



Intelligent Cyber Security Systems Using Machine Learning

Harshit Mehra^{1*}, Priyanka Behki¹, Krishna Kumar Choudhary¹, Tamanna Jha¹,
Vankayala Pallavi¹, Hiranya Sudha¹

¹Department of Computer Science and Engineering, Chandigarh University, India
harshit3875.hm@gmail.com

Abstract. The concern of Cybersecurity has increased due to drastic growth of internet technologies and digital services. In the earlier times the security systems were not that effective as they worked upon predefined rules and the attacks they knew. Those systems do not stand strong against the evolving threats like zero-day attacks. To fill this gap our study proposes a machine learning-based intrusion detection system. Our system works with the NSL-KDD dataset to train and evaluate various supervised learning models like Random Forest, Support Vector Machine, Decision Tree, Naïve Bayes, and Logistic Regression. The main parameter of measurement of the models is based on accuracy and computational efficiency. The results depicted, methods that combine multiple models like Random Forest perform better in detecting complex attack patterns. Our study mainly focuses on the importance of proper data preprocessing, feature selection, and model selection in making reliable intrusion system for the upcoming new cybersecurity applications.

Keywords: Cybersecurity, Machine Learning, Intrusion Detection System, Network Security, Threat Detection, NSL-KDD

Introduction

Each and every sector including banking, healthcare, education, and online commerce, are adopting digital platforms heavily that has gradually increased the dependency of them on networked systems. These transformations have made the organizations as well as individuals exposed to various cybersecurity threats, including denial-of-service attacks, phishing, malware, and unauthorized access [2], [9]. These threats can compromise sensitive data, and lead to heavy financial and operational losses. The older intrusion detection systems (IDS) were firstly rule based and had to comply with the predefined signatures created by security experts. Meanwhile these systems are good at detecting the known attacks but struggle in detecting the upcoming new threats like zero-day attacks and polymorphic malware [3]. The behaviour of these attacks is really difficult to detect to the traditional system because of the frequent change in their behavior. Also it is really difficult to maintain a large database containing large signatures which also does not guarantee real-time protection. Machine learning appears as the solution to all of the listed challenges. It enables the system to learn the patterns from previous network data and automatically

adapt to new threats [4], [6]. Unlike the older systems, the machine learning models are effective in detecting both the known and the unknown attacks. It is done by studying the deviations in the normal behaviour of the system. This study represents an intelligent intrusion detection system via predefined data-set to enhance the detection accuracy, security of network and reduce false alarms.

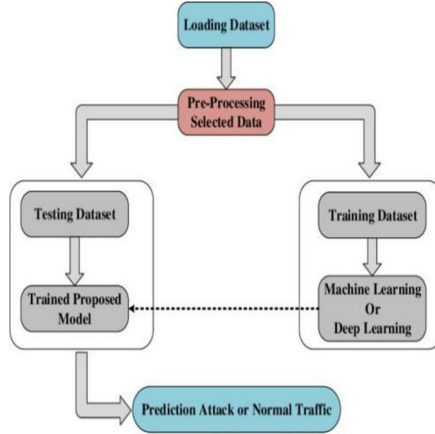


Fig. 1. Workflow of the Proposed Machine Learning–Based Intrusion Detection System

1. Related Work and Dataset Justification

Intrusion detection is one of the domains on which several research has been done using Machine Learning. The previous studies used data mining and statistical approaches to analyze the network traffic patterns and detect the anomalies. The standard datasets such as KDD Cup 99 and NSL-KDD have been widely used for evaluating the intrusion detection models due to their structured and labeled nature [1], [2], [7].

To categorize the network traffic as normal or malicious different machine learning algorithms have been used like Decision Trees, Naïve Bayes, Support Vector Machines, and Logistic Regression [4], [5]. They demonstrate strong performance in detecting common attack types such as denial-of-service and probing attacks.

Past studies have also explored various deep learning and machine learning approaches for intrusion detection. Deep learning methods can learn complex patterns from network traffic data [13], and machine learning methods can provide better solutions for intrusion detection [14]. Hybrid intrusion detection combines anomaly-based and misuse-based detection, which improves the identification of different types of attacks [15].

Yet detecting the rare and complex attacks is a bit challenge due to their class imbalance and insufficient representation in training datasets [1], [6]. Also the presence of useless and irrelevant features can impact the performance of the model negatively by increasing computational complexity and introducing noise.

