International Journal of Management and Data Analytics (IJMADA)

Int. J. Management and Data Analytics, Vol. 5 (1), 124-131
ISSN: 2816-9395
Journal Homepage: http://ijmada.com
DOI: https://doi.org/10.5281/zenodo.15331043

Performance Evaluation of Neighbor-Based Learning Models for Network Intrusion Detection System

Ngoc Thien Nguyen
University of Economics Ho Chi Minh City, Vietnam.

Received: March, 2025, Published: May, 2025

ARTICLE INFO
Keywords:
Intrusion Detection;
Neighbor-Based Learning;
k-Nearest Neighbors;
Nearest Centroid Classifier;
Radius Nearest Classifier

© 2025 Ngoc Thien Nguyen. This open-access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license, making research freely available to the public and supporting a greater global exchange of knowledge and human experiments.

Cite: Ngoc Thien Nguyen. (2025). Performance Evaluation of Neighbor-Based Learning Models for Network Intrusion Detection System. Int. Journal of Management and Data Analytics, 5 (1), 124-131. https://doi.org/10.5281/zenodo.15331043

ABSTRACT

The rapid advancement of computer networks offers substantial benefits but also introduces growing cybersecurity risks. Cyber-attacks are increasing in both frequency and sophistication, demanding continuous improvements in security systems. Consequently, intrusion detection technologies are being actively researched and optimized to enhance threat detection and prevention capabilities. In particular, Machine Learning (ML) and Artificial Intelligence (AI) have become important tools in enhancing cyber-attack detection capabilities. One of the effective approaches in this field is Neighbor-Based Learning models, a group of ML algorithms that have proven useful in identifying malicious behavior in network traffic. In this study, we apply data preprocessing techniques such as Random Under Sampling to balance the dataset and Robust Scaler to reduce the effect of outliers, thereby improving model performance. Three classification algorithms are implemented, including k-Nearest Neighbors (k-NN), Radius Nearest Classifier (RNC), and Nearest Centroid Classifier (NCC), with the goal of evaluating their effectiveness in detecting cyber-attacks. Experimental results show that after hyperparameter tuning, the RNC model achieves the highest performance with an accuracy of 83.59%, demonstrating strong potential for building efficient and reliable intrusion detection systems. The findings not only contribute to improving detection performance but also suggest new research directions, particularly the integration of Neighbor-Based Learning with Deep Learning or Ensemble techniques to further enhance adaptability in real-time cybersecurity environments.

1. INTRODUCTION

The rapid evolution of computer networks has introduced significant advantages while simultaneously escalating cybersecurity risks. Cyber-attacks are not only increasing in frequency but also advancing in sophistication, employing stealthier techniques that evade conventional detection mechanisms. These threats pose severe risks to organizational infrastructure and individual data integrity. To combat these challenges, enterprises are increasingly deploying Intrusion Detection Systems (IDS), which employ real-time network traffic analysis and anomaly detection to identify and mitigate unauthorized access attempts proactively.

Traditional IDS approaches—especially signature-based systems—are effective against known threats but often fail to detect new or evolving attack patterns (Bace, 2000; Gascon et al., 2011). To address this limitation, Machine Learning (ML) and Artificial Intelligence (AI) have emerged as promising tools for enhancing intrusion detection by enabling systems to learn from data and adapt to unseen threats (Ahmad et al., 2021; Vinayakumar et al., 2019). Among ML techniques, distance-based classifiers provide a lightweight and interpretable alternative to complex black-box models.

This study focuses on evaluating the performance of Neighbor-Based Learning models for Network Intrusion Detection Systems (NIDS). Specifically, we investigate the effectiveness of three supervised classifiers: k-Nearest Neighbors (k-NN), Radius Nearest Classifier (RNC), and Nearest Centroid Classifier (NCC). While these models are often considered simple, we hypothesize that, with appropriate data preprocessing and hyperparameter tuning, they can deliver strong results in intrusion detection tasks.

To ensure fair and optimal evaluation, we apply a systematic preprocessing pipeline involving Random Under Sampling to address class imbalance and Robust Scaler to mitigate the effect of outliers. We then conduct a thorough comparison of the three models before and after hyperparameter tuning to assess their learning capacity, adaptability, and generalization performance.

