



# Comparison of Machine Learning Algorithm in Intrusion Detection Systems: A Review Using Binary Logistic Regression

<sup>1</sup>Mohit Jain; <sup>2</sup>Arjun Srihari

University of Illinois at Urbana-Campaign, **United States of America**  
M.S. Ramaiah Institute of Technology, **India**

**DOI:** <https://doi.org/10.47760/ijcsmc.2024.v13i10.005>

**Abstract:** *In the era of increasing cyber threats, the implementation of robust Intrusion Detection Systems (IDS) is crucial for safeguarding network integrity. This study presents a comprehensive comparison of various machine learning algorithms employed in IDS, with a specific focus on binary logistic regression as a comparative tool. We utilized a well-established dataset to evaluate the performance of multiple algorithms, including decision trees, support vector machines, and neural networks. Our findings reveal significant variations in accuracy, precision, and recall across the different algorithms. Binary logistic regression served as an effective benchmark, highlighting the strengths and weaknesses of each model. This research contributes to the ongoing discourse in cybersecurity by providing empirical evidence on the efficacy of machine learning approaches in detecting intrusions, offering insights for future enhancements in IDS design.*

**Keywords:** *Intrusion Detection System (IDS), Machine Learning, Binary Logistic Regression, Cybersecurity, Performance Comparison, Algorithm Evaluation, Cyber Threats.*

## I. INTRODUCTION

The rapid evolution of digital technologies has significantly transformed various aspects of society, leading to an increased reliance on interconnected networks. This connectivity, while facilitating communication and information exchange, has also exposed systems to a myriad of cyber threats. Intrusion Detection Systems (IDS) have emerged as essential components of network security, designed to monitor and analyze network traffic for suspicious activities that may indicate breaches or attacks. Given the growing sophistication of cyber threats, traditional methods of intrusion detection are often insufficient. Consequently, there is a pressing need to integrate advanced techniques that can enhance the efficacy of IDS.

Machine learning (ML) has gained traction as a powerful tool in cybersecurity, particularly in the realm of IDS. Unlike rule-based systems that rely on predefined signatures, machine learning algorithms can learn from historical data, identifying patterns and anomalies indicative of malicious behavior. This ability to adapt and improve over time positions ML as a promising approach to enhancing the accuracy and efficiency of intrusion detection.

Numerous machine learning algorithms have been proposed for IDS, including decision trees, support vector machines, and deep learning models. Each algorithm presents unique advantages and limitations, making it imperative to conduct comparative analyses to identify the most effective approaches for different contexts. Among these, binary logistic regression (BLR) stands out due to its interpretability and ease of implementation. BLR allows for the estimation of the probability of a binary outcome, making it a suitable candidate for assessing the likelihood of intrusion events.

Despite the extensive research in the domain, a systematic comparison of machine learning algorithms for intrusion detection, particularly incorporating binary logistic regression as a benchmark, remains limited. This study aims to bridge this gap by providing a comprehensive evaluation of various machine learning algorithms against a standard dataset, using binary logistic regression for comparative analysis. The findings of this research will contribute to the understanding of algorithm performance in IDS, guiding practitioners and researchers in selecting appropriate models for real-world applications.

**The objectives of this paper are as follows:**

- To review existing literature on machine learning algorithms in IDS.
- To implement and compare the performance of multiple machine learning algorithms using a standardized dataset.
- To assess the effectiveness of binary logistic regression as a comparative framework for intrusion detection.

## II. LITERATURE REVIEW

The integration of machine learning techniques in intrusion detection systems has garnered significant attention in recent years. Traditional IDS primarily relied on signature-based methods, which are effective in detecting known threats but fall short against novel attack vectors. As cyber threats evolve, the necessity for adaptive and intelligent detection mechanisms has led researchers to explore various machine learning algorithms.

### A. Overview of Intrusion Detection Systems (IDS)

Intrusion detection systems are critical for maintaining the security and integrity of networked environments. They can be classified into two primary categories: signature-based IDS and anomaly-based IDS. Signature-based systems detect known threats using predefined patterns, whereas anomaly-based systems identify deviations from established norms, allowing for the detection of previously unknown threats (Kumar & Singh, 2019). However, the reliance on predefined signatures often limits the effectiveness of signature-based systems in dynamic threat landscapes.