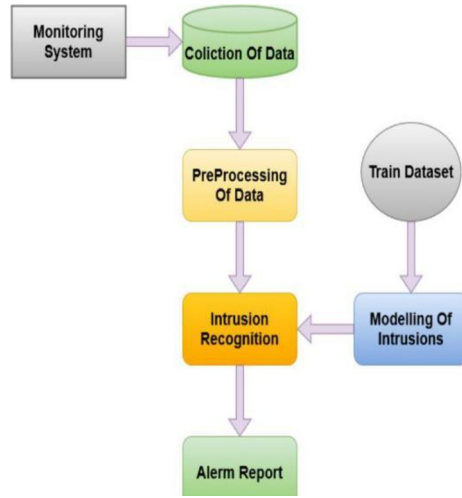


Fig. 2. Workflow of Proposed System

Dataset Justification

Besides the wide use of the KDD Cup 99 dataset, it has been criticized for containing redundant records and outdated attack patterns. To resolve this issue our study uses the NSL-KDD dataset, which is an improvement over it and also provides a more balanced distribution of attack classes [7].

Even though newer datasets like UNSW-NB15 and CICIDS2017 give more realistic and up to date traffic patterns, NSL-KDD is still considered a reliable benchmark for testing the machine learning based threat detection systems, mainly because of its organized labelling and how widely it has been used in research[11],[12].

Even though NSL-KDD is based on an older dataset, it is yet openly used as a standard model for evaluating the intrusion detection systems because of its structured format and labeled data. It gives a balanced dataset that helps in genuine comparison of various machine learning models. In future the newer datasets like CICIDS2017 and UNSW-NB15 will be used in order to validate the system in modern network environments.

2. Proposed Methodology

2.1. System Overview

Our proposed system works on a structured manner that is designed in a way to process the network traffic data efficiently and detect the malicious activities in an accurate manner. The major stages of the system are as follows:

1. Fetching data from installed datasets

2. Wipe out and reprocess the invalid data
3. Select the relevant information for reliable identification of attacks
4. Train the model using labeled data to recognize attack patterns
5. Identify the Intrusion and its type using trained model

Following this approach helps in attaining scalability, efficiency, and adaptability.

2.2. Dataset Description

In our study for the efficient and reliable detection of the complex and rare attacks we have used the NSL-KDD dataset because of its capability to detect the normal activities and various types of cyber attacks. It is able to do so because the dataset consists of the labeled network records.

Different aspects of the network communication is included in each and every record of the dataset which gradually helps the model to capture the network behaviour including both the short term and the long term.

The dataset also shows real-world challenges like class imbalance, where some attack types occur less frequently than others. This makes it suitable for evaluating how effective and how well the intrusion detection models generalize.

TABLE I. Summary of Benchmark Intrusion Detection Dataset Characteristics

Parameter	Description
Dataset Type	Benchmark intrusion detection dataset
Total Records	Large-scale network traffic records (normal + attack)
Number of Features	Multiple features describing network connections
Traffic Classes	Normal, Malicious
Attack Categories	DoS, Probing, U2R, and R2L attacks
Feature Types	Numeric & categorical
Data Labeling	Labeled data suitable for supervised learning
Network Information	Protocol type, connection duration, source & destination statistics
Traffic Characteristics	Packet-level and host-based attributes
Class Distribution	Imbalanced (attack types occur with different frequencies)
Application Domain	Network intrusion detection and cybersecurity research
Usage	Training and evaluation of machine learning-based IDS

The attributes that were seen as playing a vital role in identifying the malicious traffic were, duration, protocol type, source bytes, and connection count. These helped the models to differentiate between normal and attack behavior.

2.3. Data Preprocessing

For our proposed model, data preprocessing plays an important role because the quality of the data directly affects the efficiency of the machine learning models. They are heavily dependent on it. The network traffic datasets contain inconsistencies such as missing values, duplicate records, and mixed data types because of their large size and complexity.

To scale the numerical features to a common range, normalization is also used, which also ensures that not even a single feature disproportionately influences the model. These steps enhance the data consistency, reduce noise, and improve the overall accuracy and stability of the models [7], [10].

2.4. Feature Selection

To enhance the overall efficiency and performance of intrusion detection system feature selection plays a vital role. The NSL-KDD dataset contains a large amount of features among which many can be irrelevant or redundant.

By selecting only the most informative features, the system is able to reduce its computational complexity and improve the model's generalization. The selected features are as follows:

1. Duration of connection
2. Protocol type
3. Service type
4. Connection flag
5. Source and destination bytes
6. Connection count within a time window
7. Service rate metrics
8. Host-based traffic features

These features give valuable insights into the network behavior and are really good in telling difference between the normal and malicious activities.

Feature selection also enhances the understandability as it makes it really simple for the security analyst to understand which attribute is contributing the most and which is giving false results, therefore allowing better decision-making in cybersecurity operations [6], [10].

