International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

Interruption Identification System Using Machine Learning

Karan Mahesh Gangwani[1], Utkarsh Mishra[2], Dr. Sreekumar K[3], Dr. Sibi Amaran[4]

[1]Student, Department of ComputingTechnologies, College of Engineering and Technology, Faculty of Engineering and Technology, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur, 603203, TN, India
Email: *kg4727[at]srmist.edu.in*

[2]Student, Department of ComputingTechnologies, College of Engineering and Technology, Faculty of Engineering and Technology, SRM Institute of Science and Technology, SRM, Nagar, Kattankulathur,603203, TN, India
Email: *ub0731[at]srmist.edu.in*

[3]Associate Professor, Department of ComputingTechnologies, College of Engineering and Technology, Faculty of Engineering and Technology, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur,603203, TN, India
Email: *sibiamaa[at]srmist.edu.in*

[4]Assistant Professor, Department of ComputingTechnologies, College of Engineering and Technology, Faculty of Engineering and Technology, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur, 603203, TN, India
Email: *sreekumk[at]srmist.edu.in*

Abstract: ***Intrusion Detection Systems (IDS) are essential for safeguarding network infrastructures by continuously monitoring and detecting potential security threats. However, traditional approaches struggle to match the growing sophistication of modern attacks, prompting the need for advanced methodologies. This study presents a machine learning-driven IDS framework utilizing Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) algorithms. The process includes steps such as data collection, preprocessing, feature extraction, model training, and performance assessment. Experimental findings reveal that the proposed system achieves notable accuracy, with SVM and KNN models attaining approximately 97% and 99% accuracy, respectively. These results underscore the promise of machine learning in improving IDS performance and fortifying network security.***

Keywords: Intrusion Detection System, Machine Learning, SVM, KNN, Network Security, Cybersecurity, Feature Extraction, Model Training

1. Introduction

Motivation

With the rapid expansion of internet-connected devices and cloud computing, the potential for cyber threats has significantly increased. Traditional Intrusion Detection Systems (IDS), which rely on predefined signatures and basic anomaly detection, are increasingly inadequate against today's complex attacks. These systems often produce high false positive rates, leading to excessive alerts, and struggle to identify new threats, resulting in dangerous false negatives. This gap underscores the need for detection mechanisms that are more adaptive, intelligent, and precise in countering evolving cyber risks.

Objective

The primary goal of this research is to create and implement an Intrusion Detection System (IDS) that utilizes machine learning methods—specifically, Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) algorithms—to improve the detection accuracy and efficiency of network security systems. The framework aims to accurately identify and classify a broad spectrum of network intrusions, encompassing both known and emerging threats, while reducing false positives and false negatives. This IDS is designed for real-time threat detection, enabling network administrators to respond swiftly to potential breaches.

Another core objective is to make the IDS scalable and adaptable across various network settings, from small business networks to extensive cloud infrastructures. The research focuses on optimizing the machine learning models for high performance in real-time, balancing accuracy with computational efficiency. By fulfilling these objectives, this project aspires to support the development of stronger, more dependable cybersecurity solutions capable of protecting vital digital assets in a complex and increasingly hostile cyber landscape.

Problem Statement

Existing IDS solutions frequently struggle to detect new or evolving threats because they depend on predefined signatures or basic anomaly detection techniques. These limitations result in high false positive rates and insufficient detection of complex attacks. An urgent need exists for an IDS that can adapt to emerging attack patterns and deliver dependable, real-time detection.

Challenges

Developing a machine learning-based IDS involves several challenges:

- **Data Quality**: Ensuring the training dataset accurately reflects real-world network traffic.
- **Feature Selection**: Choosing the most relevant features to enhance model performance.
- **Computational Complexity**: Balancing model accuracy with the need for efficient real-time processing.
- **Scalability**: Ensuring that the IDS can operate effectively in large-scale network environments without compromising performance.

