Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

Development of a Communication-Efficient Framework for Intrusion Detection Using Machine Learning

Mr. Bharat Bajaj
PG Scholar
Department of Information Technology
Tulsiramji Gaikwad-Patil College of Engineering & Technology, Nagpur, India

Prof. Abhay Rewatkar
Project Guide
Department of Information Technology
Tulsiramji Gaikwad-Patil College of Engineering & Technology, Nagpur, India

Prof. Nilesh Nagrale
Project Co-Guide
Department of Information Technology
Tulsiramji Gaikwad-Patil College of Engineering & Technology, Nagpur, India

Abstract- Network traffic patterns with abnormal behavior which cause service interruptions must be analyzed through a labor-intensive process because modern networks generate huge amounts of unpredictable data. Intrusion Detection Systems (IDS) establish essential functions for detecting unauthorized activities while they notify system administrators about security breaches and attacks. Signature-based IDS systems work well for detecting known threats but they fail to identify new or developing security threats. The research introduces an intelligent intrusion detection framework which uses machine learning methods to detect network attacks automatically. PCA, or principal component analysis, is used by the system to optimize features and reduce dimensionality, which improves model performance and lowers processing requirements. To enhance detection performance, the system employs a variety of classification approaches, such as K-Nearest Neighbor (KNN), Random Forest Decision Tree, and Support Vector Machine (SVM). Through its operational capabilities, which include processing features like duration, source and destination bytes, protocol type, and connection behavior, the suggested system analyzes actual network traffic. The system can monitor activities in real time and achieve excellent detection accuracy, according to the experimental results. Flask-based web deployment combined with machine learning offers efficient, approachable solutions for contemporary network intrusion detection systems.

Keywords— *Intrusion Detection System, Principal Component Analysis, Machine Learning, Anomaly Detection, Network Security etc.*

I. INTRODUCTION

The fast development of computer networks together with cloud computing and internet-based services has created better ways to communicate and share data which helps different industries to achieve their digital transformation goals. Because cybercriminals increasingly employ a variety of attack techniques, such as ransomware, phishing, botnets, Distributed Denial the Service attacks, and unwanted network access, the development has increased the hazards that system security must contend with. Cyberattacks can lead to data breaches which result in financial losses and business operations get disrupted. The digital security of network systems has turned into an essential requirement for contemporary digital network systems. Security systems known as Intrusion Detection Systems (IDS) serve as common protective tools which track network data and detect dangerous activities [1].

The primary method used by traditional IDS systems relies on matching known attack patterns to existing signatures through signature-based detection systems. The systems can effectively identify established threats through their current capabilities, but they face challenges when attempting to identify new unknown attacks. The traditional IDS system needs ongoing updates to its signature database, which makes it difficult to keep up with the new attack methods that attackers are developing [2]. The current limitations of detection systems for intrusion are being addressed by machine learning (ML) along with artificial intelligence (AI) techniques. The ML-based IDS system learns from past data patterns to identify security breaches which they detect through their machine learning algorithms that create new detection methods without using existing security protocols [3].

In dividing network traffic into benign and malevolent categories, machine learning algorithms including support vector machine (SVM), Random Forests, Decision Trees, as well as K-Nearest Neighbor (KNN) showed encouraging results. Random Forest ensemble learning method increases detection performance while decreasing overfitting. SVM operates effectively for high-dimensional data classification tasks [4]. Machine learning models have started using Principal Component Analysis (PCA) dimensionality reduction method to decrease dataset size while removing unnecessary features and enhancing processing speed [5].

Publicly available datasets have established themselves as essential resources for conducting IDS research. Researchers use the benchmark standards established by the KDD Cup 99, NSL-KDD, CICIDS, and UNSW-NB15 datasets to create and evaluate ML-based IDS systems. Class imbalance, redundant features, and inadequate depiction of real-world attacks are the three main problems with the datasets that impair model performance and generalizability

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

to novel scenarios [6]. The use of normalization and feature extraction and feature selection preprocessing techniques functions as vital methods which enhance IDS performance and accuracy.