Our key contribution lies in demonstrating that even relatively simple models like RNC, when properly tuned, can outperform other classifiers in both accuracy and precision. This suggests that Neighbor-Based models

remain a viable option in modern NIDS, especially in scenarios where computational resources are limited.

The rest of this paper is structured as follows: Section 2 reviews related work on ML-based intrusion detection. Section 3 details the methodology and model configurations. Section 4 discusses experimental results and insights. Section 5 concludes with final remarks and future research directions.

2. LITERATURE REVIEW

A. *The Evolution of ML in NIDS*

In recent years, many studies have highlighted the promising capabilities of ML techniques in the field of network intrusion detection. Kim and Pak (2023) emphasized that advances in ML have promoted the development of integrated feature-based systems that significantly enhance detection accuracy. Their findings are consistent with those of Alhayali et al. (2021), who emphasized that intelligent NIDS, driven by ML algorithms, offer effective solutions to overcome the limitations of traditional intrusion detection methods, especially related to low detection rates (Alhayali et al., 2021). The general consensus is that ML not only improves the performance of NIDS but also provides adaptive solutions that are capable of learning from evolving attack patterns (Ahmad et al., 2020).

B. *Classification and Prediction in Intrusion Detection*

The classification capabilities of ML enable efficient classification of network traffic into normal and anomalous behaviors, facilitating the identification of potential intrusions. ML algorithms such as Support Vector Machines (SVMs), Random Forests, and Neural Networks have gained attention due to their reliability in extracting important features from complex data sets (Vanin et al., 2022; Ahmad et al., 2020). Furthermore, hierarchical organization of data in Software-Defined Networks (SDNs) enables centralized control and monitoring, which significantly enhances decision making in ML-enabled NIDS (Kumar & Alqahtani, 2023). Ahmad et al. (2020) assert that ML techniques play a vital role in reducing false positives while enhancing the overall detection capability of NIDS. Using a variety of classifiers, the researchers reported significant improvements in detection rates for different types of intrusions, thus addressing the challenges posed by increasingly sophisticated cyber threats.

C. *Using Datasets and Challenges*

The effectiveness of ML-based NIDS largely depends on the quality and representativeness of the datasets used to train these systems. Maseer et al. (2021) advocate benchmarking ML algorithms using standardized datasets to comprehensively evaluate their performance. However, many traditional datasets may not cover the entire spectrum of modern attack vectors, which can hinder the reliability and adaptability of NIDS (Daset al., 2022). The introduction of new datasets, such as UNR-IDD, aims to provide more comprehensive representations of network traffic, thereby improving model training (Das et al., 2022). However, challenges related to dataset bias and scalability of these datasets remain significant barriers in the quest for robust NIDS.

D. *Ensemble Learning and Feature Selection*

The integration of Ensemble Learning techniques is emerging as a promising strategy to improve the accuracy and reliability of ML-based NIDS. Research has demonstrated that the application of ensemble methods, such as the use of multiple classifiers, can effectively reduce variance and increase the performance of IDS (Ibraheem, 2022; Hammood & Sadiq, 2023). Additionally, effective feature selection is crucial, as the multidimensionality of input features can significantly affect the performance of the model. Ling and Wu (2019) advocate feature selection strategies that emphasize techniques that reduce complexity while preserving essential information relevant to intrusion detection. The combination of ensemble learning and informed feature selection has resulted in notable advances in intrusion detection accuracy and efficiency.

E. *Deep Learning Approaches*

Deep learning (DL), a subset of ML, has recently been applied to NIDS with considerable success. Vinayakumar et al. (2019) describe how DL models can capture complex patterns in network traffic that conventional ML techniques may miss. With the application of deep neural networks (DNNs), researchers have reported breakthroughs in detecting previously unseen types of intrusions due to their ability to learn hierarchical representations of data. Furthermore, the dynamic learning capabilities of DL models allow them to adapt over time, improving their effectiveness against evolving threats (Lánský et al., 2021). However, despite the advantages of DL, the requirement for large amounts of labeled data and computational resources remains a challenge (Ahmad et al., 2020).

