International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group| 9th ICAILLMBA (ALLAM) 2025-Conference Proceedings
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.ICAILLMBA-121

Original Article

Network Security Enhancement through Machine Learning–Driven Intrusion Detection

Madhu Raghuveer[1], Puli Lakshmi[2]
[1]Assistant Professor (Guest Faculty), Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, India.
[2]Assistant Professor (Guest Faculty), Department of Information Technology and Computer Applications, Andhra University, Visakhapatnam, India.

Abstract - *With the rapid expansion of computer networks and internet-based services, protecting network infrastructures from cyberattacks has become a critical challenge. Traditional security mechanisms often fail to detect sophisticated and evolving intrusion patterns, highlighting the need for intelligent intrusion detection systems. This study addresses the problem of effective network intrusion detection by presenting a comparative analysis of instance-based and numerical machine learning techniques for a Network Intrusion Detection System (NIDS). In this work, K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) models are employed to perform multiclass classification of network attacks, including Denial of Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R) attacks. The proposed framework analyzes network traffic patterns and behavioral features to identify malicious activities in real time. The models are trained and evaluated using benchmark intrusion datasets, and performance is assessed using metrics such as accuracy, precision, recall, and efficiency. Experimental results indicate that both KNN and SVM effectively detect multiple attack categories, with instance-based learning demonstrating strong detection capability. The study concludes that machine learning–based NIDS significantly enhances network security through improved accuracy, adaptability, and timely threat mitigation.*

Keywords - *Network Intrusion Detection System (Nids), Machine Learning, K-Nearest Neighbor (Knn), Support Vector Machine (Svm), Multiclass Classification, Dos Attack, Probe Attack, R2l Attack, U2r Attack, Cybersecurity.*

1. Introduction

The fast advancement of information and communication technologies has greatly expanded the dependence on computer networks, making them prime targets for cyber threats such as intrusions, unauthorized access, and malicious attacks. Traditional security mechanisms, including firewalls, access control, and antivirus software, provide basic protection but are often insufficient against sophisticated and evolving attack techniques. As a result, NIDShave become crucial instruments for continuously monitoring network traffic and identifying suspicious activities to safeguard network infrastructures.

An Intrusion Detection System is developed to detect malicious actions that compromise the security, reliability, and accessibility of network resources [1]. Intrusions may originate from external attackers or internal users attempting unauthorized access. IDS analyze network packets or system events to identify abnormal behavior and generate alerts for timely intervention. Early intrusion detection approaches relied on statistical analysis and rule-based expert systems; however, these methods struggle with scalability and high false alarm rates in modern high-speed networks.

To overcome these limitations, machine approaches have gained popularity in intrusion detection as they can identify patterns from large volumes of data. Machine learning enables IDS to identify and separate benign network traffic from harmful actions using classification, clustering, and pattern recognition methods. learning paradigms like supervised, unsupervised, semi-supervised, and reinforcement learning provide flexible solutions for modeling complex and evolving attack patterns, improving detection accuracy and adaptability.

Among various IDS types, Network Intrusion Detection Systems monitor traffic across entire network segments, while Host-based and Protocol-based IDS focus on individual devices and specific communication protocols, respectively. Despite these advancements, existing IDS still face challenges such as false positives, limited generalization to new attacks, and performance constraints. Therefore, this research aims to develop an effective intrusion detection framework that leverages computational and machine learning approaches to accurately differentiate between normal network connections and malicious intrusions.

2. Literature Survey

Almseidin et al. evaluated multiple machine learning techniques were applied to network intrusion detection using the KDD dataset, and the results showed that Random Forest to achieve the highest detection accuracy. However, the use of an outdated dataset limits applicability to modern networks. The study concludes that ensemble learning methods are effective for intrusion detection but require validation on recent datasets [2].Nguyen and Choi explored the application of data mining techniques for network intrusion detection by comparing multiple classifiers, including Decision Trees and Naive Bayes, to identify the most effective model for distinguishing normal and malicious traffic. Their evaluation demonstrated that certain data mining classifiers achieve higher detection accuracy on benchmark intrusion datasets, highlighting the potential of classification methods in enhancing IDS performance. However, the study's reliance on static datasets and limited attack types constrains its generalizability to evolving real-world network environments. The authors conclude that careful selection and evaluation of data mining models can significantly improve the effectiveness of intrusion detection systems [3].