In recent years, web-based intrusion detection solutions have gained attention due to their accessibility and real-time monitoring capabilities. Flask provides developers with a framework which allows them to transform machine learning-based IDS systems into user-friendly applications that let administrators upload data sets and run model tests while they observe the resulting prediction information [7].

The systems deliver both detection capabilities and automatic reporting functions which allow users to implement them in real-world situations. The research work develops a machine learning-based intrusion detection system which combines PCA-based feature optimization with multiple classification algorithms. The system aims to achieve three goals which include bettering detection accuracy and cutting down system demands and providing continuous monitoring of intrusions. The system develops practical intrusion detection solutions which function in modern network environments through its combination of machine learning models and web-based execution [8].

II. PROBLEM IDENTIFICATION

• The fast development of digital networks has raised the risk of cyber-attacks which include DDoS attacks and botnets and ransomware and brute-force incursions.
• The signature and rule-based detection methods of traditional systems cannot identify new and zero-day security threats.
• The standard IDS methods face challenges in monitoring network traffic because of the massive amount of data that needs real-time analysis.
• The NetFlow-based intrusion detection systems create high-dimensional datasets that contain duplicate data and unnecessary information.
• The dataset imbalance problem which results in insufficient representation of malicious traffic leads to learning biases that decrease attack detection performance.
• The majority of current ML-based IDS systems produce excessive false positives which diminishes system trustworthiness and analyst confidence.
• The system suffers from increased processing demands and higher data transfer expenses because of ineffective feature selection methods.
• Deep learning models deliver precise results but they need extensive computing power and prolonged learning periods.
• The existing IDS systems lose their ability to scale and adapt when network environments change.
• The application of deep learning and machine learning models in real-world security operations is hampered by our incomplete knowledge of these models.

III. RESEARCH METHODOLOGY

A. *Proposed System*

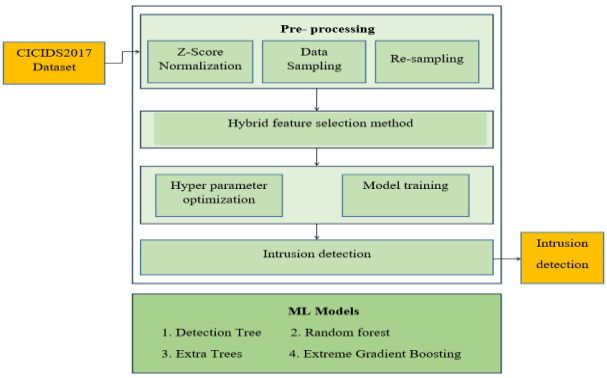

Figure 1. Architecture of the proposed intrusion detection system

CICIDS2017 Dataset Input:
• To begin, the CICIDS2017 dataset was chosen since it covers a wide range of contemporary attack methods and real-world network activity.
Pre-Processing Stage:
• The three crucial factors to take into account before advancing to statistical modeling are described in this content block.
Z-Score Normalization:
• Scales the feature values to the same range, thereby improving the stability of model training.
Data Sampling:
• Small training and testing samples are selected from which data instances with a certain distribution of properties are replicated.
Re-sampling:
• To solve the problem of class imbalance brought on by the targeted minority attack types, it may entail either oversampling or undersampling.
Hybrid Feature Selection Method:
• A hybrid choosing features method is used following preprocessing. The method uses multiple statistical measures to determine which features have the highest ability to distinguish between different classes. The process decreases data dimensions while eliminating duplicate features and enhancing both processing speed and data transmission performance.
Model Development Stage:
• Includes two interconnected processes:
Hyperparameter Optimization:
Adjust the model's settings to minimize performance issues.
Training of Models:
• Uses feature-refined and optimized data to train machine learning classifiers.
Intrusion Detection Output:
• After training, the model can identify if an activity is normal versus an attack depending on certain inputs.
Machine Learning Models Used:
Four tree-based supervised machine learning algorithms are used by the framework:

- Detection Tree
- Random Forest
- Extra Trees

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

- Extreme Gradient Boosting (XGBoost)

Though some fake news detectors belong to the family of multi-label categorization methods, the others follow multi class or two-class classification strategies.

