Global Journal of Engineering Innovations & Interdisciplinary Research

Original Article

Comparative Study of Intrusion Detection Systems: Machine Learning Vs Deep Learning Approaches

Vinod[1], Akurathi Lakshmi Pathi Rao[2], A Manikandan[3], Vanaparthi Kiranmai[4]

[1]Assistant Professor, Department of AI&DS, Guru Nanak Institutions Technical Campus, Hyderabad, India
[2]Assistant Professor, Department of CSE, Guru Nanak Institute of Technology, Hyderabad, India
[3]Associate Professor, Department of Computer Science and Engineering, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, India
[4]Research Scholar, Department of Computer Science and Engineering, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, India

Correspondence

Vinod

Assistant Professor, Department of AI&DS, Guru Nanak Institutions Technical Campus, Hyderabad, India

- Received Date: 25 May 2025
- Accepted Date: 15 June 2025
- Publication Date: 27 June 2025

Abstract

Intrusion Detection Systems (IDS) play a crucial role in safeguarding modern network infrastructures by identifying malicious activities and preventing potential security breaches. This study presents a comparative analysis of machine learning algorithms—Decision Trees, Support Vector Machines (SVM), Random Forest, and K-Nearest Neighbors (KNN)—to evaluate their effectiveness in intrusion detection. Using standard datasets such as KDD Cup 99 and NSL-KDD, each algorithm was tested based on accuracy, precision, recall, and F1-score. The results show that Random Forest outperforms other models with an accuracy of 95.3% and an F1-score of 94.2%, followed by SVM with a strong performance in high-dimensional data classification. Decision Trees demonstrated a reasonable balance between interpretability and performance, while KNN struggled with scalability and high-dimensional network traffic. These findings highlight the importance of selecting the appropriate machine learning technique for IDS, based on the specific requirements of the network environment and the complexity of potential threats.

Introduction

Intrusion Detection Systems (IDS) and Their Importance in Cybersecurity

Intrusion Detection Systems (IDS) are essential security mechanisms designed to monitor network traffic or system activities for malicious actions or policy violations. The primary goal of IDS is to detect and alert administrators of potential security breaches, such as unauthorized access, malware infections, or attempts to compromise systems. IDS can operate in various modes: network-based IDS (NIDS), which monitors the network traffic for suspicious activity, and host-based IDS (HIDS), which tracks system activities on individual devices. These systems are crucial in the early identification of attacks, allowing organizations to respond promptly before the intrusions cause severe damage.

In the age of increasing cyberattacks and sophisticated threats, IDS plays a pivotal role in enhancing an organization's cybersecurity posture. Traditional security measures, such as firewalls and antivirus software, may fail to detect more advanced or previously unknown threats. An IDS can fill this gap by continuously monitoring the system for anomalies, thus acting as a second layer of defense. As cyber threats evolve, the ability of an IDS to recognize new attack patterns is critical, which is why developing advanced detection techniques is an ongoing priority for cybersecurity professionals.

Differences Between Traditional Methods, Machine Learning (ML), and Deep Learning (DL) Approaches in Detecting Intrusions

Historically, intrusion detection relied on traditional methods, such as signature-based and rule-based systems. Signature-based IDS identifies attacks by comparing incoming data with known attack signatures, much like how antivirus programs detect malware. While effective for known threats, this approach struggles with unknown or zero-day attacks and requires constant updates to the signature database. Rule-based systems, on the other hand, rely on predefined rules to detect malicious behavior. These approaches are computationally efficient and easy to implement but lack adaptability and suffer from high false-positive rates in dynamic environments.

With the rise of machine learning (ML), IDS has seen a significant improvement in detecting anomalies and identifying new types of attacks. ML-based IDS can learn from

Citation: Vinod, Rao ALP, Manikandan A, Vanaparthi K. Comparative Study of Intrusion Detection Systems: Machine Learning Vs Deep Learning Approaches. GJEIIR. 2025;5(5):091.

historical data, identifying both known and unknown attack patterns. Algorithms such as decision trees, random forests, and support vector machines (SVM) have been employed to classify and predict intrusions based on the features extracted from network traffic or system logs. Unlike traditional methods, ML approaches can generalize from the data, making them more versatile in detecting novel attacks. However, ML methods still require manual feature engineering and may face challenges in high-dimensional data environments.