Paliwal and Gupta proposed a genetic algorithm–based intrusion detection approach for detecting Probing, Denial-of-Service (DoS), and Remote-to-User (R2L) attacks by generating optimized classification rules from network audit data. Their method leverages evolutionary search to derive effective detection rules, improving IDS capability to recognize diverse attack types compared to static rule sets. However, the approach may face challenges related to computational overhead and scalability on large real-world network traffic due to the complexity of genetic operations. The study concludes that genetic algorithms can enhance rule generation for IDS, offering a promising alternative to traditional detection techniques [4].Tavallaee et al. analyzed the KDD CUP'99 dataset and identified major issues such as data redundancy and biased distributions that negatively impact intrusion detection evaluation. To overcome these limitations, they introduced the NSL-KDD dataset, which provides a more balanced and reliable benchmark for anomaly-based intrusion detection research [5].

Arul presented a detailed overview of classification approaches applied to network intrusion detection, examining algorithms like decision trees, SVM, neural networks, and Bayesian classifiers in terms of their detection accuracy and suitability for various attack types. The authors highlighted that no single classifier consistently outperforms others across all intrusion scenarios and that hybrid or ensemble methods can offer improved performance. They also discussed challenges related to dataset characteristics and algorithm complexity. The study concludes that careful selection and combination of classification techniques is essential for designing effective and resilient intrusion detection systems [6].

Alkasassbeh and Almseidin investigated the application of machine learning classifiers, including J48, Multi-Layer Perceptron (MLP), and Bayesian Network, for network intrusion detection using the KDD dataset, emphasizing careful data preprocessing to ensure fair evaluation. Their experiments showed that the J48 decision tree classifier achieved the highest accuracy in detecting multiple attack categories such as DoS, R2L, U2R, and Probe, highlighting the importance of classifier selection and dataset preparation. However, the study's dependence on the traditional KDD dataset limits its relevance to modern network traffic characteristics. The authors conclude that selecting suitable machine learning methods and preparing balanced datasets are crucial for effective intrusion detection system performance [7].

Almseidin et al. evaluated multiple intrusion detection algorithms, including decision trees, Naive Bayes, and SVM, evaluated on the commonly used KDD-99 dataset to assess their effectiveness in recognizing multiple forms of network intrusions. Their results indicated that certain classifiers, particularly ensemble and tree-based methods, achieved higher detection accuracy across attack categories, demonstrating the impact of algorithm selection on IDS performance. However, reliance on the KDD-99 dataset, which has known limitations like redundant records and outdated attack patterns, may reduce the relevance of findings for contemporary network environments. The study concludes that careful selection and tuning of machine learning algorithms can improve intrusion detection accuracy, but benchmarking on more representative datasets is necessary for practical deployment [8].

3. Methodology

3.1. About Dataset

The NSL-KDD dataset is a widely used benchmark dataset for evaluating network intrusion detection systems, created as a refined version of the older KDD Cup 1999 dataset to address issues like redundant and biased records. It consists of labeled records of network traffic, where each connection is described by dozens of features capturing various characteristics of the communication and is tagged either as normal or as one of several types of malicious attacks. These attacks are grouped into categories such as Denial of Service (DoS), probing activities, and different forms of unauthorized access, enabling models to learn and distinguish between benign behavior and diverse intrusions; the dataset is split into training and test subsets to allow supervised machine learning evaluation.