2. Related Works

Literature Review

Overview of Intrusion Detection Systems (IDS)
Intrusion Detection Systems (IDS) have long been fundamental to network security, acting as a primary defense against unauthorized access and cyber threats. Traditional IDS methods are generally divided into two types: signature-based and anomaly-based. Signature-based IDS detects intrusions by matching network activity with predefined threat patterns or "signatures." Although effective for recognizing known threats, it is limited in identifying new or evolving attacks that lack established signatures. Anomaly-based IDS, in contrast, examines network traffic for unusual behaviors that deviate from normal patterns, allowing it to detect previously unknown threats. However, this approach often produces a high rate of false positives, as legitimate but uncommon activities may trigger alerts.

Evolution of IDS with Machine Learning
The constraints of traditional IDS have driven research toward more adaptive solutions, resulting in the integration of machine learning (ML) techniques into IDS frameworks. ML allows these systems to learn from historical data, recognize patterns, and make decisions with minimal human input. A primary benefit of applying ML in IDS is its capability to generalize from known attack data, enabling it to detect new, previously unknown threats. Numerous machine learning algorithms have been studied within IDS, each offering distinct advantages and limitations.

Machine Learning Models in IDS
Various machine learning models have become prominent in IDS research. Among them, Support Vector Machine (SVM) is well-studied for its effectiveness in handling high-dimensional data. SVM is especially suited for binary classification tasks, as it identifies the optimal hyperplane to separate different data classes with the greatest margin. Applied in IDS, SVM has shown strong performance in distinguishing between normal network traffic and malicious activities, achieving high accuracy in numerous studies

Hybrid Approaches and Ensemble Learning
Beyond standalone models like SVM and KNN, IDS research has shown substantial interest in hybrid approaches and ensemble learning. Hybrid models integrate multiple machine learning techniques to capitalize on each model's strengths, often resulting in higher detection accuracy and fewer false positives. For instance, pairing SVM with Decision Trees or Neural Networks can improve the model's ability to manage complex, non-linear patterns in network data.

Comparison

RefNo.	Title & Authors Name	Concept Used	Advantage	Disadvantages
1.	**"Intrusion Detection Using SVM and KNN"** M. Al-Qatf, Y. Lasheng, M. Al-Habib, K. Al- Sabahi	SVM combined with KNN for classification	• High accuracy in binary classification • Effective in high-dimensional spaces	• Computationally expensive • Requires careful tuning of parameters
2.	**"Deep Learning Approaches for IDS"** P. Tao, Z. Sun, Z. Sun	Deep Learning (Autoencoders, CNN) for anomaly detection	• High capability to detect complex patterns • Can handle large datasets	• Requires extensive computational resources • Less interpretable
3.	**"Hybrid IDS using Decision Trees and SVM"** S. Subbiah, K. S., M. Anbananthen, S. Thangaraj, S. Kannan, D. Chelliah	Hybrid model Combining Decision Trees and SVM	• Reduces false positives • Combines strengths of both models	• Increased complexity • Higher computational • cost
4.	**"Ensemble Learning for IDS"** P. L. S. Jayalaxmi, R. Saha, G. Kumar, M. Conti, T. -H. Kim	Bagging and Boosting with multiple classifiers	• Improved detection accuracy • Robust to overfitting	• Complexity in model implementation • Can be slow in real-time • processing

Contextualization
This table presents a comparison of various machine learning models and hybrid approaches employed in Intrusion Detection Systems (IDS). It outlines the strengths and limitations of each method, focusing on aspects like detection accuracy, computational demands, and suitability for real-time applications. This comparison offers insights into how these models, whether used individually or combined, can contribute to creating more resilient and effective IDS solutions.

3. Proposed Work

System Architecture Overview
The proposed IDS framework is designed to integrate SVM and KNN algorithms into a cohesive detection system. The architecture is composed of several key modules:

Data Ingestion Module: This module collects and preprocesses network traffic data, ensuring it is suitable for analysis.

Feature Extraction Module: Critical features,such as IP addresses, protocol types, and packet sizes, are extracted and normalized to reduce computational complexity.

Classification Module: This module implementsthe SVM and KNN algorithms to classify networktraffic as normal or malicious.

Alert Generation Module: Based on the classification results, this module generates alerts, providing network administrators with actionableintelligence.