F. *False Alarm Handling and Detection Accuracy*

One of the pressing challenges in NIDS is to minimize the false alarm rate without sacrificing detection accuracy. Zeng et al. (2021) have shown that while rule-based systems tend to generate too many false alarms, ML methods can achieve higher detection accuracy by learning from historical attack patterns. Furthermore, the integration of causal ML techniques, although relatively new, shows promise in improving the understanding and reasoning behind detected intrusions, thereby improving the accuracy of NIDS (Zeng et al., 2021). Continuous iterative training and updating of ML models to respond to

new threats can further reduce the occurrence of false alarms (Ahmad et al., 2020).

G. Real-time Intrusion Detection

The need for real-time processing capabilities in intrusion detection systems has driven innovations in ML methodologies. Abdulhameed et al. (2024) proposed optimized models that can handle high network traffic loads while maintaining real-time detection capabilities. In situations where traffic volumes are high, detection speed becomes critical in minimizing potential damage in intrusion attempts. Research on concurrency and distributed processing has shown that by leveraging modern computing frameworks, NIDS can process data streams at unprecedented speeds, effectively combining speed with enhanced accuracy (Güney, 2023). However, ensuring that these systems remain scalable and maintain performance when demand is high remains a constant challenge.

3. METHODOLOGY

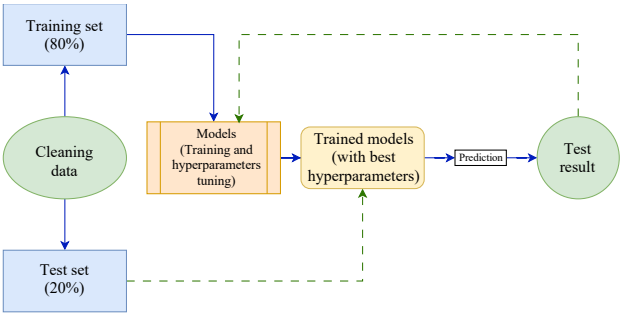

Figure 1. ML workflow

Fig. 1 shows the ML workflow we will use, which consists of six closely related cases ordered from left to right. The workflow has three main stages: cleaning and splitting data, model training and hyperparameter tuning, and evaluation.

The dataset is collected from Kaggle, including 9537 records, including 11 columns such as IP reputation score, failed logins, browser type, unusual time access... and a binary attack detection label.

A. Cleaning and Splitting Data

This process consists of four steps: Removing unnecessary columns, encoding categorical data, feature scaling data, and balance data. Proper data preprocessing can significantly improve model accuracy, robustness, and interpretability (Wong, Leung, & Ling, 2013; Karakatič, 2020).

First, unnecessary columns will be removed from the dataset to optimize the training process of the models. When unnecessary columns are discarded, the potential for misinterpretation and error in data processing diminishes, leading to more reliable analytical outputs. Study highlights that overfitting is a typical problem that arises when excessively high-dimensional datasets are used in ML tasks, ultimately leading to low performance outcomes due to noise in irrelevant features (Jeon & Oh, 2020).

The second step is to encode the data into categories. In this step, columns in the data that are not in numeric form are converted to their corresponding numeric columns. This is a very important step because most ML models cannot take a non-numeric variable as an input value. Label Encoding will be the encoding method that the researcher uses, for example the table below for the browser type column:

TABLE I. LABEL ENCODING SAMPLE

Original values	Encoded values
Chrome	0
Edge	1
Firefox	2
Safari	3
Unknown	4

Tab. 1 shows how Label Encoding works, the algorithm will rearrange the non-numeric data in alphabetical order, and will encode in order from 0 to *n-1* (*n* is the total number of unique values in the column).