Deep learning (DL), a subset of ML, takes intrusion detection to a higher level by automating feature extraction and learning from large datasets. DL models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have the capability to process unstructured data and uncover complex patterns in network traffic that traditional ML models may miss. DL-based IDS can detect sophisticated attacks like advanced persistent threats (APTs) or zero-day exploits with a higher degree of accuracy, thanks to their ability to handle large amounts of data and uncover intricate patterns. However, these models are computationally intensive, requiring significant resources for both training and real-time detection, which poses challenges for deployment in resource-constrained environments.

Objectives and Significance of the Study

The objective of this study is to conduct a comparative analysis of machine learning and deep learning approaches in intrusion detection systems, with a particular focus on their effectiveness, accuracy, and computational efficiency. As the sophistication of cyberattacks continues to evolve, it is imperative to identify which approaches provide the most robust defense mechanisms against both known and unknown threats. This study aims to explore the capabilities of ML and DL methods, contrasting their strengths and weaknesses in terms of detection rate, false positives, and adaptability to new attack patterns.

The significance of this research lies in its potential to guide organizations and cybersecurity practitioners in selecting the most appropriate IDS model based on their needs. By comparing ML and DL techniques, the study seeks to highlight which approach offers superior performance in different scenarios, whether it be a high-speed network or a low-resource environment. Additionally, this work will contribute to the ongoing development of more effective IDS, helping to shape future research directions in the field of cybersecurity.

Literature Survey

Network-Based IDS (NIDS) is deployed at key points within a network, such as gateways, routers, or network switches, where it monitors the traffic flowing through the network. The primary function of NIDS is to inspect network packets, looking for signs of malicious activity, such as known attack signatures, unusual traffic patterns, or unauthorized access attempts. Since it operates on network traffic, NIDS can provide comprehensive monitoring across multiple devices and systems connected to the network, making it an effective defense against network-wide attacks, such as distributed denial-of-service (DDoS) attacks or man-in-the-middle (MITM) attacks. However, NIDS can struggle to detect attacks on encrypted traffic and may miss intrusions occurring within a host if they don't trigger network anomalies.

On the other hand, Host-Based IDS (HIDS) is installed on individual devices, such as servers, computers, or other endpoints. Instead of monitoring network traffic, HIDS tracks system-level activities, such as file modifications, logins, or process behavior. This allows HIDS to detect a wide range of threats, including malware infections, unauthorized file access, and privilege escalation attempts. Because HIDS operates at the host level, it can provide detailed insights into system-specific attacks that may not be visible on the network level. However, it has limited visibility into network-wide events and may become resource-intensive, particularly on systems with significant activity.

Both NIDS and HIDS have their strengths and limitations, and many modern organizations opt for a hybrid approach, combining both types of IDS to provide comprehensive protection against a wide array of cyber threats.

Signature-Based vs Anomaly-Based IDS

IDS can also be classified based on the detection method they use: signature-based detection and anomaly-based detection.

Signature-Based IDS relies on predefined signatures or patterns of known threats to detect intrusions. These signatures are derived from previously identified malicious activity, such as specific sequences of bytes in network packets or system log entries. When incoming traffic or system behavior matches a signature in the IDS database, an alert is triggered. Signature-based IDS is highly effective for identifying known attacks with well-documented signatures, such as common malware or exploits targeting specific vulnerabilities. Its primary advantage lies in its low false positive rate since it only raises alarms for explicitly known threats. However, the major limitation is that it cannot detect unknown or novel attacks (such as zero-day exploits) for which no signature has been created, leaving systems vulnerable to new threats until the signature database is updated.

