**Cyber Intrusion Detection System Using Machine Learning**

 **P.Meenaroshini1.,A.Siva Ganesh2.,**

1Department of MCA,

2Assistant Professor, Department of MCA,

1,2Mepco Schlenk Engineering College, Sivakasi– 626124

1meenaroshini38\_mca25@mepcoeng.ac.in.,

 2sivaganesha@mepcoeng.ac.in

**Abstract**

In today’s digital landscape, securing computer networks against sophisticated cyberattacks has become increasingly critical. Traditional Intrusion Detection Systems (IDS) often fall short in identifying novel and evolving threats due to their reliance on static rules and known attack signatures. To address these limitations, this research introduces a Machine Learning-based Cyber Intrusion Detection System (CIDS-ML) that enhances detection accuracy and adapts to changing attack patterns. The system follows a structured workflow that includes data collection, preprocessing, feature selection, model training, evaluation, and real-time deployment. Utilizing the KDD Cup 1999 dataset, the system classifies network traffic as either normal or malicious using multiple classification algorithms, including Random Forest, K-Nearest Neighbors, Decision Tree, Support Vector Machine, and Logistic Regression. Real-time monitoring is achieved through a Flask-based web application, whileattack trends and detectionperformance are visualized using web technologies and Matplotlib. The hybrid detection approach integrating both signature-based and anomaly-based techniques demonstrates improved accuracy, scalability, and real-time applicability in modern network environments.

**Keywords**: Network traffic, machine learning, cybersecurity, anomaly detection, flask, and intrusion detection systems.

**1. Introduction**

Protecting information systems from malevolent breaches is more important than ever due to the exponential rise of digital communication and the proliferation of devices connected to networks. Zero-day attacks and changing threat patterns are difficult for traditional IDS techniques to detect because they mostly rely on static rules and predetermined signatures. Intelligent, adaptable systems that can identify strange or unusual behavior in real time are desperately needed as cyberattacks get more complex.

Machine learning (ML) offers powerful capabilities to overcome the limitations of conventional intrusion detection systems by analyzing network traffic, identifying patterns, and predicting assaults using historical data. This research proposes a cyber intrusion detection system based on machine learning (ML) that employs classification algorithms to reliably distinguish between malicious and normal traffic. The solution includes preprocessing, evaluation, supervised learning, and real-time deployment using a user-friendly dashboard and Flask framework.

**1.1 Data Preprocessing and Feature Extraction**

The KDD Cup 1999 dataset, a commonly recognized standard in intrusion detection research, is used by the system. Cleaning, dealing with missing values, label encoding to convert categorical variables to numerical values, and normalizing numerical features to guarantee consistency are all examples of data preprocessing.

 In order to decrease dimensionality and improve model performance, pertinent characteristics are chosen using correlation analysis and domain knowledge. Heatmaps are used to depict feature importance in order to find and preserve important qualities.

**1.2 Model Selection, Training, and Evaluation**

Using the Scikit-learn library, the system uses supervised machine learning methods such as K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Support Vector Machine (SVM), and Logistic Regression. Using labeled features from the KDD Cup 1999 dataset, these models are trained to categorize network traffic records into harmful or normal groups.

 To provide a fair assessment of classification performance, each model is assessed using important performance parameters like accuracy, precision, recall, and F1-score. During training, K-Fold Cross-Validation is used to avoid overfitting and confirm the models' generalizability. A solid selection of the top-performing algorithm Random Forest that showed the highest accuracy and stability for deployment in real-time scenarios was made possible by the evaluation technique, which provides a trustworthy assessment of the model's performance across various data subsets.

**1.3 Real-Time Intrusion Detection and Monitoring**

The suggested approach incorporates a trained Machine Learning model into a Flask-based web application for real-time intrusion detection. Users can upload network traffic datasets in CSV format for bulk scanning or manually enter network feature values using this interface. Following submission, the system analyzes the data and classifies each record as either harmful or benign using a trained model.

 Instantaneous predictions are supported by the application, and the dashboard shows the findings immediately. It offers thorough visual feedback by displaying the distribution of identified assaults using charts and graphs. By combining bulk scanning and real-time detection, the solution allows for proactive network activity monitoring and guarantees prompt detection of questionable activity.

 The Flask framework's lightweight architecture guarantees seamless interaction and speedy deployment, making it appropriate for real-world settings where cybersecurity requires swift decision-making.

**1.4 Deployment, Monitoring, and Reporting**

The Flask web application, which is used to deploy the intrusion detection system, makes it simple to integrate into network environments and requires little setup. The program, which can be hosted locally or on a cloud server, provides bulk CSV-based attack scanning and easily available real-time prediction services.