3.2. Proposed Model

The proposed model MKS (Multi-class Classification using KNN and SVM) presents in fig1 a supervised machine learning–based framework for network intrusion detection and attack classification. Initially, a standard network intrusion dataset is collected, which contains various network traffic features along with labeled attack categories. These categories include Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L) attacks. Prior to model training, the dataset undergoes a comprehensive preprocessing stage to ensure data quality and improve classification performance. This stage involves handling missing or inconsistent values, encoding categorical attributes into numerical representations, and normalizing numerical features to maintain uniformity across different scales.

Following preprocessing, the dataset is partitioned into training and testing subsets to enable unbiased model evaluation. The training dataset is then supplied independently to two supervised learning classifiers, namely KNN and SVM. The KNN classifier performs classification by measuring the similarity between data instances and assigning class labels based on the majority class among the nearest neighbors. In contrast, the SVM classifier constructs an optimal decision boundary that maximizes the margin between different attack classes, thereby enabling efficient separation of high-dimensional network traffic data. Both models are trained using the same dataset to ensure a fair comparative analysis.

Once training is completed, the trained KNN and SVM models are applied to the test dataset to predict the corresponding attack categories. Each model classifies network traffic into one of the predefined classes, namely DoS, Probe, U2R, or R2L. The performance of both classifiers is then evaluated using standard metrics. This evaluation enables a comparative assessment of the classifiers and helps identify the most effective model for intrusion detection. The proposed methodology demonstrates an efficient and reliable approach for detecting and classifying network intrusions, thereby enhancing the security and robustness of network systems

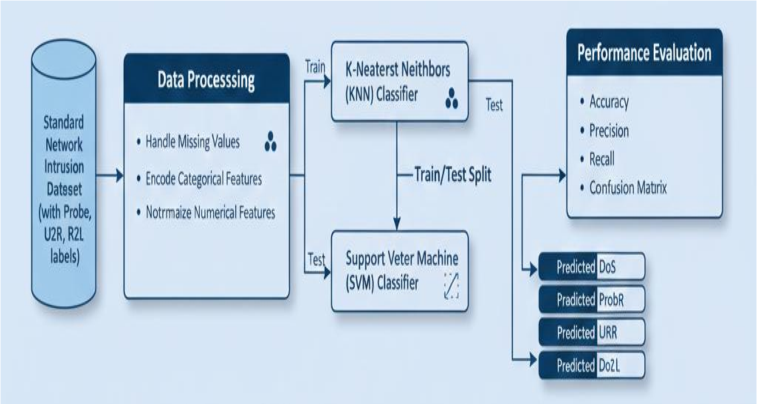
Standard Network Intrusion Dateset (with Probe, U2R, R2L labels)
Data Processsing
• Handle Missing Values
• Encode Categorical Features
• Notrmaize Numerical Features
Train
Test
K-Neaterst Neithbors (KNN) Classifier
Train/Test Split
Support Veter Machine (SVM) Classifier
Test
Performance Evaluation
• Accuracy
• Precision
• Recall
• Confusion Matrix
Predicted DoS
Pradicted ProbR
Predicted URR
Predicted Do2L

Fig 1: Proposed Model MKS Architecture

3.3. Algorithm:

```
Input: Network traffic dataset D
Output: Classification of traffic into DoS, Probe, U2R, or R2l
Step 1: Load the network intrusion dataset D
Step 2: Perform data preprocessing
    a.  Handle missing values
    b.  Encode categorical features
    c.  Normalize numerical attributes
Step 3: Split the dataset into training set D_trainand testing set D_test
Step 4: Train the KNN classifier using D_train
Step 5: Train the SVM classifier using D_train
Step 6: For each instance in D_test:
    a.  Predict attack class using KNN
    b.  Predict attack class using SVM
Step 7: Classify the traffic into one of the following categories:DoS, Probe, U2R, R2L
Step 8: Evaluate model performance using accuracy and other metrics
Step 9: Compare KNN and SVM results and select the best-performing model
```

3.4. K- Nearest Neighbors (KNN) Model

Fig2 shows, The KNN algorithm is a supervised, instance-based learning technique used for classifying network traffic based on similarity measures. In the proposed intrusion detection system, KNN classifies network traffic records by comparing them with labeled instances in the training dataset. Unlike parametric models, KNN does not require a distinct training stage; rather, it retains the entire set of training instances and performs classification during the testing phase. Each incoming network instance is assigned a class label determined by the dominant class among its closest neighbors in the feature space.Let the training dataset be represented in equation 1.

