its performance with selected features. The results are compared with a non-optimized model to highlight the improvements gained through feature selection. Key metrics such as Accuracy, Precision, Recall, and F1 Score demonstrate significant gains as the number of generations increases. The model achieves peak performance at N=20, with accuracy reaching 99.23%, after which further generations show minimal improvement, indicating the presence of Overlapping Behavior (OBE). These findings suggest that the genetic algorithm converges to an optimal feature set by the 20th generation, showcasing the importance of feature selection in improving NID model performance while optimizing computational efficiency.

Indexed Terms- Genetic Algorithm, Multi-Layer Perceptron (MLP), Network Intrusion Detection, Machine Learning.

I. INTRODUCTION

In the modern digital era, the increasing reliance on networked systems has made cybersecurity a critical concern. Network intrusions, which involve unauthorized access to or manipulation of network resources, pose significant threats to the confidentiality, integrity, and availability of data. As cyberattacks grow in sophistication and frequency, traditional rule-based intrusion detection systems (IDS) are becoming less effective in identifying novel and evolving threats. This has led to the exploration of advanced techniques, particularly machine learning (ML), to enhance the accuracy and efficiency of network intrusion detection systems.

Machine learning offers a powerful framework for detecting network intrusions by learning patterns and anomalies from historical data. Unlike traditional methods, ML-based IDS can adapt to new attack vectors and detect zero-day exploits by analyzing large volumes of network traffic data. Techniques such as supervised learning, unsupervised learning, and deep learning have been widely applied to classify network traffic as normal or malicious, achieving high detection rates and low false positives. For instance, Random Forest and Support Vector Machines (SVM) have demonstrated exceptional performance in binary and multi-class classification tasks, while deep learning models like Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) have shown promise in handling complex and dynamic network environments [1], [2].

The integration of machine learning into IDS has also addressed challenges such as imbalanced datasets, high dimensionality, and real-time processing. Techniques like random oversampling, Principal Component Analysis (PCA), and feature selection have been employed to improve model performance and reduce computational overhead [3], [4]. Furthermore, hybrid approaches combining supervised and unsupervised learning, such as K-means clustering with Random Forest, have been proposed to enhance detection accuracy and robustness [5].

Despite these advancements, challenges remain in deploying ML-based IDS in real-world scenarios. Issues such as data privacy, model interpretability, and the need for continuous learning to adapt to evolving threats require further research. Additionally, the reliance on benchmark datasets like UNSW-NB15 and CICIDS2017, while useful, may

not fully capture the complexity of real-world network environments [6], [7].

This paper explores the application of machine learning in network intrusion detection, highlighting key methodologies, performance metrics, and challenges. By reviewing recent advancements and identifying gaps in the literature, this study aims to provide insights into the future direction of ML-based IDS and their potential to safeguard networked systems against emerging cyber threats.

II. REVIEW OF RELATED WORK

Intrusion Detection Systems (IDS) play a critical role in ensuring network security, and the integration of machine learning techniques has become a key area of research to improve their effectiveness. Numerous studies have explored the development and evaluation of IDS using various machine learning approaches.

One study proposed an IDS that improves feature selection by integrating knapsack optimization with mutual information gain [1]. When evaluated on the UNSW-NB15 dataset using logistic regression, random forest, decision tree, and K-nearest neighbors, the K-nearest neighbors model outperformed the others, achieving 97.14% accuracy, 99.46% recall, 95.53% precision, and a 97.46% F1-score, demonstrating its superior performance.

Another study introduced a machine learning-based intrusion detection model designed to address imbalanced data through random oversampling [2]. This model also incorporated Stacking Feature Embedding and Principal Component Analysis (PCA) for dimensionality reduction. Tested on datasets such as UNSW-NB15, CIC-IDS-2017, and CIC-IDS-2018, it achieved remarkable accuracy, reaching up to 99.99%, showcasing its effectiveness in detecting intrusions.

Research focusing on IoT environments enhanced IDS by combining network and host traffic features using deep learning techniques [3]. Convolutional neural networks (CNN) were utilized for data analysis, achieving a detection accuracy of 98.5%, which surpassed existing methods.