In contrast, Anomaly-Based IDS does not rely on signatures. Instead, it builds a model of normal system or network behavior and flags any deviation from this baseline as suspicious. Anomaly detection techniques often involve statistical analysis or machine learning models that learn normal patterns of user activity, network traffic, or system performance. When an event occurs that significantly differs from the expected behavior (e.g., unusual traffic spikes, atypical login times, or unauthorized resource access), the system raises an alert. The advantage of anomaly-based IDS is that it can potentially detect previously unknown attacks by identifying behaviors that do not fit the norm. However, this approach tends to have a higher false positive rate, as legitimate deviations from normal behavior (e.g., a sudden spike in network traffic due to a legitimate event) may be mistakenly flagged as an intrusion.

Current Challenges in IDS

Despite the advancements in intrusion detection technology, IDS face several challenges that impact their effectiveness, particularly in today's fast-evolving cybersecurity landscape.

One of the most significant challenges is false positives. An IDS is often overwhelmed by large amounts of network or system data, and it is difficult to tune the system to detect actual intrusions without also flagging benign activities. High rates of false positives can lead to alert fatigue, where security teams may begin to ignore warnings, potentially missing actual threats. Finding the right balance between sensitivity and precision is a critical challenge, especially in environments where normal

behavior can be highly variable.

Another pressing challenge is real-time performance. With the increasing size and speed of modern networks, IDS must be able to process and analyze vast amounts of data in real-time to detect intrusions before damage occurs. For signature-based IDS, this means rapidly scanning through databases of known attack signatures without causing network bottlenecks, while anomaly-based IDS must continuously monitor and update their understanding of "normal" behavior. Both methods require significant computational power, especially for deep learning-based IDS, which can struggle with the demands of real-time analysis in high-speed networks.

A third major challenge is adapting to new threats. Cyberattacks are constantly evolving, and attackers frequently develop new tactics to bypass detection mechanisms. While signature-based systems are particularly vulnerable to novel threats due to their dependence on predefined signatures, anomaly-based systems must be continuously retrained to recognize new forms of malicious behavior. Even then, attackers can often mimic normal behavior to evade detection. This constant "cat-and-mouse" game between attackers and defenders requires IDS to be continuously updated, retrained, and fine-tuned, which can be resource-intensive for organizations.

Methodology

Common Machine Learning Algorithms Used in IDS

Machine learning (ML) has transformed the way intrusion detection systems (IDS) identify and respond to threats, offering more dynamic and adaptive methods compared to traditional signature-based approaches. Several ML algorithms have been widely used in IDS to enhance the detection of malicious activities. These include Decision Trees, Support Vector Machines (SVM), Random Forest, and K-Nearest Neighbors (KNN).

Decision Trees are one of the most intuitive and interpretable machine learning models used in IDS. A decision tree algorithm builds a model by recursively splitting the dataset into subsets based on the feature that offers the highest information gain. This process continues until the model identifies patterns in the data that differentiate between normal and malicious behavior. Decision trees are highly interpretable and fast to train, making them suitable for real-time intrusion detection. However, they are prone to overfitting, especially when the model becomes too complex by learning fine-grained details from the training data that may not generalize well to new, unseen attacks.

Support Vector Machines (SVM) are another commonly used algorithm in IDS. SVM operates by finding the hyperplane that best separates the data into different classes—such as normal and malicious traffic—by maximizing the margin between the two classes. SVM is effective in high-dimensional spaces and can handle nonlinear relationships using kernel functions. This makes it particularly useful for detecting complex attack patterns that linear algorithms may miss. However, SVM can be computationally expensive, especially in large datasets, and its performance depends heavily on the choice of kernel and hyperparameters.

Random Forest is an ensemble learning algorithm that combines the predictions of multiple decision trees to make a final classification. In IDS, Random Forest is known for its robustness and high accuracy, as it reduces the risk of overfitting by averaging the predictions of numerous trees, each built on a random subset of the features and data. This ensemble method works well in handling complex attack patterns, noisy data, and imbalanced datasets, which are common in intrusion detection scenarios. The major drawback is its higher computational cost and memory usage compared to individual decision trees, especially when applied to large-scale network traffic.

K-Nearest Neighbors (KNN) is another algorithm frequently applied in IDS. KNN is a simple yet powerful technique that classifies an instance based on the majority class of its nearest neighbors in the feature space. In the context of IDS, KNN can be used to detect outliers in network traffic, assuming that malicious activities often stand out from normal patterns. Although KNN is straightforward and easy to implement, it can become inefficient in large datasets since it requires storing the entire dataset and computing the distance to all points for every prediction. This makes KNN more suitable for offline analysis or smaller datasets.