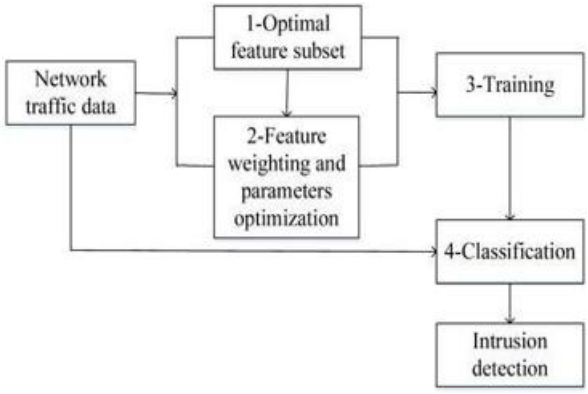
Programming Languages and Tools
The primary programming language used in this project is

Python, due to its extensive libraries and support for machine learning and data processing tasks. Key libraries include:

Scikit-learn: Used for implementing the SVM and KNN algorithms.

Pandas and NumPy: Utilized for data manipulation and preprocessing.

Matplotlib: Employed for visualizing data and results



Module Descriptions

Data Pre-processing: The NSL-KDD dataset is pre-processed to ensure compatibility with machine learning models. This includes converting categorical data to numerical values, scaling features, and handling missing data.

Model Training: The dataset is split into training and testing subsets. Hyper parameter tuning is conducted using grid search and cross-validation to optimize model performance.

Model Evaluation: The trained models are evaluated based on accuracy, precision, recall, and F1-score. Confusion matrices and ROC curves are used to assess performance.

Website Interface for Intrusion Detection System

The website interface for the Intrusion Detection System (IDS) is crafted to provide a user-friendly experience, enabling network administrators to monitor security and manage detected threats effectively. Developed with HTML5, CSS3, and JavaScript (React.js) for the front end and Flask (Python) for the back end, the interface integrates smoothly with machine learning models like SVM and KNN for real-time threat detection and analysis. Key features include a dashboard for visualizing network traffic and alerts, detailed intrusion analysis with filtering options, tools for model evaluation and tuning, real-time alerts categorized by severity, interactive traffic monitoring charts, and user management with access controls. A RESTful API enables communication between the front and back ends, supported by an SQL database that stores network data, user logs, and model performance metrics. The interface also allows users to generate reports, view performance metrics, and configure settings to optimize detection accuracy.

Database Integration

Database integration is essential for the Intrusion Detection System (IDS) to store and manage large volumes of network traffic data, intrusion alerts, user activity logs, and model performance metrics. This system employs a relational database, such as MySQL or PostgreSQL, selected for its reliability, scalability, and ability to support complex queries critical for IDS functionality. The database schema is designed to manage various data types, including raw network data, processed machine learning features, user details, and system logs, ensuring efficient data storage and retrieval. Key tables store network traffic information, categorized intrusion records, model configurations, and evaluation outcomes. The integration is managed using SQLAlchemy with Flask, providing an Object Relational Mapping (ORM) layer to simplify database interactions and improve code maintainability. Regular backups and indexing are applied to boost performance and maintain data integrity.

Programming Languages and Tools

Python is selected as the primary programming language because of its comprehensive libraries, frameworks, and user-friendly nature, which facilitate the implementation of machine learning models and data processing tasks. Its rich ecosystem supports fast development and testing, making Python an ideal choice for building a machine learning-based Intrusion Detection System (IDS).

Key Programming Languages and Tools:

Reason for Use: Python is widely used in the field of data science and machine learning due to its readability, simplicity, and the vast number of libraries available for data manipulation, machine learning, and visualization.

Key Libraries:

Scikit-learn: This library is used for implementing machine learning algorithms such as Support Vector Machine (SVM) and K-Nearest Neighbors (KNN). It provides a robust framework for model training, testing, and evaluation.

Pandas: Pandas is a powerful data manipulation library in Python that allows for easy handling and analysis of large datasets. It is used for preprocessing the NSL-KDD dataset, including tasks like data cleaning, feature selection, and data normalization.

NumPy: This library provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. It is essential for handling numerical data and performing mathematical operations needed for machine learning.