The third step is data balancing. Without this step, this imbalance can lead to biased predictions, where models may favor the majority class and ignore the minority class, ultimately affecting the performance of classification algorithms (Hayaty, Muthmainah, & Ghufran, 2020; Cenggoro et al., 2018; Gupta, Jhunjhunwalla, Bhardwaj, & Shukla, 2020). As the exploration of this domain progresses, it becomes increasingly clear that a multifaceted approach integrating sampling strategies, cost-sensitive learning, and proper performance metrics will yield the best results in overcoming the challenges posed by imbalanced data (Ruzgar & Chua, 2020; Khushi et al., 2021; Ahmadzadeh & Angryk, 2022).

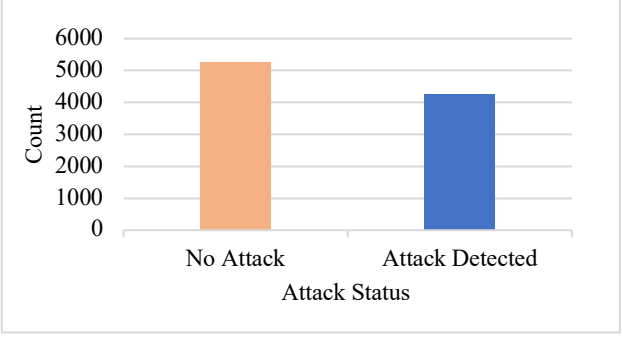

Figure 2. Distribution of Attack Status in the dataset

Fig. 2 shows the distribution between the two output variables of the data. It is easy to see that the data is unbalanced. To rebalance the data, there are many ways to do it, but in this case, a random reduction of the majority class instances will be applied to match the number of minority class instances. This method, known as Random Under Sampling, is simple, easy to implement, and widely used.

The fourth step in data cleaning and processing is feature scaling. But before scaling, it is also important to handle outliers. To streamline the process for both of these tasks, the Robust Scaler method is quite suitable. The Robust Scaler method standardizes the data by centering the data around the median, then scaling it by the range between the first quartile (Q_1) and the third quartile (Q_3), thereby significantly reducing the impact of outliers on the model performance (Sari, Stiawan, Afifah, Idris, & Budiarto, 2024; Begum, Rahman, & Omar Faruk, 2024; Al-Sayed, Khayyat, & Zamzami, 2023). The formula to calculate the transformed values for Robust Scaler is:

$$x' = \frac{x - Q_1(X)}{Q_3(X) - Q_1(X)} \quad (1)$$

Where x' is original value, X is the numerical attribute, and $Q_3(X) - Q_1(X)$ is the interquartile range.

The final step is to split the data into a training set and a test set. The training set will make up 80% of the data and the test set will make up the remaining 20%. This split is done completely randomly, effectively ensuring that the trained model can generalize far beyond the training set, thus improving its applicability in real-world scenarios (Rumala, 2023; Hou, Lo, Marks, Hwang, & Grimm, 2024).

B. ML Models and Hyperparameters

1) k-Nearest Neighbors (k-NN): k-NN is based on the idea that the samples closest to the target sample y' as the output label will provide useful label information (Kramer, 2013).

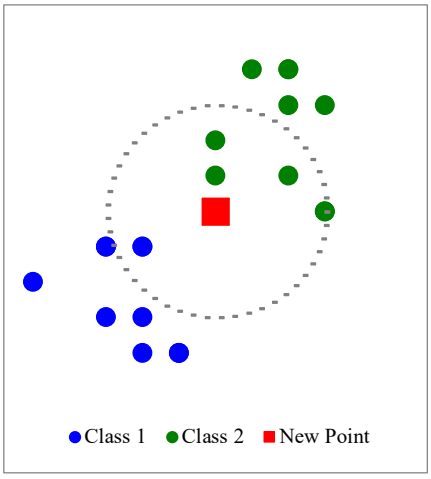

Figure 3. Example of k-NN with k = 5.

In Fig. 3, the new data point is predicted to belong to Class 2 because in the 5 closest points to the new data point, there are 4 points belonging to Class 2 and only 1 point belonging to Class 1. To calculate the distance between the points, here we will use 2 distance functions: Manhattan and Euclidean.