Pros and Cons of Machine Learning in IDS

The use of machine learning in intrusion detection systems offers several advantages. One of the main benefits is the ease of training. Unlike traditional systems that require the creation of specific rules or signatures, ML-based IDS can be trained on labeled datasets to learn patterns associated with both normal and malicious behavior. This allows the system to adapt to new threats, including zero-day attacks, which were previously unseen in signature databases. Additionally, certain ML algorithms, such as decision trees and random forests, are highly interpretable, allowing security professionals to understand why the system flagged a particular activity as suspicious.

However, there are also challenges associated with the use of ML in IDS. While ML algorithms such as decision trees are easy to interpret, others, like SVM and ensemble methods, may be less transparent, making it difficult for cybersecurity analysts to fully understand their decisions. Moreover, performance in complex patterns can vary significantly between different ML algorithms. Some models, like KNN, may struggle with high-dimensional data or large-scale traffic, while others, like SVM, require careful tuning of hyperparameters for optimal performance. Additionally, ML-based IDS can suffer from data imbalance, where the number of normal instances far exceeds malicious ones, leading to potential misclassification of rare attack instances. The computational demands of certain algorithms, especially in real-time detection scenarios, also pose a challenge, as ML models may require significant processing power to classify incoming traffic in real-time.

Overall, while ML offers flexibility and adaptability in intrusion detection, it also requires careful tuning, selection of appropriate algorithms, and access to large and diverse training datasets to ensure its effectiveness.

Relevant Datasets Used for ML-Based IDS Evaluation

To train and evaluate machine learning models in intrusion detection, various benchmark datasets have been developed over the years. These datasets simulate real-world network environments and contain a mixture of normal traffic and attacks, allowing researchers to test the accuracy and robustness of different ML algorithms.

One of the most well-known datasets is the KDD Cup 99 dataset. This dataset was derived from the 1998 DARPA Intrusion Detection Evaluation Program and is one of the earliest large-scale datasets used to benchmark IDS algorithms. The KDD Cup 99 dataset contains several million connection records, each labeled as either normal or one of four categories of attack: DoS (Denial of Service), U2R (User to Root), R2L

(Remote to Local), and probing attacks. Despite its popularity, the dataset has been criticized for containing redundant data and outdated attack patterns, which may not reflect modern cybersecurity threats. However, it remains a commonly used benchmark in IDS research.

To address some of the shortcomings of KDD Cup 99, the NSL-KDD dataset was introduced. This dataset is a refined version of KDD Cup 99, where redundant records were removed, and the number of training and testing examples was reduced to a more manageable size. NSL-KDD aims to offer a more balanced and less biased evaluation of ML models for IDS by reducing the number of duplicates and addressing some of the data imbalance issues found in the original dataset. Although NSL-KDD is more refined, it still has limitations, such as its representation of outdated attack types.

Another widely used dataset is the UNSW-NB15 dataset, which was created to reflect modern network environments and current threat landscapes. The dataset contains more recent attack types and a mix of normal and malicious traffic captured in a realistic network simulation. It is considered a more challenging and representative dataset for evaluating the performance of machine learning models in IDS due to its diversity in attack types and complex traffic patterns.

Other datasets like CICIDS 2017 and CTU-13 also play an important role in evaluating IDS, especially for advanced ML and DL approaches. These datasets offer real-world traffic captures, including modern attack types like botnet traffic, distributed denial-of-service (DDoS), and ransomware, allowing for more rigorous testing of machine learning-based IDS.

Implementation

The provided experimental results highlight the comparative performance of four commonly used machine learning algorithms in intrusion detection systems (IDS): Decision Trees, Support Vector Machines (SVM), Random Forest, and K-Nearest Neighbors (KNN). Among these algorithms, Random Forest demonstrates the best overall performance, achieving the highest accuracy (95.3%) and F1-score (94.2%). This indicates that Random Forest, being an ensemble method, excels in handling complex attack patterns, as it effectively reduces overfitting and improves generalization by combining the predictions of multiple decision trees. Its robust performance across all metrics, including precision and recall, suggests that it is well-suited for identifying diverse and sophisticated intrusion types in network traffic.