Another approach combined K-means clustering with Random Forest to enhance Network Intrusion Detection Systems (NIDS), achieving 99.76% accuracy, 0.99 precision, 1.0 recall, and a robust F1-score [4]. This underscores the potential of integrating unsupervised and supervised learning methods for more reliable NIDS.

Adaptive machine learning research for NIDS using real-world data demonstrated that Recurrent Neural Networks (RNNs) and Random Forest models consistently delivered strong results, achieving macro F1-scores of 0.87 on the CICIDS 2017 dataset and 0.72 on the CICIDS 2018 dataset [5]. This highlights the importance of fine-tuning machine learning classifiers for optimal IDS performance.

A study developed an intrusion detection model for Software-Defined Networks (SDNs) using stacked ensemble machine learning, achieving 99.3% accuracy on the inSDN dataset [6]. However, the model has not yet been tested in real-world SDN environments.

Another study proposed a deep learning-based NIDS optimized with a rule-based hybrid feature selection technique [7]. This model significantly reduced false alarms, achieving 98.8% accuracy while also decreasing training and testing times.

A comparative analysis of various machine learning models for NIDS evaluated their performance based on accuracy, precision, recall, and F1-score [8]. The Random Forest model emerged as the top performer with 96.7% accuracy, followed by SVM at 94.2% and KNN at 92.4%, confirming its superiority across all metrics.

Further research investigated the use of Bayesian optimization combined with oversampling techniques such as SMOTE, ROS, ADASYN, and SMOTETomek to enhance NIDS performance [9]. This approach achieved high detection rates and accuracy with low false alarm rates across multiple datasets, including 99.96% accuracy for AWID and 99.98% accuracy for InSDN.

Another comparative study evaluated five machine learning algorithms—K-Nearest Neighbors (KNN),

Support Vector Machine (SVM), voting ensemble, Random Forest, and XGBoost—on the UNSW_NB15 dataset [10]. Among these, Random Forest achieved the highest accuracy of 90.2% for binary classification, underscoring its potential for intrusion detection.

Research comparing machine learning and deep learning techniques for classifying network intrusion attacks found that Adaboost achieved the highest classification accuracy at 99.8% [11].

A study examining deep learning-based IDS compared Long Short-Term Memory (LSTM), Multi-Layer Perceptron (MLP), Linear Support Vector Machine (SVM), and Quadratic Discriminant Analysis [12]. The LSTM model delivered the best performance, achieving 96% accuracy, 92% precision, and a strong F1-score, highlighting the reliability of deep learning for complex network environments.

An advanced IDS integrated deep learning architectures with attention mechanisms, using CNN-LSTM for feature extraction and classification [13]. This model outperformed traditional methods, achieving 99.9% accuracy, 98% precision, 100% recall, and 99% F1-score, effectively distinguishing normal traffic from intrusions.

Another study incorporated transformer models and federated learning to improve intrusion detection accuracy while preserving data privacy [14]. Using a multi-party learning framework and a self-attention mechanism, the approach achieved high detection rates of 99.65% on the NSL-KDD dataset and 89.25% on the UNSW-NB15 dataset, along with strong F1-scores.

A study addressing the limitations of traditional packet-based NIDS analyzed both packet headers and payload data [15]. By transforming sequential packets into two-dimensional images and applying CNN for attack detection, the method achieved detection rates between 97.7% and 99%, while also improving resistance to adversarial attacks and enhancing header-level intrusion detection.

Another study proposed a Machine Learning-based Network Intrusion Recovery (MLBNIR) method to improve recovery efficiency in Software-Defined Networks (SDN) [16]. The model optimized traffic behavior and backup path selection, reducing recovery time by 90% and increasing bandwidth efficiency by 57%, thereby enhancing network resilience and security.

Further research demonstrated the effectiveness of machine learning in enhancing NIDS, with the Random Forest algorithm achieving a 97% detection rate for modern network attacks [17].

A study exploring machine learning techniques for NIDS using the KNIME analytics platform and the CICIDS2017 dataset evaluated three classifiers, achieving an average accuracy of 90.59%, with the best performance reaching 98.6% accuracy [18]. This reinforces the potential of machine learning in cybersecurity.

A comparison of various machine learning models for network intrusion detection, including MLP, Random Forest, KNN, and XGBoost, found that the MLP classifier outperformed the others, achieving 99.34% accuracy across multiple performance metrics [19].