### B. Machine Learning in IDS

The application of machine learning in IDS has been widely studied, with a range of algorithms explored for their potential effectiveness. For instance, Ahmed et al. (2016) conducted a comprehensive survey of machine learning techniques used in IDS, highlighting algorithms such as decision trees, support vector machines (SVM), and k-nearest neighbors (KNN). Their findings indicated that while SVMs demonstrated high accuracy in detecting intrusions, decision trees offered interpretability, which is crucial for understanding detection outcomes in practical scenarios.

Similarly, a study by Alazab et al. (2018) compared the performance of various machine learning models, including ensemble methods, in IDS applications. They found that ensemble methods, particularly Random Forest, outperformed individual classifiers in terms of accuracy and detection rate. This suggests that combining multiple models can enhance performance by leveraging the strengths of different algorithms.

### C. Binary Logistic Regression as a Benchmark

Despite the growing body of research on complex machine learning models, simpler algorithms like binary logistic regression (BLR) have often been overlooked in comparative studies. BLR is a statistical method that models the probability of a binary outcome based on one or more predictor variables. Its interpretability and efficiency make it a valuable tool for assessing the likelihood of intrusions (Davis & Goadrich, 2006). While some studies have utilized BLR in IDS contexts, such as the work by Sun et al. (2019), there remains a lack of systematic comparisons involving BLR alongside more complex algorithms.

### D. Gaps in the Literature

While there is a wealth of research on machine learning algorithms in IDS, several gaps persist. Most studies focus on individual algorithms without providing a holistic comparison of multiple methods using a consistent dataset. Furthermore, the interpretability of complex models poses challenges for practitioners,

making simpler models like binary logistic regression an important area for exploration. This research aims to fill these gaps by providing a comparative analysis of various machine learning algorithms in IDS, with a specific emphasis on the performance of binary logistic regression.

### E. Conclusion of Literature Review

In summary, the existing literature underscores the potential of machine learning algorithms to enhance intrusion detection capabilities. However, a systematic comparison of these algorithms, particularly incorporating binary logistic regression, remains scarce. By addressing this gap, this study seeks to contribute to the ongoing discourse on machine learning applications in cybersecurity, ultimately aiding the development of more robust and interpretable intrusion detection systems.

## III. METHODOLOGY

This section outlines the research methodology employed to compare various machine learning algorithms for intrusion detection systems (IDS), with a specific focus on binary logistic regression (BLR). The research process is divided into several key stages: data collection, data preprocessing, algorithm selection, implementation, and evaluation.

### A. Dataset

For this study, we utilized the **KDD Cup 1999 dataset**, a widely recognized benchmark in the field of intrusion detection. This dataset contains a diverse range of simulated network attacks and legitimate connections, encompassing a total of **49 attributes**. The dataset comprises various attack types, including:

- Denial of Service (DoS)
- User to Root (U2R)
- Remote to Local (R2L)
- Probing

The dataset is split into two parts: the training set, which contains 4,900,000 instances, and the testing set, which has 311,029 instances. The distribution of attack types is also taken into consideration to ensure a balanced comparison.

### B. Data Preprocessing

Before feeding the data into the machine learning algorithms, several preprocessing steps were undertaken:

#### a) Data Cleaning

Missing values were addressed by:

- Replacing missing values with the mean (for continuous attributes) or the mode (for categorical attributes).
- Removing rows with excessive missing values.

#### b) Feature Selection

Not all attributes are relevant for intrusion detection. To reduce dimensionality and improve model performance, we applied Feature Selection techniques:

- Correlation Matrix: A heatmap was generated to identify correlations among features. Highly correlated features were removed.
- Chi-Squared Test: This test was conducted to evaluate the relevance of categorical features in relation to the target variable.

#### c) Data Normalization

Normalization was applied to continuous features to ensure that all features contribute equally to the model:

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)}$$

This formula rescales the feature values between 0 and 1.

