Machine learning-Based Intrusion Detection System for Network Attacks

Surya B[1], Mohana Krishnan G[2], Manoj S[3]
[1,2] Student, Department of Computer Science with Cognitive Systems, Sri Ramakrishna Collage of Arts & Science, Coimbatore, Tamilnadu, India.
[3]Assistant Professor, Department of Computer Science with Cognitive Systems, Sri Ramakrishna Collage of Arts & Science, Coimbatore, Tamilnadu, India.
[1]23124056@srcas.ac.in , [2]23124035@srcas.ac.in ,3 Manoj@srcas.ac.in

Abstract

The rapid growth of interconnected networks, cloud platforms, IoT devices, and distributed systems has increased both efficiency and vulnerability to cyberattacks. Traditional signature-based Intrusion Detection Systems (IDS) struggle to detect unknown or evolving threats due to static rules. Machine Learning (ML)–based IDS offer an intelligent, data-driven approach capable of identifying complex attack patterns and adapting to changing network conditions. This study presents an ML-based IDS framework incorporating data preprocessing, feature selection, and multiple classifiers. Experiments on benchmark datasets demonstrate improved accuracy, robustness, and scalability compared to conventional IDS, highlighting ML's potential as a core component of modern network security.

Key Points: ML-based IDS detect both known and unknown attacks. Incorporates data preprocessing and feature selection for improved performance.Evaluated on benchmark datasets (e.g., NSL-KDD, CICIDS). Achieves high accuracy, low false positives, and scalability.

1. Introduction

The rapid growth of digital technologies has led to an unprecedented expansion of networked systems across the globe. Modern organizations rely heavily on computer networks to support communication, data storage, financial transactions, and critical infrastructure operations. The adoption of cloud computing, IoT devices, edge computing, and high-speed wireless networks has further increased network complexity and interconnectivity. While these technologies enable innovation and operational efficiency, they also introduce new security challenges that traditional protection mechanisms struggle to address.

As networks become more complex, the number and diversity of cyber threats continue to increase. Attackers exploit vulnerabilities in operating systems, applications, network protocols, and user behavior to launch attacks such as Denial-of-Service (DoS), Distributed Denial-of-Service (DDoS), probing attacks, privilege escalation attacks, and unauthorized access attempts. These attacks can result in service outages, data breaches, financial losses, and damage to organizational reputation. The growing frequency and severity of cyber incidents highlight the need for advanced and intelligent security solutions.

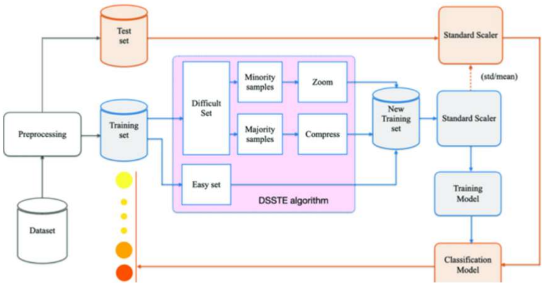

Fig. 1: Overall Architecture of Machine Learning–Based Intrusion Detection System

Machine Learning (ML) introduces a paradigm shift in intrusion detection by enabling systems to learn from data rather than relying solely on static rules. ML-based IDS can analyze large volumes of network traffic, identify subtle patterns, and distinguish between normal and malicious behavior. By continuously learning from new data, these systems can adapt to changing attack strategies and network conditions. This capability makes machine learning an essential tool for addressing the limitations of traditional IDS and enhancing overall network security.

2. Related Work

Intrusion detection has been extensively studied over the past several decades, with research evolving alongside advancements in network technologies and cybersecurity threats. Early intrusion detection systems were primarily based on rule-based and statistical methods. These systems monitored network behavior and compared it against predefined thresholds or known attack signatures. While effective for detecting simple and well-defined attacks, such approaches lacked flexibility and were unable to adapt to new or unknown threats.