B. Tools and Platform Used

• Python is the main programming language used for creating machine learning models, preparing data, and implementing system logic, because of its easy-to-use nature and wide-ranging library availability, serves as the development language.
• Pandas Library: The library enables data cleaning and preprocessing together with dataset handling and data manipulation through DataFrames and organized data processing capabilities.
• NumPy Library: The library enables users to perform numerical calculations and array handling and mathematical computations on extensive datasets with high efficiency.
• Scikit-learn (Sklearn): Along with processing and evaluation features, the platform offers machine learning tools such as SVM, Random Forest, Decision Tree, and KNN algorithms.
• The Flask Framework: This framework lets developers build an online interface that lets users run models, upload datasets, and view the outcomes.
• Google Colab and Jupyter Notebook: The platforms enable users to develop test and experiment with their models.
• Development tools and the intrusion detection system operate on Windows and Linux operating systems.
• Matplotlib: The library enables users to create performance visualizations and generate result graphs.

IV. SYSTEM METHODOLOGY

We are developing a system for intrusion detection that people may interface with. Information about the user, including time frame, src_bytes, dst_bytes, logged_in, number of hosts, srv_count, dst_host_count, protocol type, function, and flag, can be entered by users. The server will then collect the data, extract its features, match the values, categorize it, predict the intrusion, and report.

System Side Process:

• Store Dataset: The system stores the dataset that the user has supplied.
• Model Training: The user's data is collected by the system and fed into the chosen model.
• Model Predictions The system forecasts the outcomes using the data the user provides.

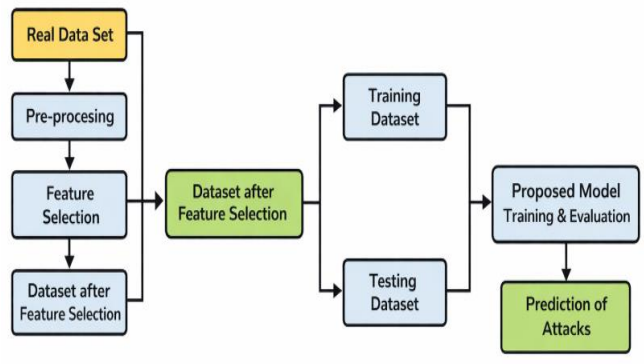

Figure 2: Flowchart of system

User Side Process:

• Load Dataset: A user has the option to load the dataset they wish to work with.
• View Dataset: The user has access to the dataset.
• Model selection: To verify the model's accuracy, the user may apply it to the dataset.
• Evaluation: The user can assess the model's performance.

Advantages:

The implementation of intrusion detection systems (IDS) provides different benefits to organizations which include the following advantages:
• The IDS system examines network traffic to find patterns that differ from typical system behavior in order to identify known threats using signature-based detection and unidentified threats using anomaly-based detection.
• The IDS system tracks network traffic in real time while alerting security personnel about possible attacks that take place during that time. The security team can establish an effective response plan through this process which will help to minimize the attack effects while decreasing their response time.
• ID systems enable organizations to enhance their network performance through two processes which involve finding network problems and implementing solutions that reduce network downtime and traffic interruptions.

Algorithms Used:

1. Random Forest Regression:

• A regression function is used in this categorization learning process. To get a common outcome or output, it uses a variety of decision trees throughout the training phase. By comparing the characteristics of the objects and computing the mean, it can be utilized to differentiate between them.
• A remedy for overfitting issues. Despite not being as accurate as gradient-boosted trees, decision trees outperform random forests. The quality of the data determines the performance.
• The variation is minimal when all of the selection trees are connected. It has a wide range of variations on its own. A good deal of any classifier's votes are taken into consideration when making the final conclusion regarding the categorized issue. This is a collection approach to tasks involving regression and classification.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