A visually appealing dashboard created with HTML, CSS, Bootstrap, and Matplotlib is part of the system. The types and frequency of detected assaults are displayed via interactive charts on this dashboard, including pie charts and bar graphs. Users can rapidly understand infiltration trends and patterns with the aid of these visuals.

The program records every scan session for monitoring, including timestamps and outcomes, which can then be examined for audit and analysis reasons. Fast response times are guaranteed by the lightweight deployment, and the interface supports a variety of reporting formats, making it appropriate for enterprise, academic, and security operations.

**2. Literature Survey**

 [1]The application of machine learning approaches to identify intrusions in sizable and unbalanced network datasets is investigated in this paper. It suggests balanced training methods and improved preprocessing tactics that improve IDS accuracy in practical settings, especially when handling class imbalance.

[2]A lightweight feature selection method that reduces computational costs without sacrificing detection performance is presented in this work. Because the system incorporates several reliable machine learning techniques, it can be used in contexts with limited resources while yet being scalable and effective.

[3]The study examines various machine learning techniques for identifying anomalous activity in network data with an emphasis on anomaly-based intrusion detection systems. In order to increase detection accuracy, it emphasizes how crucial feature engineering, data quality, and algorithm selection.

[4]Decision trees, SVMs, and ensemble models are among the machine learning techniques that are compared in this study for IDS. It provides information about these algorithms' advantages and disadvantages for intrusion detection tasks by analyzing how well they perform on well-known datasets such as KDD Cup 1999 and NSL-KDD.

[5]Random Forests and Gradient Boosting Machines are two of the ensemble models that are experimentally compared. The models have demonstrated enhanced efficacy in managing unbalanced datasets and identifying a diverse range of assaults, rendering them appropriate for practical implementation.

[6]A comparison of AI-based intrusion detection systems for critical infrastructure is presented in this research. It highlights the necessity of dependable, low-latency detection systems that can manage a variety of cyberthreats in high-stakes situations. The work also demonstrates how conventional machine learning models can be incorporated into these kinds of systems.

[7]This survey explores the use of static analysis combined with ML classifiers to detect malicious code. It discusses the advantages of ML-based systems over traditional signature-based methods, particularly for detecting new or previously unknown threats.

[8]A modular ensemble framework using one-class classifiers is proposed for detecting intrusions in network traffic. The system excels in detecting unknown attacks with minimal false positives, making it a promising solution for real-time and zero-day attack detection in network environments.

[9]Focused on Internet of Things (IoT) systems, this paper surveys ML-based IDS methods tailored for constrained devices and decentralized networks. It identifies challenges in detecting attacks in the IoT ecosystem and suggests solutions based on traditional machine learning techniques.

[10]This study integrates behavioural modeling with adaptive IDS to detect deviations in network traffic patterns. The ML-based approach allows the system to evolve as network behaviors change, improving detection capabilities over time.

[11]An extensive review covers emerging trends in hybrid IDS models that combine anomaly-based detection with signature-based methods. These hybrid systems leverage ML to improve detection accuracy and adapt to evolving threats in dynamic network environments.

 [12]An integrated network traffic filtering system is proposed that detects both known and unknown threats using a hybrid ML approach. By combining rule-based systems with behavioral analysis, the system improves threat visibility and detection performance.

[13]This paper introduces Andromaly, an adaptive IDS for Android that detects malicious activities by monitoring device behavior. The system learns from user patterns over time and adapts to distinguish between benign and suspicious activities using ML-based classifiers.

[14]This study integrates SELinux-based intrusion prevention mechanisms with traditional IDS to enhance mobile device security. It uses policy-driven security measures along with machine learning techniques to better detect unauthorized access and mitigate potential threats.

[15]An ML-based approach is introduced to detect unknown worms by classifying host behavior. The model identifies deviations from typical behavior patterns, enabling detection of polymorphic or zero-day malware threats.

[16]This work uses network analysis to identify groups in network traffic that may correspond to coordinated attacks. It aids in tracing intrusion sources and understanding attack strategies, with ML techniques playing a key role in recognizing these patterns.

[17]The study explores the performance of large-scale traffic filtering in scrubbing centers. It raises concerns about latency and performance degradation while suggesting that intelligent traffic inspection strategies, powered by ML, could improve the system’s effectiveness.

[18]This paper provides a security assessment of Android, identifying vulnerabilities and proposing mitigation strategies using IDS. It lays the foundation for building machine learning-based intrusion detection tailored to mobile platforms.

[19]This paper introduces a temporal abstraction method for intrusion detection on mobile devices. By analyzing temporal patterns of user activity, it enables the real-time detection of both known and emerging threats, relying on machine learning techniques for pattern recognition.