The Manhattan distance (d_M) between two points $A(x_a, y_a)$ and $B(x_b, y_b)$ has the following formula:

$$d_M(A, B) = |x_a - x_b| + |y_a - y_b| \quad (2)$$

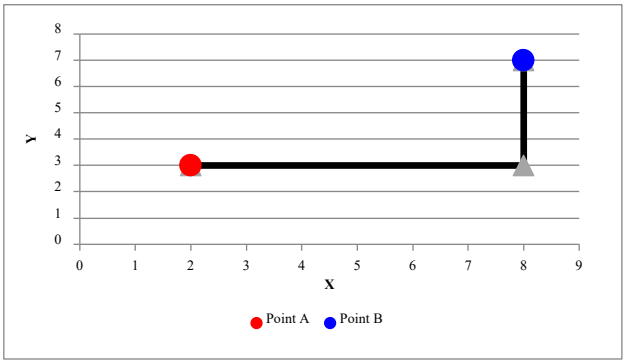

Figure 4. Example of Manhattan distance.

In Fig. 4, Point A has coordinates (2,3), point B has coordinates (8,7). So, the distance d_M will be 10, and the path is as shown. Generalizing formula (*2*) for multidimensional data, the distance formula between 2 points $A(a_1, a_2, \ldots, a_n)$ and $B(b_1, b_2, \ldots, b_n)$ is:

$$d_M(A, B) = \sum_{i=1}^{n} |a_i - b_i| \quad (3)$$

Where n is the dimension of the data, also known as the number of features of the data set.

Similar to Manhattan distance formula in (*3*), Euclidean distance (d_E) is defined as follows:

$$d_E(A,B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \quad (4)$$

k and distance functions are the two most important hyperparameters of the k-NN algorithm. Effectively tuning the distance metric can significantly influence model accuracy depending on the dataset used. Appropriate distance functions help distinguish between classes more effectively, especially in high-dimensional spaces where feature scaling and standardization become critical (Assegie, Suresh, Purushothaman, Ganesan, & Kumar, 2023).

2) Radius Nearest Classifier (RNC): RNC has a similar idea to k-NN, but instead of using the k closest data points for classification, RNC will create a circle with the center being the coordinates of the data point y' that needs to be classified, the radius is a real number r which is also the hyperparameter of the model. RNC predicts by counting the largest number of classes in the circle and concluding that point y' is that class.

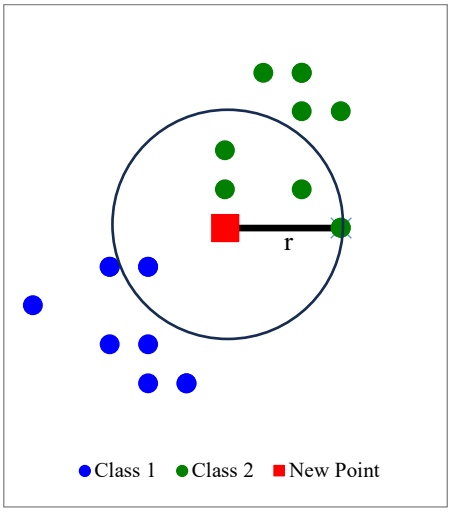

Figure 5. Example of RNC

Fig. 5 illustrates how the RNC model works. In this example case, there are 4 points belonging to Class 2 in the circle, only 1 point belonging to Class 1 is inside, so the new data point will be predicted as class 2. To determine whether point y_i is inside the circle, we have the formula:

$$d(y', y_i) \le r \quad (5)$$

Where d is the distance function, y_i is a point in the dataset.

3) Nearest Centroid Classifier (NCC): This approach uses "denoised" versions of the centroids as prototypes for each class (Tibshirani, Hastie, Narasimhan, & Chu, 2002).