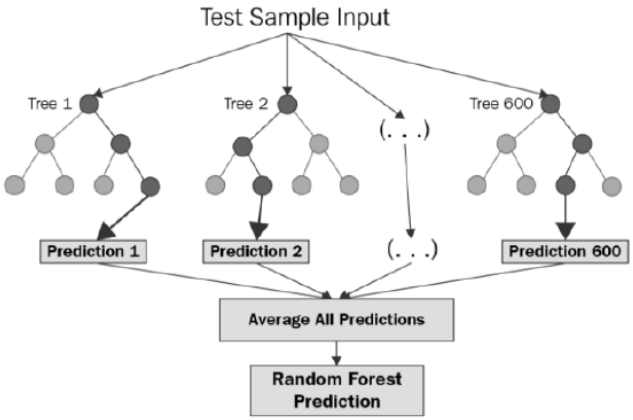

Figure 3: Random Forest Regression

2. Support Vector Machines:

• This classification method is used to identify the data spots on a hyperplane. This is done in a space of N dimensions. Support vector machines are shortened to SVM.
• This does the same duties with Random Forest Regression in terms of classifying the data and producing an standard deviation for it.
• Its main purpose is classification. While data points that do not fall on the identical hyperplane are ignored, those that do are considered to be in the exact same category. Accuracy is enhanced by this.
• Support vectors are necessary to accurately determine the hyperplane's location and inclination. This is used to broaden the scope of the categorization process. This procedure will be removed in order to modify the plane's orientation.

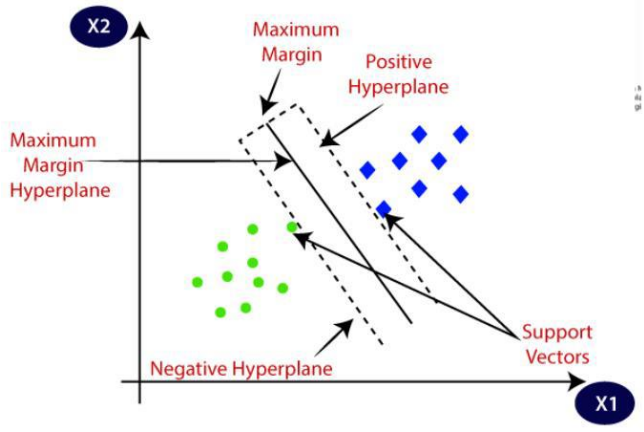

Figure 4: Support Vector Machines

3. Decision Trees:
A well-known machine learning technique that handles both regression and classification tasks is the decision tree. The system uses a tree structure, with leaf nodes that display expected outputs and class labels and core nodes that base choices on feature values.

• Recursively dividing the data on each node according to the feature that maximizes information gain or minimizes impurity is how the decision tree technique builds the tree. The procedure is repeated until
• The system continues until all data in a leaf node becomes identical to a single class of objects. The system ends when leaf nodes reach their established minimum count of instances.

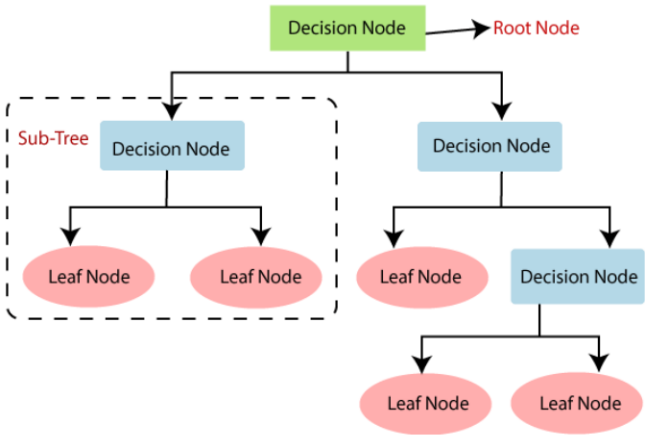

Figure 5: Decision Tree

4. K-Nearest Neighbor:

One of the main machine learning techniques that produces effective results for regression as well as classification issues is the K closest neighbor (KNN) algorithm. Since it doesn't require any particular assumptions regarding the underlying data distribution, the algorithm functions as a non-parametric system.

• The KNN approach locates the K training set data points that are most similar to the input point of data. Euclidean distance is the distance metric used in distance calculations. The K closest neighbors vote by majority to establish the projected result or category label.
•To improve the model's performance, the hyperparameter K needs to be changed. When K is selected as a low number, the model will have overfitting issues; when K is given to a high value, the model will have underfitting issues.