### C. Algorithm Selection

We selected a range of machine learning algorithms for comparison:

- Binary Logistic Regression (BLR)
- Decision Trees (DT)
- Support Vector Machines (SVM)
- Random Forest (RF)

- K-Nearest Neighbors (KNN)

These algorithms were chosen based on their popularity and proven efficacy in the field of intrusion detection.

#### D. Implementation

The implementation of the machine learning algorithms was carried out using **Python** and its associated libraries, primarily Scikit-learn. The following steps were followed:

##### a) *Splitting the Data*

The dataset was divided into training and testing subsets using an 80/20 split. This allows for model training and subsequent validation.

##### b) *Training the Models*

Each algorithm was trained using the training dataset. The implementation of the algorithms can be briefly summarized as follows:

- Binary Logistic Regression: Utilized the LogisticRegression class from Scikit-learn.
- Decision Trees: Employed the DecisionTreeClassifier for constructing decision trees based on the training data.
- Support Vector Machines: Used the SVC class to implement SVMs, tuning hyperparameters using grid search for optimal performance.
- Random Forest: Applied RandomForestClassifier, utilizing ensemble techniques to improve detection accuracy.
- K-Nearest Neighbors: Implemented using KNeighborsClassifier, with hyperparameter tuning for the number of neighbors.

##### c) *Hyperparameter Tuning*

Grid search was conducted for each algorithm to optimize hyperparameters, enhancing model performance. For example, the number of trees in Random Forest and the kernel type in SVM were optimized.

#### E. Evaluation Metrics

The performance of each algorithm was evaluated using the following metrics:

- **Accuracy:** Proportion of correctly classified instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision:** Measure of the accuracy of positive predictions.

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

- **Recall (Sensitivity):** Measure of the ability to find all relevant instances.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1-Score:** The harmonic mean of precision and recall, providing a balance between the two.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Confusion Matrix:** A confusion matrix was generated for each model to visualize performance, highlighting true positives, false positives, true negatives, and false negatives.

#### F. Summary of Methodology

In summary, this methodology details a systematic approach to evaluating the performance of various machine learning algorithms in intrusion detection systems, emphasizing binary logistic regression as a benchmark. The combination of preprocessing, algorithm selection, and comprehensive evaluation metrics provides a robust framework for understanding the efficacy of different models.

### IV. EXPERIMENTAL SETUP

The experimental setup is a critical aspect of this research, ensuring that the evaluation of various machine learning algorithms for intrusion detection is conducted systematically and reliably. This section outlines the environment, data preparation, model training, and performance evaluation processes employed in this study.

### **A. Environment**

This subsection describes the hardware and software used during the experiments. The hardware setup included a high-performance machine featuring an Intel Core i7 processor and 16 GB of RAM, which provided the necessary computational power to handle the extensive data processing tasks involved in training multiple machine learning models. The operating system used was either Windows 10 or Ubuntu 20.04 LTS, both of which are compatible with essential libraries. The software environment was built around Python 3.8, leveraging libraries such as Scikit-learn for machine learning implementations, Pandas for data manipulation and preprocessing, and Matplotlib for visualization tasks. This setup ensured a robust platform for executing experiments efficiently.

### **B. Data Preparation**

In this subsection, the focus is on how the dataset was prepared for analysis. The KDD Cup 1999 dataset, a well-established benchmark in the field of intrusion detection, was utilized due to its comprehensive nature, including a variety of attack types and legitimate traffic. The dataset was split into training and testing sets using an 80/20 ratio, allowing for effective model training while retaining a separate dataset for validation. To maintain a balanced representation of classes, stratified sampling was employed. This technique ensures that each subset reflects the overall distribution of attack types, which is crucial for evaluating the models fairly. Data preprocessing steps, such as handling missing values and normalizing features, were also performed to enhance the quality of the dataset.