**3. Proposed Work**

This research presents the design and implementation of a Machine Learning-based Intrusion Detection System (IDS) aimed at detecting and classifying a wide range of cyberattacks in real-time or near-real-time. The core objective is to strengthen network security by accurately identifying malicious activities, unauthorized access attempts, and policy violations, enabling swift and informed responses to potential threats.

The proposed system leverages multiple supervised learning algorithms to build a robust and scalable detection engine. Emphasis is placed on effective data preprocessing, strategic feature selection, and thorough model evaluation to ensure high accuracy and reliability. The system integrates both signature-based and anomaly-based detection techniques to improve its capability in identifying known attack patterns and emerging threats.

A lightweight web-based interface is developed using Flask for real-time monitoring, while data visualization and reporting components are built using HTML, CSS, Bootstrap, and Matplotlib to enhance usability and situational awareness.

**3.1 Data Collection and Preprocessing**

The main source of raw network traffic statistics is the KDD Cup 1999 dataset. To guarantee uniformity and enhance the performance of the machine learning models, this data goes through a thorough preparation phase that includes handling missing values, eliminating noise, and normalizing features.



**3.2 Feature Extraction and Selection**

Relevant features are extracted using domain knowledge and correlation-based analysis. Feature selection techniques, including heatmap visualization, are employed to reduce dimensionality while retaining essential characteristics.





**3.3 Data Splitting**

The dataset is divided into three subsets: training, testing, and validation. The training set is used to train the machine learning models, while the testing set is reserved for evaluating model performance. A separate validation set is used for fine-tuning model hyperparameters and ensuring the generalizability of the mode.



**3.4 Model Training**

Several supervised machine learning algorithms are trained

**Logistic Regression:**

It is a binary classification algorithm used to predict the probability of a binary outcome. It models the relationship between a dependent variable and one or more independent variables by estimating probabilities using the logistic function.

**K-Nearest Neighbor's:**

K-Nearest Neighbors is a simple and intuitive classification algorithm used for both classification and regression tasks. The value of K is a hyperparameter that needs to be set before training the model. KNN is effective for data with non-linear decision boundaries.

**Decision Trees:​**​

Decision Trees are a non-linear supervised learning algorithm used for both classification and regression tasks. It creates a tree-like model where each internal node represents a decision based on a feature, each branch represents an outcome of the decision, and each leaf node represents a class label or a value.

**Support Vector Machine (SVM):​**​

Support Vector Machine is a powerful supervised learning algorithm used for classification and regression tasks. SVM tries to find the optimal hyperplane that best separates data points of different classes in the feature space.

**Random Forest:​**​

Random Forest is an ensemble learning method that combines multiple decision trees to improve the accuracy and reduce overfitting. Each tree in the forest is trained on a random subset of the data and a random subset of features. The final prediction is made by averaging the predictions of all the individual trees in the forest.

 **3.5 Model Evaluation**

The trained models were evaluated using accuracy, precision, recall, and F1-score on the KDD Cup 1999 dataset. Decision Tree and Random Forest achieved the highest accuracy, effectively classifying attack types like DoS, Probe, U2R, and R2L. Logistic Regression and SVM showed moderate performance, while KNN was slightly less accurate due to sensitivity to data distribution. Overall, tree-based models proved most reliable for real-time intrusion detection.

**3.6 Real-Time Monitoring and Detection**

The best-performing model is integrated into a real-time monitoring system using Flask. The model continuously analyzes incoming network traffic and flags potential intrusions.

**3.7 Deployment**

To assess real-time performance and response capabilities, the completed intrusion detection system is set up in a network environment simulation. A Flask-based web framework incorporates the learned machine learning model, allowing for ongoing network intrusion detection and monitoring. To test the system under real-world circumstances, simulated traffic is created, including both controlled attacks and typical behavior. Both real-time and batch detection modes are supported by the lightweight, modular design, which also guarantees seamless connection with current infrastructure.

**3.8 Visualization and Reporting**

Interactive dashboards and graphical reports are developed using HTML, CSS, Bootstrap, and Matplotlib to provide insights into intrusion detection performance and attack trends. The proposed IDS incorporates both signature-based and anomaly-based detection techniques. Signature-based detection matches predefined attack patterns, while anomaly-based detection identifies deviations from normal behavior. This hybrid approach improves detection accuracy and minimizes false positives.

**3.9 Advantages**

Utilizes supervised machine learning algorithms that effectively detect and classify complex attack types such **as DoS, Probe, U2R, and R2L**, enhancing overall threat identification. Integrates real-time data analysis using Flask, enabling **instant detection of malicious activities** and **faster response** to threats. Designed with a **modular architecture** that supports both batch and real-time detection, making it scalable for small to large network environments. Incorporates **Firebase authentication** to ensure secure and authorized access, reducing risks associated with unauthorized users. Provides a lightweight **web-based dashboard** using HTML, CSS, Bootstrap, and Matplotlib for easy monitoring and visualization of system status and detected threats. Built using open-source technologies like **Flask and Firebase**, making it suitable for academic use and organizations with limited budgets. Enables better decision-making through **graphical reports**, highlighting attack trends and performance metrics.