Matplotlib and Seaborn: These libraries are utilized for data visualization, helping in plotting graphs such as accuracy graphs, confusion matrices, and ROC curves to evaluate the performance of the machine learning models.

Integrated Development Environment (IDE):

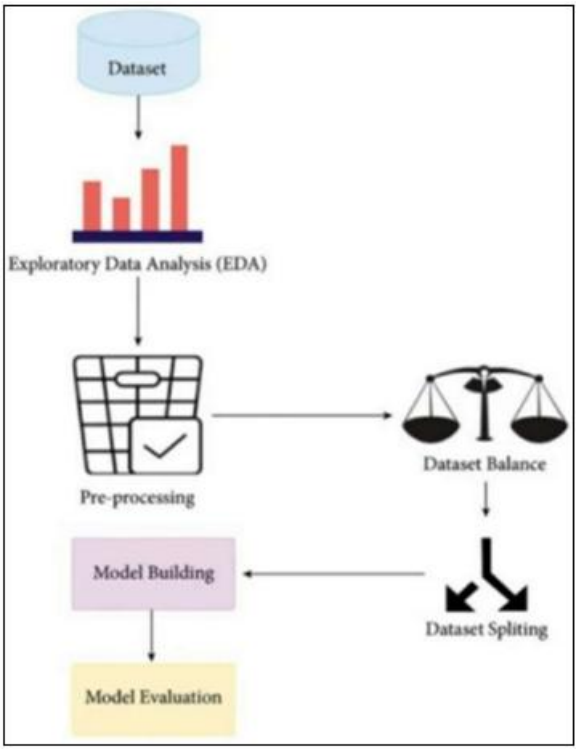
Jupyter Notebook: Jupyter Notebook is used as the primary development environment because of its interactive features, which make it easy to write and debug code, visualize data, and document the research process.

Data Processing Tools:

Scikit-learn Pipelines: Used for creating workflows that automate the data preprocessing steps (e.g., scaling, encoding) and model training. This ensures that all steps are performed consistently and efficiently.

Model Evaluation and Deployment:
TensorFlow and Keras (if used in extended versions): If exploring deep learning models in future research, TensorFlow and Keras can be used for building and training deep neural networks to enhance detection accuracy further

4. Methodology Flow Diagram



5. Result Analysis

We propose an approach that improves intrusion detection in networks by utilizing the SVM and KNN machine learning techniques. We evaluate the performance of our methodology on a publicly available dataset that is updated on a regular basis, encompasses multiple network protocols, and a range of attack methods. Our experimental results demonstrate the effectiveness of our proposed technique, which outperforms SVM on average. A final performance evaluation is obtained by assessing our approach's overall classification and prediction efficacy using multiple indicators

Performance Analysis Metrics

Accuracy
The capacity of a predictor to accurately forecast class labels for fresh data is measured by its accuracy. It assesses how well a certain predictor can forecast an attribute's value for data that hasn't yet been seen. An essential component of evaluating the predictor's performance is how well it can make predictions

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

Precision
The ratio of correctly identified positive cases to the total of correctly and mistakenly identified positive instances is known as precision, and it is a performance statistic.

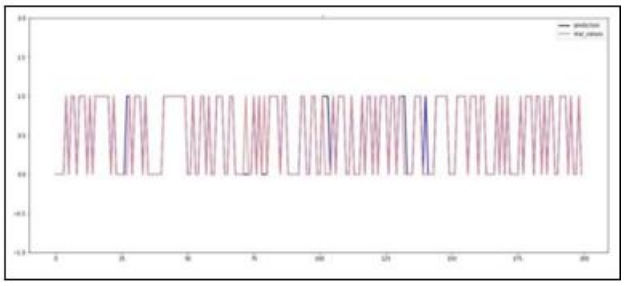
$$\text{Precision} = TP / (TP + FP)$$

Recall
A performance statistic called recall assesses a binary classifier's accuracy in identifying positive cases. This measure assesses the percentage of correctly detected positive cases compared to all positive cases, including those that were incorrectly classified as negative