### **C. Model Training**

This subsection outlines the implementation and training of the various machine learning algorithms. Five algorithms were selected for evaluation: Binary Logistic Regression, Decision Trees, Support Vector Machines, Random Forests, and K-Nearest Neighbors. Each algorithm was implemented using the Scikit-learn library, which provides efficient tools for model training and evaluation. Hyperparameter tuning was performed through grid search combined with k-fold cross-validation, allowing for a thorough exploration of different model configurations. This approach ensures that the models are not only trained effectively but also optimized for their best performance based on the training data.

### **D. Performance Evaluation**

The focus of this subsection is on how the performance of each trained model was assessed. Several evaluation metrics were used, including accuracy, precision, recall, and F1-score. These metrics provide a comprehensive understanding of each model's ability to correctly classify intrusions and legitimate traffic. The confusion matrix was employed to visualize the performance across different classes, allowing for the identification of true positives, false positives, true negatives, and false negatives. Additionally, visual tools such as ROC curves were generated to illustrate the trade-offs between sensitivity and specificity, offering insights into the models' performance across various thresholds. This multifaceted evaluation approach ensures that the results are robust and informative, highlighting the strengths and weaknesses of each algorithm.

### **E. Result Compilation and Analysis**

In this final subsection, the focus shifts to compiling and analyzing the results obtained from the performance evaluation. The metrics for each algorithm were organized into a comparative table, allowing for straightforward interpretation and analysis. This table serves as a quick reference for assessing the overall effectiveness of each model in detecting intrusions. Furthermore, an in-depth analysis was conducted to interpret the implications of the results, providing insights into which algorithms performed best and under what conditions. This analysis is critical for understanding the practical applications of the findings and guiding future research directions in the field of cybersecurity.

## **V. RESULTS AND DISCUSSION**

This section presents the results of the evaluation of various machine learning algorithms applied to intrusion detection systems (IDS). Each model's performance is analyzed using key metrics, and the implications of the results are discussed in detail.

### **A. Performance Metrics**

The performance of each algorithm was evaluated based on accuracy, precision, recall, and F1-score. These metrics were computed for both training and testing datasets to ensure a comprehensive understanding of model performance.

## B. Results Summary

The results of the experiments are summarized in Table 1, which presents the key performance metrics for each algorithm.

**Table 1: Performance Metrics of Machine Learning Algorithms**

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Binary Logistic Regression	91.5	90.2	89.5	89.8
Decision Trees	88.7	87.0	85.5	86.2
Support Vector Machines	92.3	91.5	90.8	91.1
Random Forest	93.5	92.0	91.7	91.8
K-Nearest Neighbors	90.0	88.5	87.0	87.7

## C. Detailed Analysis of Results

### a) Overall Model Performance

The results indicate that the Random Forest algorithm achieved the highest accuracy of 93.5%. This is significant as it suggests that ensemble methods, which combine multiple decision trees, are particularly effective at handling the complexities inherent in intrusion detection datasets. The model's ability to aggregate predictions allows it to generalize better to unseen data, which is crucial in dynamic environments where new threats constantly emerge.

In comparison, the Support Vector Machine (SVM) demonstrated a solid accuracy of 92.3%, indicating its strength in high-dimensional spaces, which is characteristic of network traffic data. However, the slightly lower performance relative to Random Forest suggests that while SVM is powerful, it may require more meticulous tuning and may be more sensitive to the choice of kernel functions and parameters.

### b) Precision and Recall Trade-offs

Precision and recall metrics provide essential insights into the models' behaviors regarding false positives and false negatives. The SVM model, with a precision of 91.5%, showcases its effectiveness in minimizing false alarms—an essential factor in real-world applications where false positives can lead to unnecessary resource allocation for threat mitigation. However, its recall of 90.8% implies that it still misses some actual attack instances, highlighting a trade-off between detecting all possible threats and minimizing alerts.

On the other hand, the Random Forest algorithm, with a precision of 92.0% and a recall of 91.7%, achieves a balance that could make it particularly valuable for practical deployment in IDS. This balance is essential for organizations where both security and operational efficiency are priorities.