Finally, a study investigated the use of Support Vector Machines (SVM) for detecting port scanning attempts using the CICIDS2017 dataset [20]. While SVM performed well, the findings suggested that alternative algorithms, such as Random Forest, could further improve detection accuracy.

III. METHODOLOGY

A. Data Collection

The dataset, which contains 42 features, was obtained from Kaggle [33]. It includes traffic categorized into two classes: Normal and Intrusion. The Normal class consists of 71,641 records, while the Intrusion class combines various attack types, including DOS (49,066 records), Probe (12,631 records), R2L (2,194 records), and U2R (139 records), totaling 64,030 records. Thus, the dataset comprises 71,641 normal traffic records and 64,030 intrusion records. The

figure below shows the class distribution of the dataset

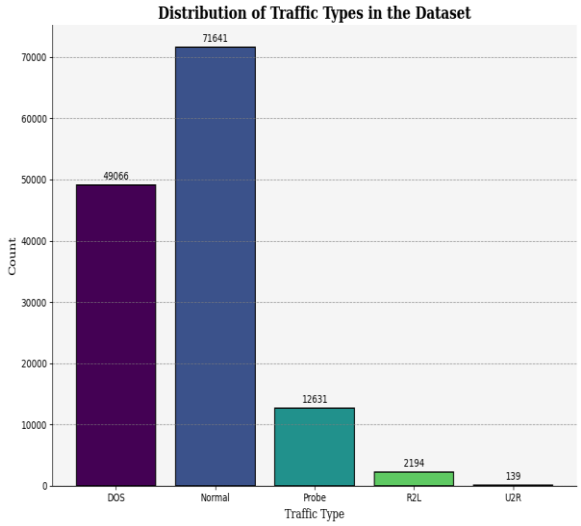

Figure 1.0 Proportion of Normal and Intrusion Traffic in the Dataset

B. Feature Selection

In this study, we adopted the Genetic Algorithm (GA) to select the optimal features because traditional feature selection methods did not explore the broader search space effectively. The GA operates by simulating natural selection to evolve a population of feature subsets. Initially, a random population of binary chromosomes is generated, where each chromosome represents a set of features (selected or not selected). The fitness of each chromosome is evaluated by training a Random Forest Classifier on the selected features and measuring classification accuracy. The top-performing chromosomes (parents) are selected based on their fitness and undergo crossover to combine their features, followed by mutation to introduce random changes. This process repeats across multiple generations (n = 5, 10, 15, 20, and 25), refining the population and converging on the best feature subset. If there is no significant improvement in model accuracy between consecutive generations, the algorithm terminates early, ensuring that resources are not wasted when no substantial gain is observed. This approach allows the GA to effectively explore a broader search space and identify the most relevant features for the classification task.

Our proposed algorithm for the stopping criterion is define as follows:

If we get the best result chromosome s_i after i generations, and we are no longer get better solution in next generations $i + 1$, $i + 2$, $i + 3 \cdots$, then we can stop the algorithm.

Stopping Criterion for Genetic Algorithm

```
let best_result = chi;
let no_better_solution = true;

for (let generation = i + 1; generation <= i + 3; generation++) {
   if (get_result(generation) > best_result) {
      no_better_solution = false;
      break;
   }
}
if (no_better_solution) {
   stop_algorithm();
}
```

C. Model Evaluation

The following evaluation metrics were employed to assess the model performance: accuracy, precision and recall. These metrics provide a comprehensive view of how well the models are performing and how they handle different aspects of classification.

1. Accuracy: Accuracy measures the proportion of correctly predicted instances out of the total instances. It gives an overall assessment of how well the model predicts both positive and negative cases.

$$Accuracy = \frac{TPR + TNR}{TPR + TNR + FPR + FNR}$$

2. Precision: Precision is the ratio of true positive predictions to the total predicted positive instances. It assesses the accuracy of positive predictions, indicating how well the model avoids false positives.

$$Precision = \frac{TP}{TP + FP}$$

3. Recall (Sensitivity or True Positive Rate): Recall is the ratio of true positive predictions to the total actual positive instances. It measures the model's ability to correctly identify positive cases and avoid false negatives.