$$\text{Recall} = TP / (TP + FN)$$

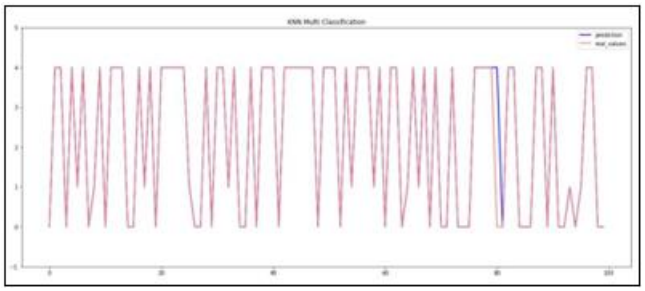
The Y axis represents the binary classification for data ranging from [-1, 1] where value 0 points to the data in "Normal category" and 1, points to the data in "Attack Category"

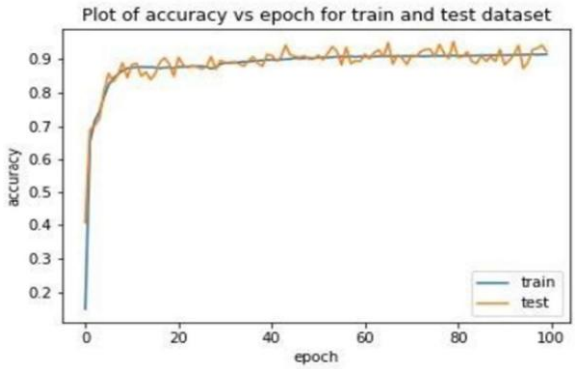
The X axis represents the data serial no. of data from the tabular entry



The Y axis represents the multiclass classification for data ranging from [-1, 1] where value 0 points to the data in "Normal category" and [1, 4] points to various attacks.

The X axis represents the data serial no. of data from the tabular entry





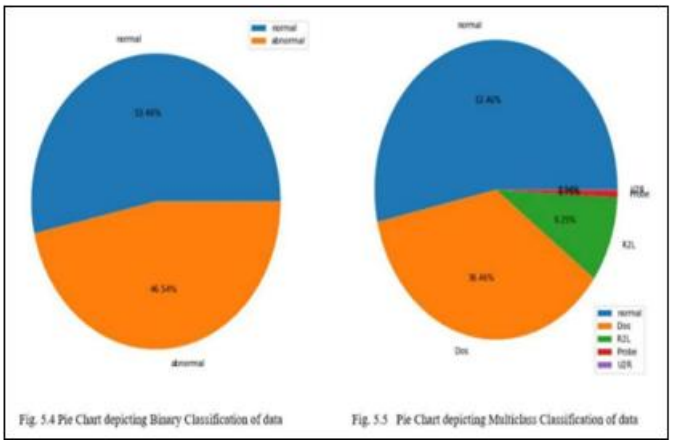


Fig. 5.4 Pie Chart depicting Binary Classification of data

Fig. 5.5 Pie Chart depicting Multiclass Classification of data

6. Conclusion and Future Scope

This system applies machine learning algorithms, specifically SVM and KNN, to introduce an innovative approach for detecting network intrusions. A series of tests, covering both binary and multi-class classification scenarios, were conducted on a widely recognized dataset to thoroughly evaluate this method's effectiveness. The dataset was chosen for its regular updates, variety of attack types, and inclusion of multiple network protocols. Experimental findings confirm the system's accuracy in identifying various network intrusions, including command and control attacks, denial of service (DoS) attacks, and reconnaissance activities.

In summary, our approach provides a practical and cost-efficient solution for detecting intrusions in increasingly complex network environments. By leveraging machine learning algorithms, this method contributes to strengthening network security.