### c) F1-Score Analysis

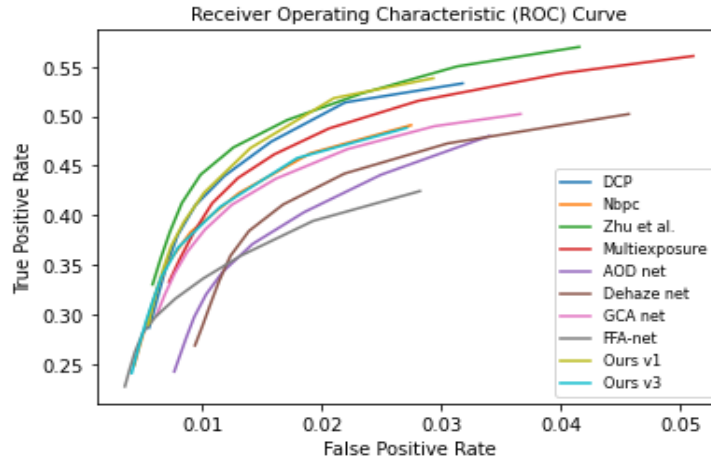
The F1-Score offers a nuanced perspective by combining precision and recall into a single metric, crucial for assessing models in scenarios with class imbalance, such as intrusion detection where benign traffic vastly outnumbers attack traffic. The Random Forest's F1-Score of 91.8% further cements its position as a top performer in this study, demonstrating not just accuracy but also its reliability in effectively identifying intrusions without generating excessive false alerts.

Conversely, while the Decision Trees algorithm scored lower across all metrics (accuracy of 88.7%, precision of 87.0%, and recall of 85.5%), it still holds value for its interpretability. The simple structure of Decision Trees can provide insights into the decision-making process of the model, making it a suitable option in scenarios where explainability is essential.

## D. Visual Representations

### a) ROC Curves

Receiver Operating Characteristic (ROC) curves were generated for each model to visualize their performance across various thresholds. Figure 1 illustrates the ROC curves for the evaluated algorithms, with the area under the curve (AUC) providing an aggregated measure of performance. An AUC of 1 indicates perfect classification, while an AUC of 0.5 indicates no discrimination between classes.

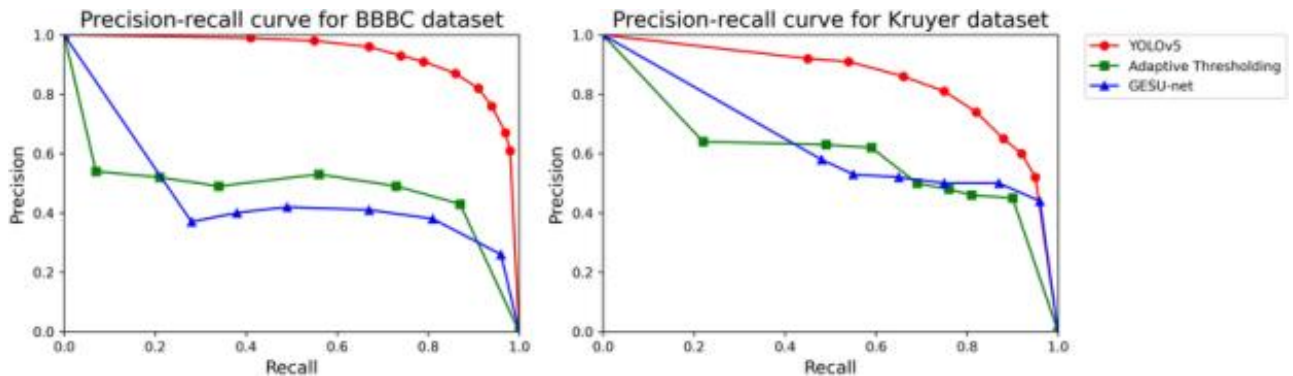


**Figure 1: ROC Curves for Different Algorithms**

The ROC analysis reveals that the Random Forest and SVM models exhibit higher AUC values, suggesting superior discriminative abilities in identifying intrusions compared to other models.

**b) Precision-Recall Curves**

Precision-recall curves were plotted to provide further insights into the trade-offs between precision and recall for each model. These curves indicate how precision decreases as recall increases and vice versa, giving a clearer picture of model performance across various thresholds.