Table II. Selected features used for Intrusion detection

Feature Name	Feature Category	Description
Duration	Basic Feature	Time for which the connection remains active
Protocol Type	Basic Feature	Protocol used for the network connection
Service	Basic Feature	Service used in the connection
Flag	Basic Feature	Indicator of connection status
Source Bytes	Traffic Feature	Data sent from the source to the destination
Destination Bytes	Traffic Feature	Data returned from the destination to the source
Count	Time-based Feature	Connections made to the same host within a given time period
Same Service Rate	Time-based Feature	Share of connections using the same service
Different Service Rate	Time-based Feature	Share of connections involving different services
Host Service Count	Host-based Feature	Total connections with the same host
Host Same Source Port Rate	Host-based Feature	Frequency of connections using a same source port

2.5. Machine Learning Models

Various machine learning models are evaluated by our proposed system to check and verify the overall effectiveness in intrusion detection. The models used in this study include Naïve Bayes, Decision Tree, Random Forest, Support Vector Machine, Logistic Regression, and Gradient Boosting.

The different classifiers used in this study follow different learning approaches. Naïve Bayes works on probability theory and assumes that the features are independent of each other. Logistic Regression is a linear classification algorithm used to distinguish between two classes. The optimal decision boundaries are identified by the margin-based classifier named Support Vector Machine (SVM). There are two tree-based models named Decision Tree and Random Forest. Random Forest is an ensemble method that combines multiple decision trees to improve accuracy and reduce overfitting.

Different learning approaches are highlighted by all of the models like probabilistic, linear, non-linear, or ensemble based. These variety of models helps us to decide which

model performs best in detecting network intrusions by a proper comparison between all.

100 decision trees were used to build the Random Forest model. To do the classification SVM used a RBF (Radial Basis Function) whereas the parameters of Logistic Regression and Naïve Bayes were kept at default. 100 estimators were implemented with Gradient Boosting with a learning rate of 0.1 balancing the performance and computational cost. The main idea behind choosing these Model settings was to maintain a balance between the detection accuracy and execution efficiency.

Labeled data was used to train the models and evaluate them on an unseen test data, using an 80:20 train-test split. Using this Comparison we identified the most suitable model for the intrusion detection, based on different parameters like accuracy, efficiency, and robustness.

TABLE III. Machine Learning models used for Intrusion Detection

Model	Learning Type	Key Characteristics
Naïve Bayes (NB)	Probabilistic	Attain quick results and assume each feature as independent information
Decision Tree (DT)	Tree-based	Can handle complex data patterns and its decision making is easy to understand
Random Forest (RF)	Ensemble	Gives most accurate results and performs best on new and unseen data
Support Vector Machine (SVM)	Margin-based	It works well with data containing many features

3. Experimental Results

The successful execution of the experiments were performed using Python on a Windows-based system. All the libraries that were essential for machine learning and data analysis were used. The dataset was divided into 80% training data and 20% testing data to ensure a fair evaluation.

Performance Analysis

On the basis of classification the models were initially evaluated and ensemble-based methods came out to be the one with best results that the individual classifiers.

1. The maximum accuracy of 99.96% was achieved by the Random Forest which was the highest.
2. Effective performance was also noticed by Support Vector Machine and Gradient Boosting.

3. Good results with reasonable complexity was provided by Decision Tree and Logistic Regression
4. Due to the assumption of feature independence by Naïve Bayes, it showed lowest accuracy

While conducting the experiment not only accuracy but computational efficiency is also taken into consideration. In the simpler models like Naïve Bayes and Logistic Regression a Faster execution time were observed making them suitable for systems with limited computing power. Ensemble models showed better performance but required more computational resources. The overall results highlight the balance between accuracy and execution time showing the importance of selecting a model based on the requirements of the specific application. The best performance of Random forest indicates that ensemble learning methods are effective in handling complex intrusion patterns.