**4. Results**

****

**Fig. 4.1 Dashboard**

The Real-time Dashboard Module offers ongoing network activity monitoring and presents system performance, identified threats, and intrusion patterns in an intuitive user interface. Security professionals can swiftly spot irregularities and take appropriate action to it. By guaranteeing prompt response to important threats, integrated alert messages improve the accuracy, usability, and overall security posture of the Cyber Intrusion Detection System.

****

**Fig. 4.2 Prediction Page**

**[ ATTACK DETECTED ]**

****

**Fig. 4.3prediction Page[ NORMAL ATTACK]**

The Prediction Module analyzes incoming network data using the trained machine learning model. Quick reaction to security threats is made possible by its real-time analysis, threat classification, and instant notifications for suspicious activities.

****





**Fig. 4.4 Scan Page**

This module allows users to upload network log files for offline analysis. It processes the data, applies the trained machine learning model, and detects potential threats. It supports post-attack investigation and helps identify intrusion patterns, enhancing the system’s overall security posture.

**5. Conclusion**

A scalable and effective method for detecting and categorizing intrusions is the Cyber Intrusion Detection System using Machine Learning (CIDS-ML). Using supervised machine learning methods, the system successfully identifies attacks such as DoS, Probe, U2R, and R2L on the KDD Cup 99 dataset. Usability and security are improved via secure user authentication through Firebase, real-time monitoring, and an interactive dashboard. CIDS-ML provides a proactive approach to contemporary network security while guaranteeing data integrity through role-based access and encryption.

**References**

**[1]**Zhang, Y., Bai, Y., et al. (2024). Machine learning-based network intrusion detection for big and imbalanced data. Journal of Big Data.

**[2]**Sharma, S. K., Mishra, A. K., et al. (2025). Robust machine learning-based intrusion detection system using lightweight feature selection. Scientific Reports.

**[3]**Kimanzi, R., Kimanga, P., et al. (2024). Deep learning algorithms used in intrusion detection systems – A review. arXiv preprint.

**[4]**Maseer, Z. K., Yusof, R., et al. (2023). Meta-analysis and systematic review for anomaly network intrusion detection systems: Detection methods, dataset, validation methodology, and challenges. arXiv preprint.

**[5]**Tait, K.-A., Khan, J. S., et al. (2021). techniques: An experimental comparison. arXiv preprint.

**[6]**Otoum, S., Kantarci, B., &Mouftah, H. (2020). A comparative study of AI-based intrusion detection techniques in critical infrastructures. arXiv preprint.

**[7]**Elovici, Y., Shabtai, A., et al. (2009). Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey. Information Security Technical Report.

**[8]**Giacinto, G., Perdisci, R., et al. (2008). Intrusion detection in computer networks by a modular ensemble of one-class classifiers. Information Fusion.

**[9]**Kumar, S., "Network Security and Intrusion Detection: A Practical Guide," Springer, 2022.

**[10]**Zhang, B., Yin, J., et al. (2007). Autonomic and trusted computing. Lecture Notes in Computer Science.

**[11]**Locasto, M. E., Wang, K., et al. (2005). Recent advances in intrusion detection. Lecture Notes in Computer Science.

**[12]**Shabtai, A., Potashnik, D., et al. (2011). Monitoring, analysis, and filtering system for purifying network traffic of known and unknown malicious content. Security and Communication Networks.

**[13]**Shabtai, A., Kanonov, U., et al. (2012). Andromaly: a behavioral malware detection framework for Android devices. Journal of Intelligent Information Systems.

**[14]**Shabtai, A., Fledel, Y., et al. (2010). Securing Android-powered mobile devices using SELinux. IEEE Security & Privacy.

**[15]**Sonmez, M., "Learning Firebase," Packt Publishing, 2023.

**[16]**Tang, D., "Cybersecurity Threats and Intrusion Detection," CRC Press, 2023.

**[17]**Zilberman, P., Puzis, R., &Elovici, Y. (2017). On network footprint of traffic inspection and filtering at global scrubbing centers. IEEE Transactions on Dependable and Secure Computing.

**[18]**Shabtai, A., Fledel, Y., et al. (2010). Google Android: A comprehensive security assessment. IEEE Security & Privacy.

**[19]**Shabtai, A., Kanonov, U., et al. (2010). Intrusion detection for mobile devices using the knowledge-based, temporal abstraction method. Journal of Systems and Software.