Future Scope

Future research can investigate unsupervised algorithms for network intrusion detection. Unsupervised approaches can have an advantage when the dataset is minimal or the types of assaults are not clearly defined, as they do not require labeled data. Using clustering techniques like k- means and hierarchical clustering, network traffic may be separated into various groups. These techniques can also be used to identify anomalous groupings that might indicate an assault or intrusion. It is also possible to find network traffic anomalies by employing techniques like Isolation Forest and one-class SVM. Furthermore, a multilayered model that incorporates several machine learning and deep learning methods might improve detection performance even further. Because the output of one layer may be utilized as the input for the following layer, the model can capture more intricate relationships in the network traffic data. Deep learning techniques can be used in conjunction with other machine learning algorithms, such as support vector machines and decision trees, to improve overall performance

Furthermore, the performance of the constructed model may be evaluated on a bigger and more diverse dataset in order to improve its generalizability. To evaluate the model's ability to identify and identify assaults, real networks can be used. It will also be important to address difficulties related to real-time processing, network scalability, and computational efficacy in order to apply the model on a large scale. Lastly, techniques like attribution analysis and feature visualization may be applied to improve the produced model's comprehensibility. This will improve our understanding of the model overall and enable us to pinpoint the aspects that are most crucial for identifying intrusions. The use of explainable AI approaches can also help to build confidence and transparency in the decision-making process of the produced model. Exploring unsupervised algorithms for intrusion detection in networks is a promising area for future research. Unsupervised algorithms are advantageous in situations where labeled data is limited or the types of attacks are not clearly defined because they do not rely on labeled data. Network traffic can be grouped into different clusters using clustering algorithms like k-means and hierarchical clustering, allowing the identification of anomalous clusters that may point to the presence of an intrusion. One can also use algorithms for detecting anomalies in network traffic, such as one-class SVM

In conclusion, future research might concentrate on investigating unsupervised algorithms, creating multilayered models, analyzing performance on bigger datasets, testing in actual networks, addressing scalability and efficiency issues, and enhancing interpretability utilizing explainable AI techniques. These developments will strengthen the security of networks and increase the identification of attacks

References

[1] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep Learning Approach Combining Sparse Autoencoder With SVM for Network Intrusion Detection," in IEEE Access, vol. 6, pp. 52843-52856, 2018, doi: 10.1109/ACCESS.2018.2869577

[2] P. Tao, Z. Sun, and Z. Sun, "An Improved Intrusion Detection Algorithm Based on GA and SVM," in IEEE Access, vol. 6, pp. 13624-13631, 2018, doi: 10.1109/ACCESS.2018.2810198.

[3] S. Subbiah, K. S. M. Anbananthen, S. Thangaraj, S. Kannan, and D. Chelliah, "Intrusion detection technique in wireless sensor network using grid search random forest with Boruta feature selection algorithm," in Journal of Communications and Networks, vol. 24, no. 2, pp. 264-273, April 2022, doi: 10.23919/JCN.2022.000002.

[4] P. L. S. Jayalaxmi, R. Saha, G. Kumar, M. Conti, and T. -H. Kim, "Machine and Deep Learning Solutions for Intrusion Detection and Prevention in IoTs: A Survey," in IEEE Access, vol. 10, pp. 121173-121192, 2022, doi: 10.1109/ACCESS.2022.3220622.

[5] Nitasha Sahani, Ruoxi Zhu, Jin-Hee Cho, and Chen-Ching Liu. "Machine Learning-based Intrusion Detection for Smart Grid Computing: A Survey." ACM Trans. Cyber-Phys. Syst. 7, 2, Article 11 (April 2023).

[6] Wenjuan Wang, Xuehui Du, Na Wang, "Building a Cloud IDS Using an Efficient Feature Selection Method and SVM" in IEEE Access, vol. 7, pp 1345 – 1354, 2018, doi: 10.1109/ACCESS.2018.2883142.

[7] Z. Yuyang, C. Guang, J. Shanqing and D. Mian, "Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier," in Computer Networks, 2019.

[8] Usman Musa, Megha Chhabra, Aniso Ali, Mandeep Kaur, “Intrusion Detection System using Support Machine Vector Machine,” in 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020 doi: 10.1109/ICOSEC49089.2020.9215333.

[9] M. Shafiq, X. Yu, A. K. Bashir, H. N. Chaudhry, and D. Wang, “A machine learning approach for feature selection traffic classification using security analysis,” The Journal of Supercomputing, vol. 74, no. 10, pp. 4867–4892, 2018.

[10] P. Patel and S. Priya, "Development of a student attendance management system using RFID and face recognition: a review," International Journal of Advance Research in Computer Science and Management Studies, vol. 2, no. 8, pp. 109-119, 2014