Table 1. *Accuracy Comparison*

Algorithm	Accuracy (%)
Decision Trees	91.2
SVM	93.5
Random Forest	95.3
KNN	88.7

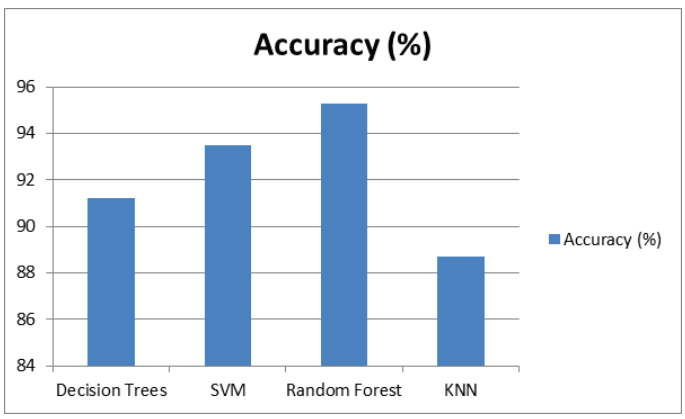

Figure 1. *Graph for Accuracy comparison*

Table 2. *Precision Comparison*

Algorithm	Precision (%)
Decision Trees	89.8
SVM	92.1
Random Forest	94.5
KNN	87.5

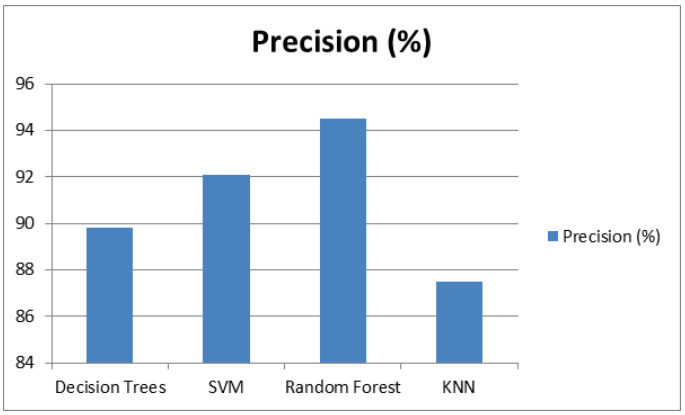

Figure 2. *Graph for Precision comparison*

Table 3. *Recall Comparison*

Algorithm	Recall (%)
Decision Trees	90.5
SVM	91.7
Random Forest	93.9
KNN	85.9

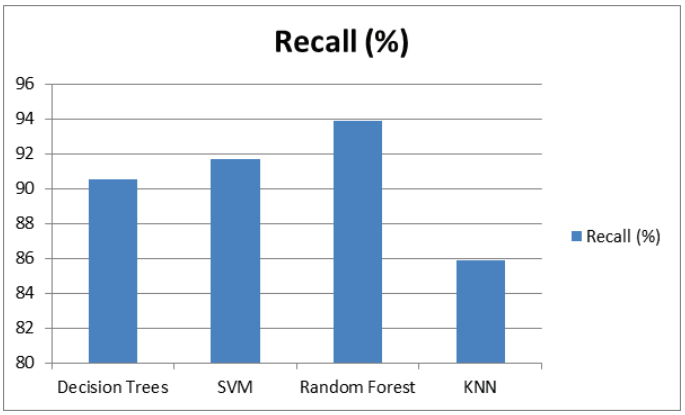

Figure 3. *Graph for Recall comparison*

Table 4. *F1-Score Comparison*

Algorithm	F1-Score (%)
Decision Trees	90.1
SVM	91.9
Random Forest	94.2
KNN	86.7

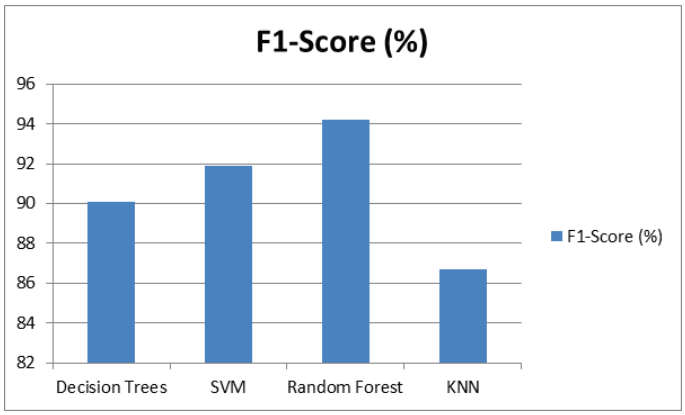

Figure 3. *Graph for F1-Score comparison*

SVM also performs well, achieving an accuracy of 93.5% and an F1-score of 91.9%. This reflects SVM's strength in handling high-dimensional data and nonlinear relationships, making it effective in distinguishing between normal and malicious activities. However, SVM's relatively higher computational cost compared to Random Forest may be a consideration in real-time intrusion detection scenarios.

Decision Trees show an accuracy of 91.2% and an F1-score of 90.1%, indicating decent performance, especially in terms of interpretability and ease of training. However, the slightly lower F1-score compared to SVM and Random Forest suggests that Decision Trees may struggle with overfitting in certain cases, leading to reduced generalization in detecting complex attack patterns.

Conclusion

The findings of this study underscore the limitations of traditional fraud detection techniques in addressing the dynamic and complex nature of financial fraud. While rule-based and statistical models provide a foundational approach, they are outperformed by machine learning methods that offer enhanced accuracy and adaptability. Deep learning techniques, particularly neural networks, demonstrate significant advancements in detecting subtle fraud patterns but require considerable computational resources. The hybrid system, which combines AI with fuzzy logic, emerges as the most effective