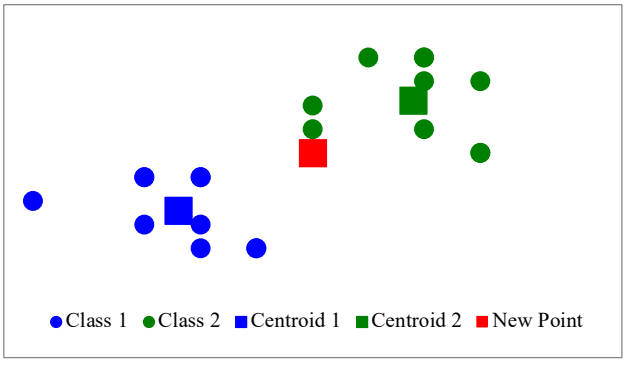

Figure 6. Example of RNC' classification

Fig. 6 illustrates the NCC algorithm for finding the centroids. First, we compute the centroids for each cluster of class C using the formula:

$$\mu_C = \frac{1}{N_C}\sum_{x_i \in C} x_i \quad (6)$$

Where μ_C is the centroid of class C, N_C is the number of points in class C, and x_i is the data points belonging to class C.

Then, the distance between the point to be classified y' and the centroid of each class μ_C will be calculated. The distance is calculated according to formula (*3*) or (*4*) above. The classification result will be the class with the smallest distance value. In NCC, the most common hyperparameter is the distance function, making it a relatively low-complexity model in Neighbor-Based learning methods.

C. Evaluation methods (Metrics)

In this paper, we will use the four most popular evaluation methods for classification models: Accuracy, Recall, Precision and F1-Score (F_1). These four methods are calculated based on the confusion matrix as shown in the table below:

TABLE II. CONFUSION MATRIX WITH 2 TYPES OUTPUT VALUES

	Predicted value	
Actual value	***Positive***	***Negative***
Positive	**True Positive (TP)**	False Negative (FN)
Negative	False Positive (FP)	**True Negative (TN)**

These four evaluation methods have formulas based on the confusion matrix, each method will have its own advantages and disadvantages:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (10)$$

If the primary concern is overall accuracy, then Accuracy is the right choice due to its simplicity and ease of interpretation. However, if the goal is to evaluate the performance of the model on specific data classes, Precision, Recall, and F1-Score are more appropriate metrics. Precision reflects the proportion of attacks that are correctly predicted, which helps reduce false alarms. Recall measures the ability to detect all attacks, reducing the risk of missing potential threats. F1-Score provides a balance between Precision and Recall, providing a more comprehensive view of the study.

4. Analysis and Discussion

A. Before Hyperparameter tuning

TABLE III. Result before Hyperparameter tuning

	Accuracy	Precision	Recall	F1
k-NN	0.8165	0.8257	0.8165	0.8152
RNC	0.6970	0.8059	0.6970	0.6673
NC	0.7104	0.7110	0.7104	0.7102

Based on the performance comparison of the three classification models k-NN, RNC and NC, it is clear that k-NN is the best performing model in this case. With the highest accuracy of 81.65%, along with the highest Precision, Recall and F1 score, the model maintains a good balance across all metrics and demonstrates stable classification performance.

RNC, despite having a relatively high Precision (80.59%), showed significantly lower Accuracy and Recall (69.70%). This showed that while the model makes highly accurate positive predictions, it tends to miss many other cases, resulting in a lower overall performance.

Meanwhile, NC achieves slightly higher Accuracy than RNC (71.04%) and maintains a fairly balanced Precision, Recall and F1 score. However, it is still less efficient than k-NN.

In this case, if the goal is to have a model that outperforms all key metrics, then k-NN is the optimal choice. On the other hand, NC serves as an intermediate choice, providing a simpler model while maintaining a balance between different metrics.

B. After Hyperparameter tuning

TABLE IV. Result after Hyperparameter tuning

	Accuracy	Precision	Recall	F1
k-NN	0.8247	0.8336	0.8247	0.8236
RNC	0.8359	0.8745	0.8359	0.8315
NC	0.7104	0.711	0.7104	0.7102

After hyperparameter tuning, the RNC showed a significant performance boost, surpassing the other models. Its accuracy improved from 69.70% to 83.59%, while Precision increased from 80.59% to 87.45%. This dramatic enhancement suggests that RNC is highly sensitive to the radius parameter, which when appropriately chosen, allows the model to capture the optimal density threshold for class separation.

In contrast, k-NN exhibited only marginal improvements, indicating that its performance was already near-optimal or less affected by the selected distance metrics. The NCC remained mostly unchanged, reinforcing the notion that its simplicity limits adaptability to complex decision boundaries even with tuning.