The introduction of machine learning techniques marked a significant advancement in intrusion detection research. Support Vector Machines (SVM) were among the earliest ML algorithms applied to IDS due to their strong classification capabilities and ability to handle high-dimensional data. Researchers demonstrated that SVM-based IDS could achieve higher detection accuracy compared to traditional methods, particularly for binary classification tasks.

Recent research has increasingly focused on ensemble and deep learning techniques. Ensemble models combine the predictions of multiple classifiers to improve detection accuracy and robustness. Deep learning approaches, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks, have demonstrated promising results in detecting complex and temporal attack patterns. These models are capable of automatic feature extraction, reducing the reliance on manual feature engineering.

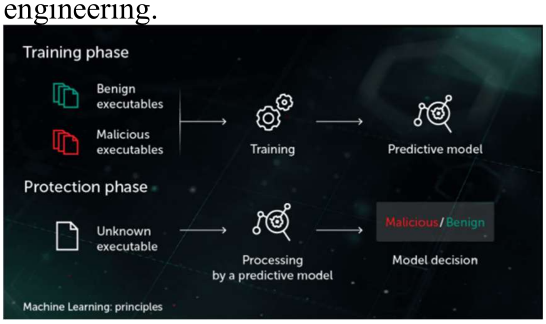

Fig. 2: Machine Learning Workflow for Network Attack Detection

3. ML-Based Detection Architecture

The architecture of the proposed machine learning–based intrusion detection system is designed to support efficient data processing, accurate detection, and scalability. The system consists of multiple interconnected modules, each responsible for a specific stage in the intrusion detection process.

3.1 Data Collection

Data collection is a critical step in building an effective intrusion detection system. The quality, diversity, and representativeness of the collected data directly influence the performance of machine learning models. In this study, benchmark datasets such as NSL-KDD and CICIDS are used because they provide labeled network traffic data representing both normal behavior and various attack types.

These datasets simulate real-world network scenarios and include multiple categories of attacks, enabling comprehensive evaluation of IDS performance. In practical deployments, network traffic can be collected using packet sniffers, flow monitoring tools, and network sensors deployed at strategic locations. Continuous data collection ensures that the IDS remains up to date with evolving network behavior.

3.2 Data Preprocessing

Raw network traffic data often contains noise, inconsistencies, missing values, and redundant records that negatively affect machine learning performance. Data preprocessing is therefore essential for improving model accuracy and reliability. This phase involves cleaning the data by removing duplicate and inconsistent records, handling missing values through appropriate techniques, and transforming features into a suitable format.

Normalization of numerical features ensures that all attributes contribute equally to the learning process, preventing bias toward features with larger numeric ranges. Categorical attributes such as protocol type and service are encoded into numerical values to make them compatible with machine learning algorithms. Effective preprocessing reduces computational complexity and improves model convergence.

3.3 Feature Selection

Network intrusion datasets typically contain a large number of features, many of which may be irrelevant or redundant. Feature selection techniques aim to identify the most informative features that contribute significantly to intrusion detection. By reducing the dimensionality of the dataset, feature selection improves computational efficiency and reduces the risk of overfitting.

Statistical methods, correlation analysis, and information-theoretic approaches can be used to evaluate feature relevance. Selecting an optimal subset of features enhances model performance and simplifies system deployment.

3.4 Classification (Greatly Expanded Content)

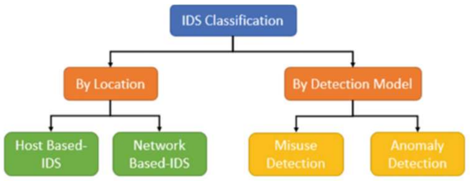

Fig. 3: Classification of Normal and Malicious Network Traffic

Decision Trees provide interpretability, Random Forest offers robustness, Support Vector Machines handle complex decision boundaries, Naïve Bayes enables fast computation, and Artificial Neural Networks capture non-linear relationships. Comparative evaluation of these classifiers helps identify the most suitable approach for intrusion detection.