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \qquad (1)$$

where $x_i \in \mathbb{R}^m$denotes an m-dimensional feature vector and $y_i$represents the corresponding attack class label (DoS, Probe, U2R, or R2L). For a given test instance x_q, the distance between $x_q$and each training instance $x_i$is computed using a distance metric. In this work, the **Euclidean distance** is employed and is defined in equation 2.

$$d(x_q, x_i) = \sqrt{\sum_{j=1}^{m} (x_{qj} - x_{ij})^2} \qquad (2)$$

After computing distances, the algorithm selects the $k$closest training instances with the smallest distance values. The class label of the test instance is then determined using majority voting among these $k$neighbors. The predicted class $\hat{y}$is given by:

$$\hat{y} = \arg \max_{c \in C} \sum_{i \in N_k} \delta(y_i = c)$$

where Cdenotes the set of possible attack classes, $N_k$represents the set of $k$nearest neighbors, and $\delta(\cdot)$is an indicator function that returns 1 if the condition is true and 0 otherwise.

To ensure effective distance computation, feature normalization is applied prior to classification so that all attributes contribute equally. The value of $k$is chosen experimentally to balance bias and variance, where a smaller $k$improves sensitivity to local patterns and a larger $k$enhances generalization. Due to its simplicity and effectiveness in multi-class classification, KNN is well suited for identifying different categories of network intrusions in the proposed model.

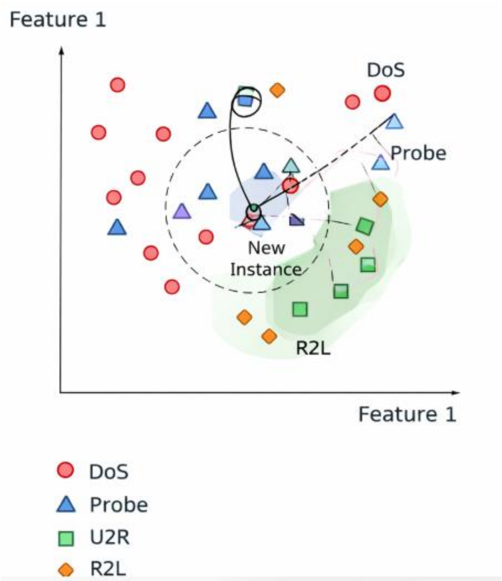

Fig 2: KNN Model Architecture

3.5. Support Vector Machine (SVM) Model

SVM displayed in fig3 is an effective supervised learning method commonly applied to classification problems, especially when handling datasets with a large number of features. In the proposed intrusion detection system, SVM is employed to classify network traffic into different attack categories, namely Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L). The primary objective of SVM is to construct an optimal decision boundary, referred to as a hyperplane, that maximally separates data instances belonging to different classes while minimizing classification errors.

Given a training dataset

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \qquad (4)$$

where $x_i \in \mathbb{R}^m$represents an m-dimensional feature vector and $y_i \in \{-1, +1\}$denotes the class label, SVM aims to find a separating hyperplane defined as:

$$w \cdot x + b = 0 \qquad (5)$$

where $w$is the weight vector and $b$is the bias term.

The optimal hyperplane is obtained by solving the following optimization problem:

$$\min_{w,b} \frac{1}{2} \parallel w \parallel^2 \qquad (6)$$

subject to the constraint:

$$y_i(w \cdot x_i + b) \geq 1, i = 1,2, \dots, n \qquad (7)$$

To handle non-linearly separable data, slack variables $\xi_i$are introduced, resulting in the soft-margin SVM formulation:

$$\min_{w,b,\xi} \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{n} \xi_i \qquad (8)$$

subject to:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \qquad (9)$$

where $C$is a regularization factor that balances margin maximization with the reduction of classification errors.