These results emphasize that careful hyperparameter selection, particularly for RNC, can unlock significant performance gains in intrusion detection without the need for more computationally expensive models.

5. Conclusion

Information security and safety are urgent needs in the context of cyber-attacks increasing in number and sophistication. To minimize risks and improve intrusion detection capabilities, we have studied in detail data preprocessing methods, to ensure the most optimal input for machine learning algorithms. At the same time, we apply Neighbor-Based Learning models, a group of algorithms based on the distance between data points for classification and anomaly detection. In addition, we also propose hyperparameter optimization methods, which help improve model performance, minimize errors and increase the ability to generalize on different data sets.

With this proposal, the deployed intrusion detection system will support developers in enhancing the security of web applications, especially in cases where they miss or have not detected potential security vulnerabilities. The

system can serve as an additional layer of protection, helping to mitigate the risk of attack.

In the future, we will continue to expand the research by testing on larger, more diverse datasets to ensure that the model can effectively detect various types of cyber-attacks. At the same time, we will combine Neighbor-Based Learning with DL techniques or Ensemble Learning methods to increase accuracy and adaptability in real-world environments. In addition, optimizing computational efficiency is also an important direction, to ensure that the system can detect intrusions in near real-time without affecting the performance of the network system.

With these improvements, we expect that IDS will become a powerful tool, helping businesses and organizations improve their defenses against increasingly complex cyber threats.

REFERENCES

[1] Abdulhameed, A. A., Alazawi, S. A. H., & Hassan, G. M. (2024). An optimized model for network intrusion detection in the network operating system environment. Mesopotamian Journal of CyberSecurity, 4 (3), 75–85.

[2] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32 (1), e4150.

[3] Ahmadzadeh, A., & Angryk, R. A. (2022). Measuring class-imbalance sensitivity of deterministic performance evaluation metrics. In 2022 ieee international conference on image processing (icip) (pp. 51–55).

[4] Alhayali, R. A. I., Aljanabi, M., Ali, A. H., Mohammed, M. A., & Sutikno, T. (2021). Optimized machine learning algorithm for intrusion detection. Indonesian Journal of Electrical Engineering and Computer Science, 24 (1), 590–599.

[5] Al-Sayed, A., Khayyat, M. M., & Zamzami, N. (2023). Predicting heart disease using collaborative clustering and ensemble learning techniques. Applied Sciences, 13 (24), 13278.

[6] Assegie, T. A., Suresh, T., Purushothaman, R., Ganesan, S., & Kumar, N. K. (2023). Early prediction of gestational diabetes with parameter-tuned k-nearest neighbor classifier. Journal of Robotics and Control (JRC), 4 (4), 452–457.

[7] Bace, R. G. (2000). Intrusion detection. Sams Publishing.

[8] Begum, N., Rahman, M. M., & Omar Faruk, M. (2024). Machine learning prediction of nutritional status among pregnant women in bangladesh: Evidence from bangladesh demographic and health survey 2017–18. Plos one, 19 (5), e0304389.

[9] Cenggoro, T. W., et al. (2018). Deep learning for imbalance data classification using class expert generative adversarial network. Procedia Computer Science, 135 , 60–67.

[10] Das, T., Abu Hamdan, O., Shukla, R., Sengupta, S., & Arslan, E. (2022). Unridd: intrusion detection dataset using network port statistics. techrxiv. Preprint. https://doi. org/10.36227/techrxiv, 198 (77311), v3.

[11] Gascon, H., Orfila, A., & Blasco, J. (2011). Analysis of update delays in signature-based network intrusion detection systems. Computers & Security, 30 (8), 613–624.

[12] Güney, H. (2023). Awc-nids: Attack-wise customized network intrusion detection system using machine learning, concurrency, and distributed systems. Concurrency and Computation: Practice and Experience, 35 (26), e7832.

[13] Gupta, S., Jhunjhunwalla, M., Bhardwaj, A., & Shukla, D. (2020). Data imbalance in landslide susceptibility zonation: Under-sampling for class-imbalance learning. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 42 , 51–57.