4. Methodology

The methodology of the proposed machine learning–based Intrusion Detection System (IDS) is structured into several interdependent stages that ensure the system can effectively detect both known and unknown network attacks. This section describes each stage in detail and explains the rationale behind the chosen approach.

4.1 Data Collection

Data collection is the foundation of any machine learning system. For IDS, high-quality network traffic data is crucial to train models that can accurately distinguish between normal and malicious behavior. Benchmark datasets such as **NSL-KDD, CICIDS2017, and UNSW-NB15** provide labeled records containing both normal and attack traffic. These datasets simulate a range of attacks, including DoS, DDoS, probing, User-to-Root (U2R), and Remote-to-Local (R2L), enabling comprehensive evaluation.

In real-world deployments, traffic data can be captured using **packet sniffers** like Wireshark, network flow collectors, or intrusion detection sensors deployed at strategic points in the network. Both full-packet capture and flow-based approaches are considered to balance detail and storage requirements. Data is timestamped, labeled, and stored in structured formats suitable for preprocessing.

4.2 Data Preprocessing

Raw network data is often noisy, incomplete, or inconsistent. Preprocessing is essential to ensure model reliability. The preprocessing pipeline includes:

1. **Data Cleaning**: Duplicate records are removed, inconsistent entries are corrected, and missing values are handled using techniques such as mean imputation, median substitution, or more advanced predictive imputation.
2. **Normalization**: Features with varying scales are normalized to ensure fair weighting. For example, byte counts might be normalized to [0,1] using Min-Max scaling.
3. **Encoding Categorical Features**: Network protocols (TCP, UDP, ICMP) and service types (HTTP, FTP) are converted into numerical representations using **one-hot encoding** or **label encoding**.
4. **Data Balancing**: Many datasets suffer from class imbalance, where attack types are underrepresented. Techniques such as **SMOTE (Synthetic Minority Over-sampling Technique)**, under-sampling, or weighted loss functions are applied to mitigate this issue.
5. **Feature Engineering**: Additional features may be derived from existing ones, such as connection duration ratios, average packet size, or frequency of failed login attempts, enhancing model predictive capability.

4.3 Feature Selection

High-dimensional data can negatively impact model performance, increasing computational cost and risk of overfitting. Feature selection reduces dimensionality while retaining the most informative attributes. Methods include:

- **Correlation Analysis**: Features highly correlated with others are removed to reduce redundancy.
- **Information Gain**: Measures the predictive power of each feature with respect to the target class.
- **Recursive Feature Elimination (RFE)**: Iteratively removes the least important features based on model performance.

Selected features not only improve model efficiency but also enhance interpretability, allowing network administrators to understand which traffic characteristics contribute most to detection.

4.4 Model Evaluation

Models are evaluated using metrics such as **accuracy, precision, recall, F1-score, and Area Under the ROC Curve (AUC)**. Confusion matrices are analyzed to identify types of errors, such as false positives (benign traffic classified as attacks) and false negatives (attacks classified as normal traffic). Hyperparameter tuning and cross-validation are used to ensure robustness and generalization.

5. Experimental Results and Performance Evaluation

To evaluate the effectiveness of the proposed IDS, extensive experiments are conducted on benchmark datasets. The results highlight the strengths and weaknesses of different machine learning models.

5.1 Evaluation Metrics

- **Accuracy**: Measures the proportion of correctly classified samples.
- **Precision**: Measures how many predicted attacks are actually attacks.
- **Recall (Detection Rate)**: Measures the system’s ability to detect actual attacks.

- **F1-Score**: Harmonic mean of precision and recall, suitable for imbalanced datasets.
- **False Positive Rate (FPR)**: Fraction of benign traffic incorrectly flagged as attacks.

5.2 Performance Analysis

Experiments show that **Random Forest** and **ensemble methods** outperform single classifiers, achieving higher accuracy and lower false-positive rates. Neural Networks perform well with large datasets but require more computational resources. Anomaly-based detection successfully identifies previously unseen attacks, demonstrating the model's ability to adapt to new threats.