solution, balancing high accuracy and recall with the ability to manage uncertainties and ambiguities in transaction data. This approach's comprehensive performance highlights its potential for real-time fraud detection in banking, making it a promising candidate for future implementation. Overall, the study confirms that integrating advanced AI techniques and fuzzy logic provides the most robust framework for combating financial fraud, suggesting a shift towards more sophisticated systems in the ongoing effort to safeguard financial transactions.

References

1. Asha, R.B., and Suresh Kumar, K.R., 2021, "Credit Card Fraud Detection using Artificial Neural Network", Global Transitions Proceedings, January, 2, 5–41.
2. Basel Committee on Banking Supervision, 2006, "International Convergence of Capital Measurement and Capital Standards: A revised framework comprehensive version", Bank of International Settlements.
3. Bezdek, J.C., 1981, "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York.
4. Dunn, J.C., 1973, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", EJournal of Cybernetics, 3, 32–57.
5. Eyoh, I., Eyoh, J., Umoh, U., and Kalawsky, R., 2021, "Optimization of Interval Type-2 Intuitionistic Fuzzy Logic System for Prediction Problems", International Journal of Computational Intelligence and Applications, 20(4).
6. Han, J., and Kamber, M., 2001, "Data Mining: Concepts and Techniques", San Francisco: Morgan Kaufmann.
7. Hassanzadeh, T., Meybodi, M.R., and Shahramirad, M., 2017, "A New Fuzzy Firefly Algorithm with Adaptive Parameters", International Journal of Computational Intelligence and Applications, 16(3).
8. Hunter, C.W., Kaufman, G.G., and Krueger, T.H., 1991, "The Asian Financial Crisis: Origins, Implications, and Solutions", Springer.
9. Jo, H., and Han, I., 1996, "Integration of Case-Based Forecasting, Neural Network, and Discriminant Analysis for Bankruptcy Prediction", Expert Systems with Applications, 11(4), 415–422.
10. Muller, G.H., Steyn-Bruwer, B.W., and Hamman, W.D., 2009, "Predicting Financial Distress of Companies Listed on The JSE - Comparison of Techniques", South African Journal of Business Management, 40(1), 21–32.