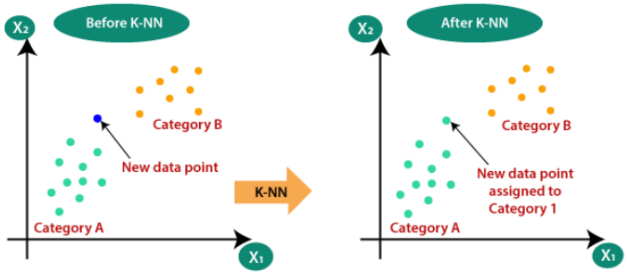

Figure 6: K-Nearest Neighbor

V. RESULTS ANALYSIS

The model was trained using four artificial intelligence classifier that considered the following features: "src bytes," "duration," "logged in," "count," "srv count," "dst host count," "protocol type," "dst bytes," "service," and "flag" check for any detected intrusion.

Ui Output:

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

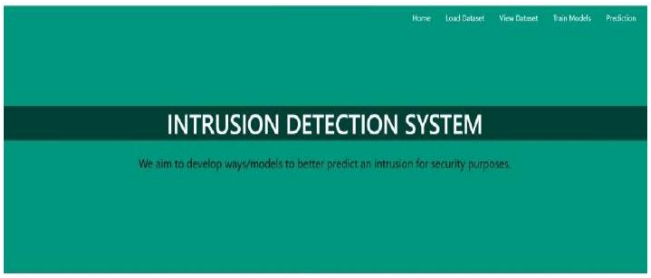

Figure 7: Home page

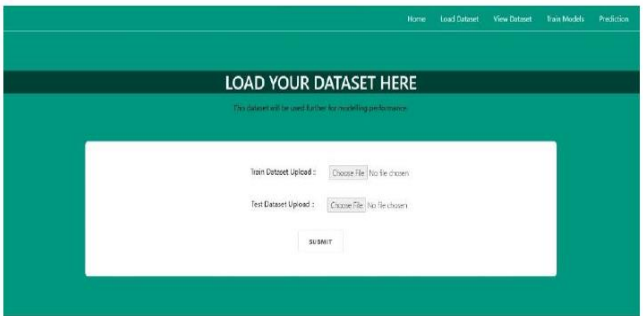

Figure 8: Dataset Uploading page

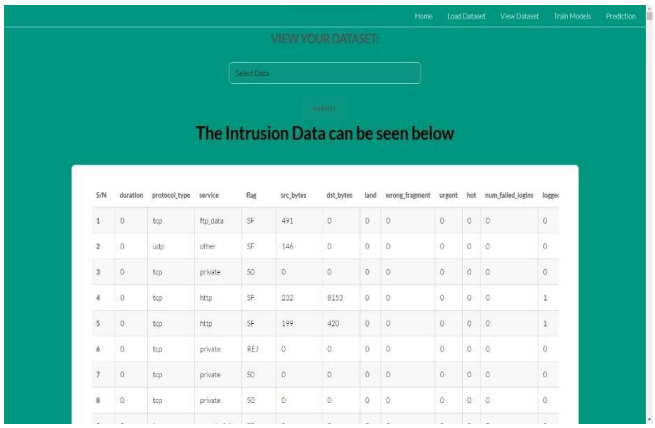

Figure 9: Dataset view page

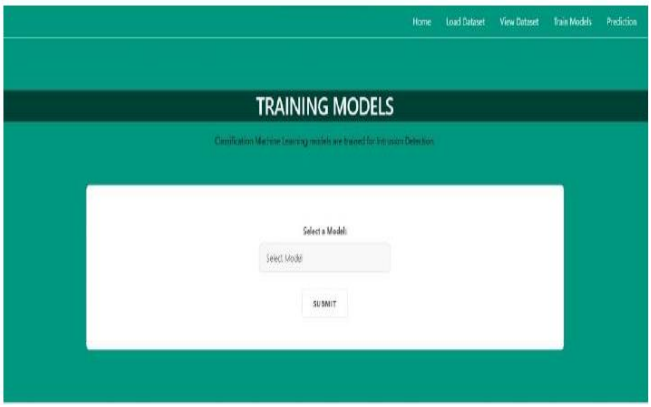

Figure 10: Model Selection page

Support Vector Machine(SVM):