TABLE IV. Comparison of Performance for different Machine Learning models

Machine Learning Model	Accuracy (%)	Execution Time (Seconds)
Naïve Bayes (NB)	87.90	Low
Decision Tree (DT)	99.05	Moderate
Random Forest (RF)	99.96	High
Support Vector Machine (SVM)	99.87	High
Logistic Regression (LR)	99.35	Low
Gradient Boosting (GB)	99.71	High

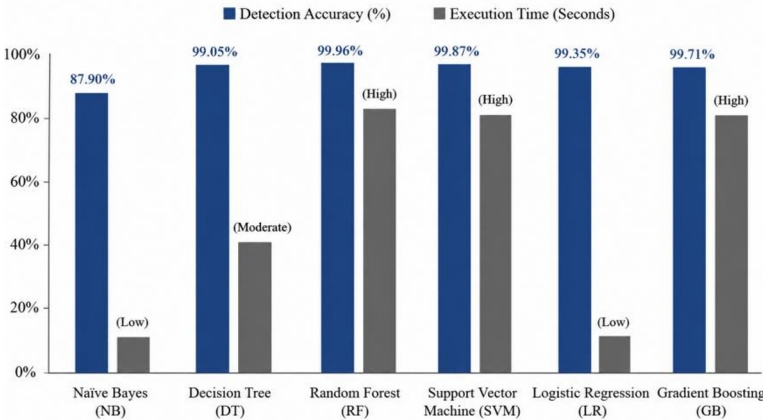


Fig. 3. Accuracy comparison of machine learning classifiers for intrusion detection

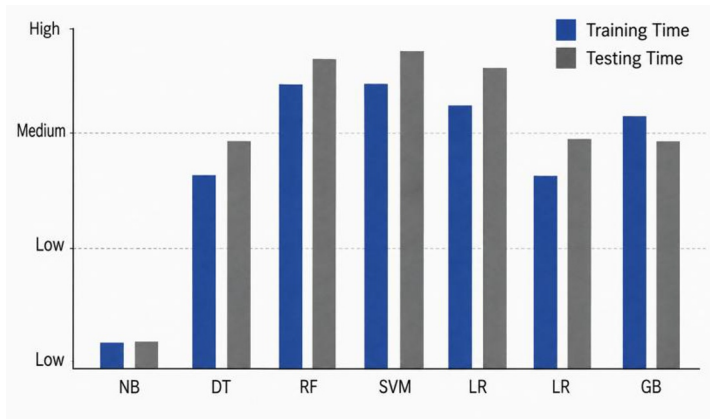


Fig. 4. Comparison of training and testing time for different machine learning models used in intrusion detection

Conclusion

The limitations of the older rule-based systems is addressed by our study of machine learning-based intrusion detection system. The system is made effective against both the known and unknown attacks or threats using machine learning models and feature selection.

The experimental results indicates that ensemble methods gives out the best results particularly Random Forest. Our Study on the importance of selecting useful features and creating a balance between accuracy and computational efficiency.

The future studies will majorly focus on integrating deep learning techniques, implementing real-time detection systems, and evaluating performance on the upcoming new datasets like CICIDS2017 and UNSW-NB15, in order to further improve the system's ability to handle modern cyber threats.

Acknowledgments. The authors would like to thank Chandigarh University for providing the necessary support and resources for this research work.

Disclosure of Interest: The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Landge, P., Sherekar, S.S., Chavan, S.R.: Intelligent Cyber Security System Using Machine Learning Techniques on KDD-99 Dataset. IEEE (2024)
2. Tavallaei, M., et al.: A Detailed Analysis of the KDD Cup 99 Data Set. IEEE CISDA (2009)
3. Sarker, I.H.: Machine Learning for Intelligent Data Analysis in Cybersecurity. Annals of Data Science (2023)
4. Farnaaz, N., Jabbar, M.A.: Random Forest Modeling for Network Intrusion Detection System. Procedia Computer Science (2016)
5. Kilincer, I.F., et al.: Machine Learning Methods for Cyber Security Intrusion Detection. Computer Networks (2021)
6. Moustafa, N., Slay, J.: Feature Selection for Intrusion Detection Systems. IEEE Transactions on Big Data (2019)
7. Dhanabal, L., Shantharajah, S.P.: Intrusion Detection Using NSL-KDD Dataset. IJARCCCE (2015)
8. Wang, Y., et al.: Network Intrusion Detection Using Deep Learning. Mobile Information Systems (2022)
9. Naik, B., et al.: AI Techniques in Cybersecurity. Complex & Intelligent Systems (2022)
10. Khammassi, C., Krichen, S.: Feature Selection Using Genetic Algorithms for IDS. Computers & Security
11. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. ICISPP (2018)
12. Moustafa, N., Slay, J.: UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems. Military Communications and Information Systems Conference (2015)
13. Vinayakumar, R., et al.: Deep Learning Approach for Intelligent Intrusion Detection System. IEEE Access (2019)
14. Ahmad, Z., et al.: Network Intrusion Detection System Using Machine Learning Techniques. IEEE Access (2021)
15. Kim, G., Lee, S., Kim, S.: A Novel Hybrid Intrusion Detection Method Integrating Anomaly Detection with Misuse Detection. Expert Systems with Applications (2014)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