For complex and non-linear decision boundaries, SVM employs kernel functions to map input data into a higher-dimensional feature space. The commonly used kernel function in this study is the Radial Basis Function (RBF), defined as:

$$K(x_i, x_j) = \exp\left(\ -\gamma \parallel x_i - x_j \parallel^2 \right) \qquad (10)$$

where $\gamma$is a kernel parameter that determines the influence of individual training samples.

For multi-class intrusion detection, the SVM classifier is extended using strategies such as one-versus-one or one-versus-all classification. The trained SVM model effectively distinguishes between different intrusion categories based on learned decision boundaries. Due to its robustness, ability to handle high-dimensional data, and strong generalization capability, SVM proves to be a reliable classifier for network intrusion detection in the proposed model.

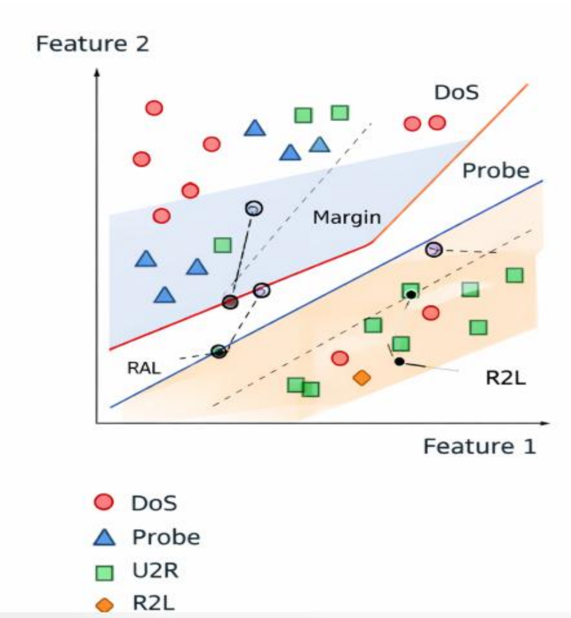

Fig 3: SVM Model Architecture

4. Results

This section presents the performance evaluation of the proposed Network Intrusion Detection System using KNN and SVM classifiers. The results are analyzed across four primary intrusion classes, namely DoS, Probe, R2L, and U2R, assessed with commonly used evaluation metrics. Figure 2 illustrates the comparative performance of KNN and SVM for each attack type. As shown in the accuracy plot, both classifiers achieve high accuracy for DoS and Probe attacks, with KNN slightly outperforming SVM across most categories. This indicates that instance-based learning is effective in distinguishing normal and malicious traffic patterns. The precision results in Figure 2 show that KNN consistently yields higher precision than SVM, particularly for R2L and U2R attacks, reflecting a lower false-positive rate and improved reliability in attack identification.

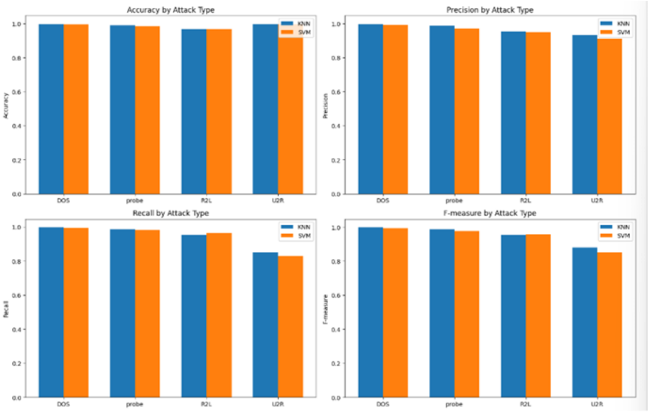

Fig 4: Visualization of the Evolutionary Metrics

The recall comparison in Fig 4 demonstrates strong detection capability for DoS and Probe attacks for both classifiers, while a noticeable decline is observed for U2R attacks due to the limited number of instances in the dataset. Despite this challenge, KNN maintains relatively better recall than SVM. Furthermore, the F-measure results highlight that KNN achieves a better balance between precision and recall across all attack categories, confirming its robustness and stability.