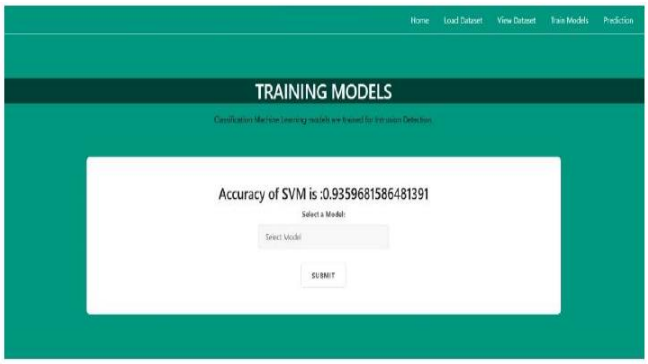

Figure 11: SVM Accuracy page

Random Forest Regression:

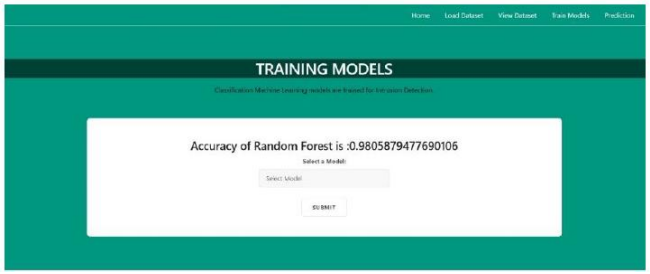

Figure 12: Random Forest Accuracy page

Decision Tree:

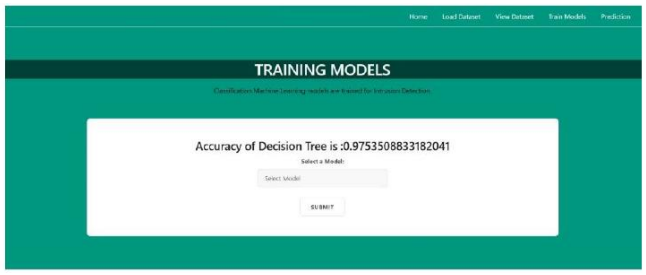

Figure 13: Decision Tree Accuracy page

K-Nearest Neighbor:

Figure 14: KNN Accuracy page

Out-page:

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

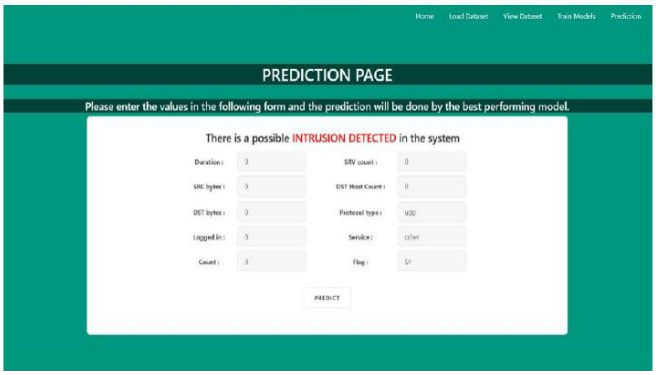

Figure 15: Intrusion Detected page

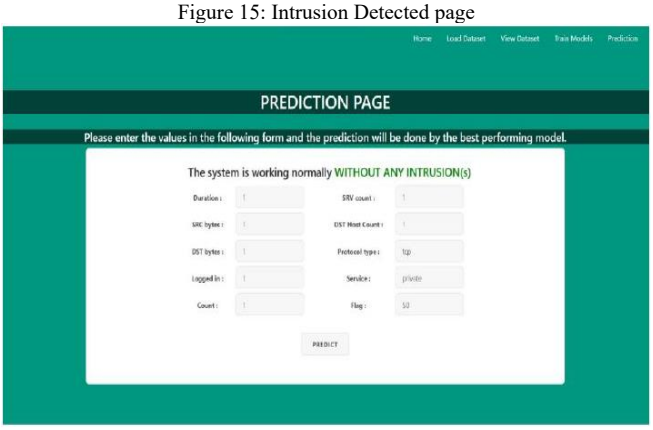

Figure 16: No Intrusion Detected page

Table 1: Performance Comparison of IDS Models

Model	Accuracy (%)	F1 Score
SVM	93.5	0.92
Random Forest	98.1	0.97
Decision Tree	97.4	0.96
KNN	97.6	0.95