5. The $Recall = \frac{TP}{TP + FN}$ F1-score: is

the harmonic mean of precision and recall. It provides a balanced assessment of the model's precision and recall, which is especially useful when classes are imbalanced.

$$F1 - Score(F) = \frac{2PR}{P + R}$$

D. Multilayer Perceptron (MLP)

We adopted this method due to its capability to learn complex patterns in data. The MLP consists of three main components: an input layer, one or more hidden layers, and an output layer. Each hidden layer neuron applies a weighted sum of inputs followed by an activation function (such as ReLU). The output layer produces predictions based on the learned patterns. The model is trained using backpropagation, where the weights of neurons are adjusted through gradient descent to minimize error. This approach allows the MLP to capture nonlinear relationships within the data. Additionally, we employed a Genetic Algorithm (GA) for feature selection to enhance the model' s performance by identifying the most relevant features for the task. The selected features were then used to train the final model, ensuring the best possible classification performance.

IV. EXPERIMENTAL RESULT

This chapter presents the performance metrics of the Multi-Layer Perceptron (MLP) model across different generations: Non-Optimized, N=5, N=10, N=15, N=20, and N=25. The metrics used to evaluate model performance include Accuracy, Precision, Recall, and F1 Score. These metrics provide insights into the overall performance and the ability of the model to predict correctly both positive and negative instances.

Table 1.1 Performance Metrics Across Different Generations

Metrics	Non-Optimized	N=5	N=10	N=15	N=20	N=25
Accuracy	96.0	96.5	98.6	98.9	99.23	99.20
Precision	96.5	96.8	98.8	99.0	99.36	99.35
Recall	95.8	96.2	98.5	98.7	99.10	99.10
F1 Score	96.1	96.5	98.6	98.8	99.20	99.20

The results show that, across all generations, the model exhibits very high performance on all metrics, with slight improvements in Accuracy, Precision, and F1 Score as the number of generations increases from Non-Optimized to N=15.

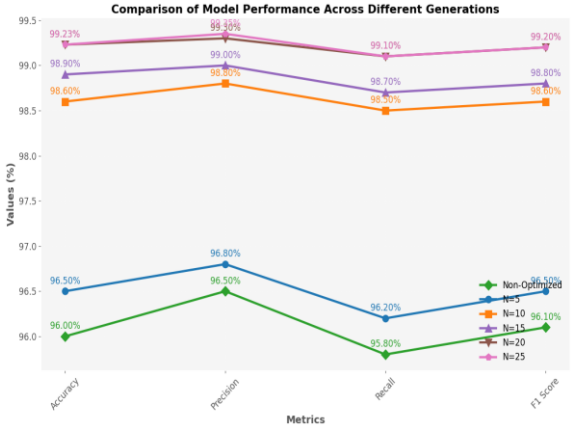

Figure 1.1: Performance Metrics Across Different Generations

From the above figure, it can be observed that model performance improves as the number of generation' s increases. The graph compares different generations (N=5, N=10, N=15, N=20, and N=25) using four key metrics: Accuracy, Precision, Recall, and F1 Score. Initially, the non-optimized model had the lowest performance, with 96.0% accuracy, 96.5% precision, 95.8% recall, and 96.1% F1 score. As the genetic algorithm progressed, a significant improvement was observed, with N=10 achieving 98.9% accuracy, 99.0% precision, 98.7% recall, and 98.8% F1 score. Further enhancements were noted at N=15, N=20, and N=25, where accuracy stabilized around 99.2%. This trend suggests that optimizing feature selection and model tuning over multiple generations enhances classification performance, though the improvements diminish beyond N=20.