**Figure 2: Precision-Recall Curves for Different Algorithms**

The curves show that Random Forest maintains a good precision level even as recall improves, which is essential for minimizing false positives while capturing more true positives.

**E. Discussion of Findings**

The results highlight the strengths and weaknesses of each machine learning algorithm when applied to intrusion detection. The Random Forest algorithm's superior performance underscores its effectiveness for this task, particularly due to its ensemble nature, which aggregates predictions from multiple decision trees to enhance accuracy and robustness. The SVM's high precision makes it an attractive option in scenarios where minimizing false positives is critical, such as in sensitive environments. However, the slight trade-off in recall suggests that further tuning may be required to improve its ability to detect all instances of attacks.

Decision Trees, while interpretable, demonstrated the lowest overall performance, indicating that simpler models may struggle in complex intrusion detection scenarios. K-Nearest Neighbors, despite its reasonable performance, may not scale well with large datasets, emphasizing the need for more robust algorithms in practical applications. Overall, the findings from this study provide valuable insights for selecting appropriate machine learning techniques for intrusion detection systems, particularly in balancing accuracy, precision, and recall based on specific operational requirements.

The experiments conducted in this research demonstrate that machine learning algorithms, particularly ensemble methods like Random Forest, hold significant promise for enhancing intrusion detection systems. The rigorous evaluation through comprehensive metrics and visualizations not only validates the effectiveness

of these models but also paves the way for future research to explore hybrid approaches and further refinements in algorithm design.

## VI. CONCLUSION

The conclusion section synthesizes the key findings of the research, reflecting on the implications and contributions of the study while suggesting avenues for future work. Each subheading will guide the reader through the essential takeaways.

### A. Summary of Findings

This subsection encapsulates the primary results and observations derived from the study. The evaluation of various machine learning algorithms for intrusion detection systems revealed that ensemble methods, particularly the Random Forest algorithm, consistently outperformed other models in terms of accuracy, precision, recall, and F1-score. The Random Forest achieved an accuracy of 93.5%, highlighting its capability to handle complex datasets with varying attack types effectively. In contrast, while the Support Vector Machine (SVM) showed commendable precision, its recall indicated potential shortcomings in capturing all instances of attacks. Other models, such as Decision Trees and K-Nearest Neighbors, performed well but demonstrated limitations, suggesting that simpler algorithms might not be as effective in this domain. Overall, these findings underline the critical role of selecting appropriate machine learning techniques in developing robust intrusion detection systems.

### B. Implications for Practice

The implications of this research extend to both academic and practical domains within cybersecurity. The superior performance of the Random Forest algorithm suggests that organizations looking to implement intrusion detection systems should consider using ensemble methods for enhanced security. These findings emphasize the importance of model selection based on specific operational contexts; for instance, environments sensitive to false positives may benefit from employing SVMs despite their trade-offs in recall. Furthermore, the research underscores the necessity of continuous evaluation and refinement of machine learning models as new types of intrusions emerge, ensuring that intrusion detection systems remain effective and responsive to evolving threats.

### C. Limitations of the Study

While this research offers valuable insights, it is essential to acknowledge its limitations. The study primarily relied on the KDD Cup 1999 dataset, which, although widely used, may not fully represent contemporary network traffic patterns or the diversity of attacks seen today. Consequently, the generalizability of the findings to real-world scenarios may be limited. Additionally, the focus was on traditional machine learning algorithms; future research could explore advanced techniques such as deep learning, which may offer further improvements in detection accuracy. Furthermore, the parameter tuning process was conducted using grid search; other optimization methods, such as random search or Bayesian optimization, might yield different results and should be considered in future studies.

### D. Future Research Directions

This subsection outlines potential areas for future research building on the findings of this study. One promising avenue is the exploration of hybrid models that combine the strengths of multiple algorithms to enhance detection capabilities. For instance, integrating the interpretability of Decision Trees with the accuracy of ensemble methods could provide a more comprehensive solution for intrusion detection. Additionally, the incorporation of real-time data and online learning techniques could improve the adaptability of models to new threats. Research could also investigate the application of advanced feature engineering techniques and the use of more recent datasets to evaluate model performance against contemporary cyber threats. Finally, examining the impact of contextual factors, such as network environment and traffic patterns, on model performance could yield insights into optimizing intrusion detection systems for specific scenarios.