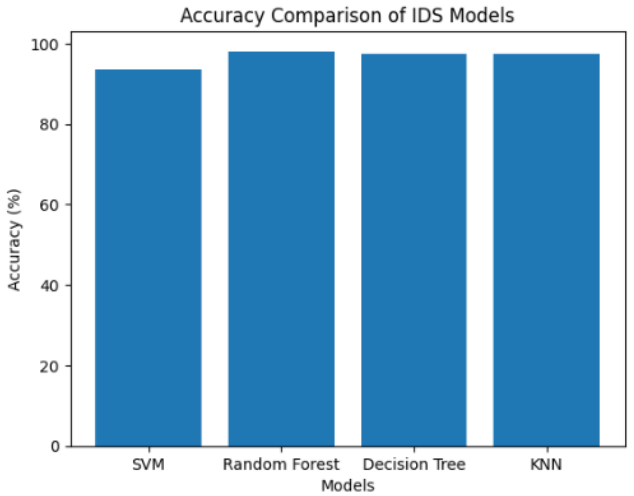

Figure 17. The accuracy comparison of IDS models

The findings from the examined intrusion detection research show that machine learning-based classifiers are highly effective in identifying cyberattacks. With an accuracy of roughly 98.1%, the method known as Random Forest outperformed the other evaluated algorithms. K-Nearest Neighbor with Decision Tree came in second and third, respectively, with 97.6% with 97.4% accuracy. With a reliability rating of 93.5%, the supporting vector machine model demonstrated excellent performance. Because of its ensemble learning method, which reduces overfitting and improves model generalization capabilities, the Random Forest algorithm performs well.

The F1-score comparison shows that Random Forest model reliability because it reached a score of 0.97, which indicates the model achieves equal precision and recall. The Decision Tree and KNN algorithms demonstrated strong ability to classify network traffic patterns, which resulted in good performance. SVM showed slightly lower F1 performance because it needed specific parameter settings to handle dataset imbalance.

The research results demonstrate that ensemble models provide better stability for intrusion detection purposes when applied to high-dimensional network datasets. The feature optimization method PCA enabled the model to achieve better performance while reducing the amount of processing power required. The system developed with Flask technology enables users to perform real-time intrusion detection through its practical usability.

The ability to deal with zero-day attacks and to create models that work across various datasets continues to present difficulties for researchers. The results demonstrate that machine learning methods lead to better performance in IDS detection accuracy and operational efficiency than existing traditional approaches.

VI. Conclusion

Intrusion Detection Systems (IDS) allow computer networks to maintain their security because these systems identify and stop unauthorized activities which endanger the security of data. This study shows that traditional intrusion detection systems based on signature are less successful in identifying attacks than machine learning-based solutions. By combining PCA (principal component analysis) methods for reducing dimensionality with support vector machine (SVM) The random forest method Decision Tree with K-Nearest Neighbor (KNN) methods for classification, the system achieves outstanding performance in network intrusion detection. The Random Forest algorithm achieved the highest accuracy, outperforming all other evaluated models. Decision Tree with KNN came in second and third, respectively, while SVM produced consistent accuracy results.

The use of feature optimization techniques helped decrease computation requirements while boosting system performance for the optimized model. The system gained better usability through the development of a Flask-based web application which allowed users to monitor and predict system performance in real time. The system still faces two main challenges because it needs better techniques to detect zero-day attacks and it needs improved model performance across various datasets. The proposed IDS framework delivers a network security solution which organizations can expand while maintaining high operational efficiency and precise performance amid changing cybersecurity threats.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

REFERENCES

[1] A. Pinto *et al*., "Survey on intrusion detection systems based on machine learning," *Sensors*, vol. 23, no. 15, 2023.