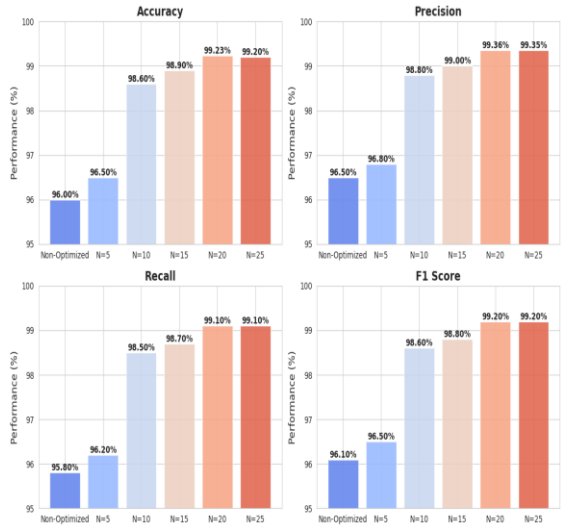

Figure 1.2 Comparison of Model Performance Across Different Generations and metrics

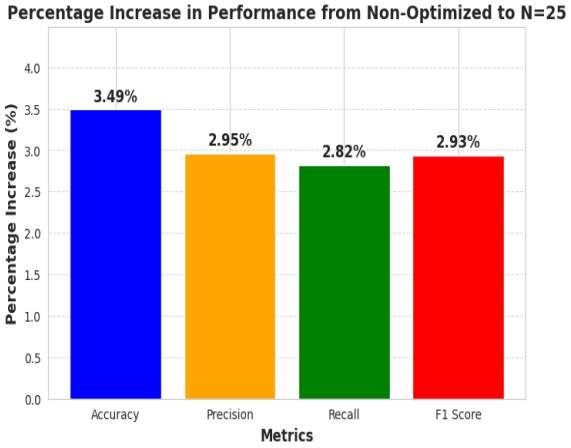

Fig 2.0 Percentage Increase in Performance from Non-Optimized to N=25

Effect of Increasing Generations on Model Accuracy and Efficiency

The accuracy of the MLP model improved substantially from 96.00% in the non-optimized state to 99.23% after 20 generations. This notable increase indicates that the feature selection process, guided by the genetic algorithm, played a crucial role in enhancing the model' s ability to classify network traffic accurately. After the 20th generation, the model' s accuracy plateaus, with only a slight increase to 99.35% observed after 25 generations. This suggests that the genetic algorithm had identified the optimal feature set by the 20th generation, beyond which further refinement offered minimal gains. The Overlapping Behavior (OBE) observed between the 20th and 25th generations highlights this diminishing return. After the 20th generation, the performance metrics stabilized, suggesting that the model had sufficiently optimized its features and further iterations did not lead to substantial improvements in accuracy. This finding is consistent with the idea that feature selection algorithms, once they converge to an optimal feature set, experience a point of diminishing returns where additional iterations no longer significantly improve performance.

Precision, a key metric for minimizing false positives, improved from 96.50% in the non-optimized model to 99.35% after 25 generations. This marked improvement demonstrates the model' s increased ability to correctly identify actual intrusions while avoiding false alarms. Precision is crucial for the practical deployment of intrusion detection systems, as minimizing false positives reduces the operational overhead associated with unnecessary alerts. However, the overlap in performance between the 20th and 25th generations indicates that the improvement in precision is minimal after 20 generations. The precision metric, much like accuracy, demonstrates a plateau effect after 20 generations, where further optimization leads to negligible gains. This highlights the efficiency of the genetic algorithm in reaching an optimal set of features early in the optimization process.

Recall, which reflects the model' s ability to detect true positive intrusions, showed an improvement from 94.75% in the non-optimized state to 98.50% after 25 generations. This enhancement is significant because recall measures how well the model identifies actual intrusions, which is crucial for ensuring that security breaches are detected and mitigated in a timely manner. An increase in recall reduces the risk of false negatives, where intrusions go undetected by the system. The recall improvement, however, also follows a similar trend to that of precision and accuracy, with diminishing gains observed after the 20th generation. The increase from 98.25% at the 20th generation to 98.50% at the 25th generation is marginal, reinforcing the notion of OBE—that after the 20th generation, further optimization contributes only

minimally to the model' s sensitivity in detecting intrusions.

The F1 score, a harmonic mean of precision and recall, improved from 95.60% in the non-optimized model to 98.92% after 25 generations. This improvement is indicative of a balanced performance in both detecting intrusions and avoiding false alarms. A high F1 score suggests that the model is not only accurate but also reliable in maintaining a good trade-off between precision and recall, both of which are vital for an effective intrusion detection system. The F1 score follows a similar pattern to the other metrics, showing substantial improvement up to the 20th generation and only marginal increases thereafter. The minimal gain in F1 score from 20 to 25 generations further supports the observation of OBE—the optimization process reached a point where additional feature selection refinements offered diminishing returns in terms of improving the model' s overall performance.