Overall, the results presented in the Table 1 confirm that both KNN and SVM are effective for intrusion detection; however, KNN consistently outperforms SVM across all evaluation metrics, making it more suitable for multi-class intrusion detection tasks. The obtained results demonstrate the reliability of machine learning-based approaches for enhancing network security and detecting diverse cyber threats.

Table 1: Performance of KNN and SVM for Different Attack Types

Attack Type	KNN Accuracy				SVM Accuracy			
	Accuracy	Precision	Recall	F-Measure	Accuracy	Precision	Recall	F-Measure
DoS	0.9971	0.9967	0.9966	0.9967	0.9937	0.9910	0.9945	0.9927
Probe	0.9907	0.9860	0.9850	0.9855	0.9845	0.9690	0.9836	0.9761
R2L	0.9674	0.9532	0.9548	0.9540	0.9679	0.9485	0.9626	0.9552
U2R	0.9970	0.9314	0.8507	0.8783	0.9963	0.9105	0.8290	0.8486

5. Conclusion

The authors performed class-wise intrusion detection analysis using the KDD dataset by applying supervised machine learning techniques for intrusion detection systems (IDS). Separate training and testing datasets were prepared to evaluate the models' effectiveness in identifying various categories of network attacks. The impact of dataset size on model training time was examined, and performance was assessed by gradually increasing the amount of training data. Improvements in evaluation measures such as accuracy, precision, recall, and F-score were observed as the dataset size increased. Based on the experimental results, the SVM model achieved the best performance, reaching a maximum accuracy of 98.69%.The evaluation criteria considered in this study include accuracy, computational complexity, model training time, classification time for unseen data, and ease of interpreting the final results. Relying on a single metric, such as accuracy, is insufficient for identifying the most effective machine learning approach. For a fair comparison based on accuracy, all models must be trained and tested using identical training and testing datasets. However, in several existing studies, although the same dataset and machine learning techniques were employed, different subsets of features were selected. As a result, the training and testing conditions were not always consistent across studies.

References

[1] Tsai, Flora. (2009). Network intrusion detection using association rules. LETTERS International Journal of Recent Trends in Engineering. 2.

[2] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of Machine Learning Algorithms for Intrusion Detection System," arXiv, Jan. 2018. [Online]. Available: https://arxiv.org/abs/1801.02330

[3] Nguyen, Huy &Deokjai, Choi. (1970). Application of Data Mining to Network Intrusion Detection: Classifier Selection Model. 399-408. 10.1007/978-3-540-88623-5_41.

[4] S. Paliwal and R. Gupta, "Denial-of-Service, Probing & Remote to User (R2L) Attack Detection using Genetic Algorithm," International Journal of Computer Applications, vol. 60, no. 19, pp. 57–62, Dec. 2012, doi: 10.5120/9813-4306.

[5] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 2009, pp. 1-6, doi: 10.1109/CISDA.2009.5356528.

[6] Arul, Amudha &Subburathinam, Karthik &Sivakumari, S.. (2013). Classification Techniques for Intrusion Detection An Overview. International Journal of Computer Applications. 76. 33-40. 10.5120/13334-0928.

[7] M. Alkasassbeh and M. Almseidin, “Machine Learning Methods for Network Intrusion Detection,” arXiv preprint, Sep. 2018. [Online]. Available: https://arxiv.org/abs/1809.02610

[8] Almseidin, Mohammad & Alzubi, Maen &Alkasassbeh, Mouhammd& Szilveszter, Kovács. (2019). Applying Intrusion Detection Algorithms on the KDD-99 Dataset. Production Systems and Information Engineering. 8. 51-67. 10.32968/psaie.2019.004.