[2] Wiley Editors, "A systematic and comprehensive survey of recent intrusion detection system strategies," *Wiley Journal*, 2023.

[3] B. R. Kikissagbe, M. Aguirre, T. Gryta, and M. Skowron, "Machine learning-based intrusion detection methods in IoT and networks: A review," *Electronics*, vol. 13, no. 9, pp. 1–22, 2024.

[4] V. Z. Mohale *et al*., "Evaluating machine learning-based intrusion detection: Explainable AI, performance, and trust," *Frontiers in Computer Science*, 2025.

[5] S. L. Jacob and P. Sultana, "A systematic analysis and review on IDS using machine learning and deep learning algorithms," *Journal of Computational and Cognitive Engineering*, 2024.

[6] A. J. A. Immastephy, M. Kai, and M. Izzaty, "A systematic review of deep learning-based intrusion detection systems," in *Proc. E3S Conf. / ICPES*, 2024.

[7] E. E. Abdallah, D. E. A. Mansour, and M. S. E. Moursy, "Intrusion detection using supervised machine learning: A review," *Procedia Computer Science*, vol. 215, pp. 350–357, 2022.

[8] IJCA Authors, "Advances in intrusion detection systems: ML, DL, and federated learning," *International Journal of Computer Applications*, 2024.

[9] SSRG Authors, "ML-driven intrusion detection for cloud and critical infrastructure: A survey," *International Journal of SSRG*, 2024–2025.

[10] M. Landauer, H. Jacobsen, and J. Hernandez, "Deep learning for anomaly detection in log data: A survey," *arXiv preprint*, arXiv:2207.00123, 2022.

[11] K. Anyaso, N. O. Peters, and S. Akinboro, "Transforming animal tracking frameworks using wireless sensors and machine learning algorithms," *World Journal of Advanced Research and Reviews*, vol. 24, no. 1, pp. 996–1008, 2024.

[12] V. K. Pandey *et al*., "An efficient and robust framework for IoT security using machine learning techniques," *Procedia Computer Science*, vol. 258, pp. 118–124, 2025.

[13] P. Gangwani, A. Perez-Pons, and H. Upadhyay, "Evaluating trust management frameworks for wireless sensor networks," *Sensors*, vol. 24, no. 9, p. 2852, 2024.

[14] A. Mitra and S. Das, "Leveraging AI-enabled WSNs for environmental monitoring," in *Wireless Ad-hoc and Sensor Networks: Architecture, Protocols, and Applications*, 2024, p. 214.

[15] S. K. Kuppuchamy *et al*., "Journey of computational intelligence in sustainable computing and optimization techniques: An introduction," in *Computational Intelligence in Sustainable Computing and Optimization*. Morgan Kaufmann, 2025, pp. 1–51.

[16] H. M. A. Fahmy, "WSNs applications," in *Concepts, Applications, Experimentation and Analysis of Wireless Sensor Networks*. Springer, 2023, pp. 67–242.

[17] I. Aqeel, "Enhancing security and energy efficiency in wireless sensor networks for IoT applications," *Journal of Electrical Systems*, vol. 20, no. 3s, pp. 807–816, 2024.

[18] M. Moslehi, "Exploring coverage and security challenges in wireless sensor networks: A survey," *SSRN Electronic Journal*, SSRN 5084663, 2024.

[19] J. Akram *et al*., "GalTrust: Generative adversarial learning-based framework for trust management in spatial crowdsourcing drone services," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 3, pp. 6196–6207, 2024.

[20] M. Y. B. Murthy and A. Koteswararao, "Applications, merits and demerits of WSN with IoT: A detailed review," *International Journal of Autonomous and Adaptive Communications Systems*, vol. 17, no. 1, pp. 68–88, 2024.

[21] V. K. Pandey *et al*., "Enhancing intrusion detection in wireless sensor networks using a Tabu search-based optimized random forest," *Scientific Reports*, vol. 15, no. 1, p. 18634, 2025.

[22] K. V. Vidyapeeth and L. Kalbhor, "Secure and scalable data aggregation techniques for healthcare monitoring in WSN," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 27, pp. 441–452, 2024.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)