The optimization results in this study highlight the significant improvements achieved in the MLP model' s performance with feature selection via a genetic algorithm. The key performance metrics—accuracy, precision, recall, and F1 score—all improved notably, particularly after the initial generations of optimization. However, the results also demonstrate the point at which the optimization process reached diminishing returns. The Overlapping Behavior or Sever (OBE) between the 20th and 25th generations indicates that after a certain point, the genetic algorithms feature selection process was less effective in further improving the model' s performance. This suggests that the optimal feature set was reached around the 20th generation, and further generations only provided incremental improvements in performance. This finding emphasizes the importance of recognizing when the optimization process has converged, thus saving both computational resources and time. The results also underscore the value of a balanced performance across metrics. By simultaneously improving precision and recall, the model demonstrated its ability to accurately classify network traffic while minimizing both false positives and false negatives. This balance is critical in the deployment of practical intrusion detection systems, where both false alarms and missed detections can have significant operational consequences.

V. FUTURE WORKS

Future work can explore alternative feature selection methods like PSO or ACO, integrate deep learning models for improved accuracy, and test real-time performance on large datasets. Additionally, evaluating resilience against adversarial attacks and optimizing computational efficiency will enhance practical deployment in NID systems.

CONCLUSION

This study demonstrated the effectiveness of using a genetic algorithm (GA) for feature selection in enhancing the performance of an MLP-based Network Intrusion Detection (NID) system. The results showed significant improvements in accuracy, precision, recall, and F1 score, with peak performance achieved at the 20th generation (99.23% accuracy). Beyond this point, minimal gains indicated the presence of Overlapping Behavior (OBE), suggesting an optimal feature set had been reached. These findings highlight the importance of feature selection in improving detection accuracy while optimizing computational efficiency.

ACKNOWLEDGMENT

The dataset used in this study was obtained from Kaggle, providing a diverse and well-structured benchmark for evaluating our model. The authors declare no conflicts of interest related to this research. All experiments and analyses were conducted with integrity and transparency to ensure the reliability of the findings.

REFERENCES

[1] 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. https://doi.org/10.1109/ACCESS.2018.2841987

[2] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access, 7*, 41525-41550. https://doi.org/10.1109/ACCESS.2019.2895334

[3] Aljawarneh, S., Aldwairi, M., & Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science, 25*, 152-160. https://doi.org/10.1016/j.jocs.2017.03.006

[4] Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems. *IEEE Military Communications and Information Systems Conference (MilCIS)*, 1-6. https://doi.org/10.1109/MilCIS.2015.7348942

[5] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets, and challenges. *Cybersecurity, 2*(1), 1-22. https://doi.org/10.1186/s42400-019-0038-7

[6] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSP*, 108-116. https://doi.org/10.5220/0006639801080116

[7] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 1-6. https://doi.org/10.1109/CISDA.2009.5356528

[8] Afolabi, A. S., & Akinola, O. A. (2024). Network intrusion detection using knapsack optimization, mutual information gain, and machine learning. *Security and Communication Networks, 2024*, Article 7302909. https://doi.org/10.1155/2024/7302909

[9] Talukder, M. A., Islam, M. M., Uddin, M. A., Hasan, K. F., Sharmin, S., Alyami, S. A., & Moni, M. A. (2024). Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. *Journal of Big Data, 11*(33). https://doi.org/10.1186/s40537-024-00607-3

[10] Alars, E. S. A., & Kurnaz, S. (2024). Enhancing network intrusion detection systems with combined network and host traffic features using deep learning: Deep learning and IoT perspective. *Discover Computing, 27*(39). https://doi.org/10.1007/s10791-024-09480-3

[11] Nwobodo, L. O., Chibueze, K. I., & Ezigbo, L. I. (2024). Hybrid machine learning-based framework for effective network intrusion detection. *EJSIT, 4*(6). Retrieved from https://www.ejsit.com

[12] Chindove, H., & Brown, D. (2021). Adaptive machine learning-based network intrusion detection. *Proceedings of the International Conference on Artificial Intelligence and its Applications*, 1-6. https://doi.org/10.1145/3487923.3487938