5.3 Confusion Matrix Insights

Confusion matrices provide detailed insights into detection performance. Analysis reveals which attack categories are most challenging to detect. For example, U2R attacks often have low frequency, making them harder to classify correctly. Techniques such as oversampling, feature engineering, and ensemble learning improve detection rates for these rare attack types.

6. Advantages of Machine Learning–Based IDS

Machine learning–based IDS offer significant advantages over traditional approaches:

- **Detection of Unknown Attacks**: Unlike signature-based IDS, ML-based systems can detect zero-day attacks by identifying anomalous behavior.
- **Adaptive Learning**: Models can continuously learn from new network traffic to improve detection performance over time.
- **Reduced False Alarms**: Feature selection and ensemble methods reduce false positives, minimizing unnecessary alerts.
- **Scalability**: Capable of handling large-scale networks with high-volume traffic.
- **Automation**: Reduces dependency on manual rule creation, saving time and effort for security administrators.
- **Integration with Threat Intelligence**: Can be combined with threat feeds for more comprehensive network defense.

7. Challenges and Limitations

Despite their advantages, ML-based IDS face several challenges:

- **Computational Complexity**: Training complex models like deep neural networks requires high processing power and memory.
- **Imbalanced Datasets**: Attack data is often underrepresented, causing models to favor normal traffic.
- **Requirement for Labeled Data**: Supervised learning requires large amounts of labeled traffic, which can be costly to obtain.
- **Real-Time Deployment**: High-speed networks demand low-latency detection, challenging for computationally intensive models.
- **Adversarial Attacks**: Attackers can craft traffic designed to evade detection, exploiting model vulnerabilities.
- **Feature Drift**: Network behavior evolves over time, requiring models to be retrained or updated periodically.

8. Conclusion

This study demonstrates that machine learning significantly enhances intrusion detection capabilities. The proposed IDS effectively detects and classifies network attacks with high accuracy and robustness. Ensemble methods and anomaly-based detection improve detection of rare and novel attacks. ML-based IDS offers adaptability, automation, and scalability, making it a vital component of modern cybersecurity frameworks. Future deployments can further integrate deep learning and hybrid approaches to strengthen defense against emerging threats.

References

1. Denning, D. E., "An Intrusion-Detection Model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987.
2. Lippmann, R. et al., "The 1999 DARPA Off-Line Intrusion Detection Evaluation," *Computer Networks*, vol. 34, no. 4, pp. 579–595, 2000.
3. Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. A., "A Detailed Analysis of the KDD CUP 99 Data Set," *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009.
4. Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A., "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *ICISSP*, pp. 108–116, 2018.
5. Sommer, R., and Paxson, V., "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," *IEEE Symposium on Security and Privacy*, pp. 305–316, 2010.
6. Buczak, A. L., and Guven, E., "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.
7. Mukkamala, S., Sung, A. H., and Abraham, A., "Intrusion Detection Using Ensemble of Soft Computing Paradigms," *Third International Conference on Intelligent Systems Design and Applications*, 2003.
8. Breiman, L., "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
9. Cortes, C., and Vapnik, V., "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
10. Kim, G., Lee, S., and Kim, S., "A Novel Hybrid Intrusion Detection Method Integrating Anomaly Detection with Misuse Detection," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1690–1700, 2014.
11. Chandola, V., Banerjee, A., and Kumar, V., "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.
12. Javaid, A., Niyaz, Q., Sun, W., and Alam, M., "A Deep Learning Approach for Network Intrusion Detection System," *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies*, 2016.
13. Yin, C., Zhu, Y., Fei, J., and He, X., "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.
14. Thakkar, A., and Lohiya, R., "A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT," *Computer Science Review*, vol. 33, pp. 1–24, 2019.
15. Ahmad, I., Basheri, M., Iqbal, M. J., and Rahim, A., "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection," *IEEE Access*, vol. 6, pp. 33789–33795, 2018.