[14] Hammood, B. A. K., & Sadiq, A. T. (2023). Ensemble machine learning approach for iot intrusion detection systems. Iraqi Journal for Computers and Informatics, 49 (2), 93–99.

[15] Hayaty, M., Muthmainah, S., & Ghufran, S. M. (2020). Random and synthetic over-sampling approach to resolve data imbalance in classification. International Journal of Artificial Intelligence Research, 4 (2), 86–94.

[16] Hou, R., Lo, J. Y., Marks, J. R., Hwang, E. S., & Grimm, L. J. (2024). Classification performance bias between training and test sets in a limited mammography dataset. Plos one, 19 (2), e0282402.

[17] Ibraheem, I. K. (2022). Enhancing intrusion detection systems using ensemble machine learning techniques. Data and Metadata, 1, 33–33.

[18] Jeon, H., & Oh, S. (2020). Hybrid-recursive feature elimination for efficient feature selection. Applied Sciences, 10 (9), 3211.

[19] Karakatič, S. (2020). Evopreprocess—data preprocessing framework with nature-inspired optimization algorithms. Mathematics, 8 (6), 900.

[20] Khushi, M., Shaukat, K., Alam, T. M., Hameed, I. A., Uddin, S., Luo, S., . . . Reyes, M. C. (2021). A comparative performance analysis of data resampling methods on imbalance medical data. IEEE Access, 9 , 109960–109975.

[21] Kim, T., & Pak, W. (2023). Integrated feature-based network intrusion detection system using incremental feature generation. Electronics, 12 (7), 1657.

[22] Kramer, O. (2013). K-nearest neighbors. In Dimensionality reduction with unsupervised nearest neighbors (pp. 13–23). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-38652-7$_2$ doi: 10.1007/978-3642-38652-7$_2$

[23] Kumar, G., & Alqahtani, H. (2023). Machine learning techniques for intrusion detection systems in sdn-recent advances, challenges and future directions. CMES-Computer Modeling in Engineering & Sciences, 134 (1).

[24] Ling, J., & Wu, C. (2019). Feature selection and deep learning based approach for network intrusion detection. In 3rd international conference on mechatronics engineering and information technology (icmeit 2019) (pp.764–770).

[25] Maseer, Z. K., Yusof, R., Bahaman, N., Mostafa, S. A., & Foozy, C. F. M. (2021). Benchmarking of machine learning for anomaly based intrusion detection systems in the cicids2017 dataset. IEEE access, 9 , 22351–22370.

[26] Rumala, D. J. (2023). How you split matters: data leakage and subject characteristics studies in longitudinal brain mri analysis. In Workshop on clinical image-based procedures (pp. 235–245).

[27] Ruzgar, N. S., & Chua, C. (2020). Data level approach for multiclass imbalance financial data. WSEAS Transactions on Computers, ISSN/E-ISSN , 1109–2750.

[28] Sari, R. E., Stiawan, D., Afifah, N., Idris, M. Y., & Budiarto, R. (2024). Classification of darknet traffic using the adaboost classifier method. Indonesian Journal of Electrical Engineering and Informatics (IJEEI), 12 (2), 384–396.

[29] Tibshirani, R., Hastie, T., Narasimhan, B., & Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proceedings of the National Academy of Sciences, 99 (10), 6567–6572.

[30] Vanin, P., Newe, T., Dhirani, L. L., O'Connell, E., O'Shea, D., Lee, B., & Rao, M. (2022). A study of network intrusion detection systems using artificial intelligence/machine learning. Applied Sciences, 12 (22), 11752.

[31] 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.

[32] Wong, G. Y., Leung, F. H., & Ling, S.-H. (2013). A novel evolutionary preprocessing method based on over-sampling and under-sampling for imbalanced datasets. In Iecon 2013-39th annual conference of the ieee industrial electronics society (pp. 2354–2359).

[33] Zeng, Z., Peng, W., & Zhao, B. (2021). Improving the accuracy of network intrusion detection with causal machine learning. Security and Communication Networks, 2021 (1), 8986243.