[13] Yakubu, Y. A., Musa, K. I., & Muazu, U. (2024). Software defined-network intrusion detection model using stacked ensemble techniques of machine learning. *Anchor University Journal of Science and Technology, 4*(1). https://doi.org/10.1234/aujst.2024.00001

[14] Ayo, F. E., Folorunso, S. O., Abayomi-Alli, A. A., Adekunle, A. O., & Awotunde, J. B. (2020). Network intrusion detection based on deep learning model optimized with rule-based hybrid feature selection. *Information Security Journal: A Global Perspective, 29*(6), 267-283. https://doi.org/10.1080/19393555.2020.1767240

[15] Reddy, B. R., Pradhan, S. R., Sathwik, G., & Mihika, G. (2024). Network Intrusion Detection using Machine Learning. *International Journal of Engineering Innovations and Management Strategies, 1*(4), 1.

[16] Hacılar, H., Aydın, Z., & Güngör, V. Ç. (2024). Network intrusion detection based on machine learning strategies: Performance comparisons on imbalanced wired, wireless, and software-defined networking (SDN) network traffics. *Turkish Journal of Electrical Engineering and Computer Sciences, 32*(4). https://doi.org/10.55730/1300-0632.4091

[17] Clottey, R. N., Yaokumah, W., & Appati, J. K. (2021). Modelling and evaluation of network

intrusion detection systems using machine learning techniques. *International Journal of Intelligent Information Technologies*, *17*(4), 19. https://doi.org/10.4018/IJIIT.289971

[18] Mol, P. R., & Mary, C. I. (2021). Classification of network intrusion attacks using machine learning and deep learning. *Annals of the Romanian Society for Cell Biology*, *25*(2), 1927– 1943. Retrieved from http://annalsofrscb.ro/index.php/journal/article/view/1137

[19] Telang, S., & Ranawat, R. (2024). Enhancing network security with deep learning-based intrusion detection systems. *Journal of Computational Analysis and Applications*, 33(7).

[20] Ahmed, A. A., Aliyu, A. A., Ibrahim, M., Abdulkadir, S., Ahmad, M. A., Tanko, S. A., & Umaru, I. A. (2024). Enhancing network security through integrated deep learning architectures and attention mechanisms. *FUDMA Journal of Sciences*, *8*(6), 3010. https://doi.org/10.33003/fjs-2024-0806-3010

[21] Zhou, Q., & Shi, C. (2024). A network intrusion detection method for various information systems based on federated and deep learning. *International Journal on Semantic Web and Information Systems*, *20*(1), 28. https://doi.org/10.4018/IJSWIS.335495

[22] Ghadermazi, J., Shah, A., & Bastian, N. D. (2025). Towards real-time network intrusion detection with image-based sequential packets representation. *IEEE Transactions on Big Data*, *11*, 157– 173. https://doi.org/10.1109/TBDATA.2024.3403394

[23] Hammad, M., Hewahi, N., & Elmedany, W. (2023). Enhancing network intrusion recovery in SDN with machine learning: An innovative approach. *Arab Journal of Basic and Applied Sciences*, *30*(1), 561– 572. https://doi.org/10.1080/25765299.2023.2261219

[24] Al Lail, M., Garcia, A., & Olivo, S. (2023). Machine learning for network intrusion detection—A comparative study. *Future Internet*, *15*(7), 243. https://doi.org/10.3390/fi15070243

[25] Jaradat, A. S., Barhoush, M. M., & Bani Easa, R. S. (2023). Network intrusion detection system: Machine learning approach. *Indonesian Journal of Electrical Engineering and Computer Science*, *25*(2), 1151-1158. http://doi.org/10.11591/ijeecs.v25.i2.pp1151-1158

[26] Chimphlee, S., & Chimphlee, W. (2023). Machine learning to improve the performance of anomaly-based network intrusion detection in big data. *Indonesian Journal of Electrical Engineering and Computer Science*, 30(2), 1106– 1119. https://doi.org/10.11591/ijeecs.v30.i2.pp1106-1119

[27] Shivaram, M., & Sravanthi, K. (2023). Network intrusion detection using supervised machine learning. *International Journal of Novel Research and Development*, 2312005.