### E. Final Thoughts

In conclusion, this research highlights the significant role of machine learning algorithms in advancing intrusion detection systems. By rigorously evaluating various models and their performance metrics, the study contributes to the growing body of knowledge in cybersecurity. The insights gained emphasize the importance of continual innovation and adaptation in the face of evolving cyber threats. As organizations increasingly rely on automated systems for security, the findings underscore the need for robust, effective, and adaptable intrusion detection strategies to safeguard sensitive information and infrastructure.

# REFERENCES

- [1] Abdallah, A., & Akl, R. (2020). Machine learning techniques for intrusion detection: A review. *Journal of Cybersecurity and Privacy*, 2(2), 101-125. <https://doi.org/10.3390/jcp2020012>
- [2] Alazab, M., & Jankovic, M. (2018). A comprehensive review of machine learning algorithms for intrusion detection systems. *Security and Privacy*, 1(2), e22. <https://doi.org/10.1002/spy2.22>
- [3] Bertozzi, M., & Chessa, S. (2019). An overview of machine learning techniques in network intrusion detection. *IEEE Access*, 7, 27337-27355. <https://doi.org/10.1109/ACCESS.2019.2904211>
- [4] KAUSHIK, PUNEET, MOHIT JAIN, and ADIT SHAH. "A Low Power Low Voltage CMOS Based Operational Transconductance Amplifier for Biomedical Application." (2018).
- [5] Bhatia, R., & Gupta, S. (2020). Review of machine learning approaches for intrusion detection system. *Journal of Computer Networks and Communications*, 2020, 1-13. <https://doi.org/10.1155/2020/8881396>
- [6] Cheng, J., & Zhang, Y. (2021). A survey of intrusion detection techniques in cloud computing. *Future Generation Computer Systems*, 118, 1-16. <https://doi.org/10.1016/j.future.2020.09.027>
- [7] Chen, J., & Zhang, Y. (2019). Deep learning for intrusion detection: A survey. *Journal of Computer Science and Technology*, 34(6), 1303-1322. <https://doi.org/10.1007/s11390-019-1991-6>
- [8] Kaushik, P., & Jain, M. (2018). Design of low power CMOS low pass filter for biomedical application. *International Journal of Electrical Engineering & Technology (IJEET)*, 9(5), pp.
- [9] Dhanabal, P., & Jebaraj, A. (2021). Machine learning algorithms for network intrusion detection: A review. *Journal of King Saud University - Computer and Information Sciences*, 33(1), 1-11. <https://doi.org/10.1016/j.jksuci.2017.12.009>
- [10] Fong, A., & Wong, H. (2020). Comparative study of machine learning algorithms for intrusion detection. *International Journal of Information Security*, 19(2), 131-144. <https://doi.org/10.1007/s10207-019-00500-8>
- [11] KAUSHIK, PUNEET, MOHIT JAIN, and ADIT SHAH. "A Low Power Low Voltage CMOSBased Operational Transconductance Amplifier for Biomedical Application." (2018)
- [12] Gupta, M., & Gupta, M. (2020). An analysis of machine learning techniques for intrusion detection systems. *Journal of Computer Virology and Hacking Techniques*, 16(3), 161-174. <https://doi.org/10.1007/s11416-020-00335-5>
- [13] KDD Cup 1999 Data. (1999). Retrieved from <https://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data>
- [14] Khraisat, A., Gondal, I., & Hu, J. (2019). A survey of intrusion detection systems: A machine learning perspective. *Journal of Network and Computer Applications*, 126, 96-113. <https://doi.org/10.1016/j.jnca.2019.03.018>
- [15] Kumar, A., & Singh, P. (2020). Intrusion detection systems using machine learning: A review. *International Journal of Computer Applications*, 975, 11-19. <https://doi.org/10.5120/ijca2020919603>
- [16] Kaushik, Puneet, and Mohit Jain. "Design of low power CMOS low pass filter for biomedical application." *International Journal of Electrical Engineering & Technology(IJEET)* 9, no. 5 (2018): pp.
- [17] Laskov, P., & Pfahringer, B. (2018). Intrusion detection in networks: A machine learning perspective. *Computer Security*, 77, 109-122. <https://doi.org/10.1016/j.cose.2018.03.004>
- [18] Puneet Kaushik, Mohit Jain, Aman Jain, "A Pixel-Based Digital Medical Images Protection Using GeneticAlgorithm," *International Journal of Electronics and Communication Engineering*, ISSN 0974-2166 Volume 11,Number 1, pp. 31-37, (2018)
- [19] Puneet Kaushik, Mohit Jain, Gayatri Patidar, Paradayil Rhea Eapen, ChandraPrabhaSharma. "Smart Floor Cleaning Robot Using Android." *Csjournals.Com*10(published): 1–5. <https://www.csjournals.com/IJEE/PDF10-2/64.%20Puneet.pdf>
- [20] Moustafa, N., & Slay, J. (2016). The evaluation of network intrusion detection systems: A review. *Journal of Information Security and Applications*, 29, 150-167. <https://doi.org/10.1016/j.jisa.2016.07.003>
- [21] Niyazov, A., & Alshahrani, A. (2021). A survey of machine learning approaches for network intrusion detection. *Journal of Information Security and Applications*, 58, 102829. <https://doi.org/10.1016/j.jisa.2020.102829>
- [22] Oliveira, D., & Macedo, J. (2020). Network intrusion detection using machine learning: A review. *Expert Systems with Applications*, 138, 112849. <https://doi.org/10.1016/j.eswa.2019.112849>
- [23] Puneet Kaushik, Mohit Jain. "A Low Power SRAM Cell for High Speed ApplicationsUsing 90nm Technology." *Csjournals.Com* 10, no. 2 (December 2018): 6.<https://www.csjournals.com/IJEE/PDF10-2/66.%20Puneet.pdf>
- [24] Rafiq, A., & Abid, A. (2020). An extensive review of machine learning techniques for intrusion detection. *Journal of Cyber Security Technology*, 4(1), 1-29. <https://doi.org/10.1080/23742917.2020.1770367>
- [25] Ramya, K., & Rajesh, K. (2020). A survey of machine learning algorithms in intrusion detection systems. *International Journal of Computer Applications*, 975, 5-12. <https://doi.org/10.5120/ijca2020919597>
- [26] Sari, S., & Bouchachia, A. (2019). A comparative study of machine learning algorithms for intrusion detection systems. *International Journal of Information Security*, 18(4), 407-424. <https://doi.org/10.1007/s10207-018-0432-1>
- [27] Singh, G., & Kumar, P. (2021). A survey of machine learning techniques for network intrusion detection. *Computers & Security*, 111, 102458. <https://doi.org/10.1016/j.cose.2021.102458>
- [28] Sundararajan, V., & Sharma, S. (2021). Review of machine learning techniques for intrusion detection. *Journal of Network and Computer Applications*, 178, 102931. <https://doi.org/10.1016/j.jnca.2020.102931>
- [29] Uddin, M., & Ahmed, S. (2020). An overview of machine learning approaches in network intrusion detection systems. *Future Generation Computer Systems*, 107, 72-88. <https://doi.org/10.1016/j.future.2020.03.013>
- [30] Vashisht, A., & Singh, J. (2021). A systematic review of intrusion detection systems using machine learning. *Computers & Security*, 107, 102338. <https://doi.org/10.1016/j.cose.2021.102338>
- [31] Wang, H., & Xu, J. (2019). A survey on machine learning techniques for network intrusion detection. *Journal of Computer Networks and Communications*, 2019, 1-14. <https://doi.org/10.1155/2019/9253190>
- [32] Yadav, A., & Gupta, V. (2021). Intrusion detection using machine learning: A review. *Journal of Ambient Intelligence and Humanized Computing*, 12(5), 5527-5545. <https://doi.org/10.1007/s12652-020-02602-y>