ANALYSIS OF NETWORK INTRUSION DECTECTION USING FEATURE ENGINEERING IN ML

G Tejaswini[1], Dasari Tejaswi[2], Jogu Kavya[3], Bulla Kavitha[4], Ameesha Shaik[5]
Tadi Shalini[6]

[1-5] Undergraduate Students, Department of ECE, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur, Andhra Pradesh, India
[6] Assistant Professor, Department of ECE, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur, Andhra Pradesh, India

ABSTRACT

Intrusion detection Systems (IDS) play a critical role in safeguarding computer networks from malicious activities. This project aims to enhance IDS performance through the integration of feature engineering, specifically Recursive Feature Elimination (RFE), coupled with the Random Forest algorithm. Feature engineering involves selecting and transforming input variables to improve model performance. RFE, a subset selection technique, iteratively removes less significant features, enhancing model efficiency and interpretability. Random Forest, a powerful ensemble learning method, leverages decision trees to classify data, offering robustness and accuracy in complex datasets. The evaluation metrics include accuracy, precision, recall, F1-score, and Cross Validation graphs. Experimental results will demonstrate the efficiency of the proposed approach in improving IDS performance in terms of detection accuracy and efficiency. This project contributes to the field of cybersecurity by providing insights into the effectiveness of feature engineering techniques, particularly RFE, in enhancing IDS capabilities. Ultimately, this project seeks to contribute to the advancement of intrusion detection techniques, offering insights into effective feature selection strategies and algorithmic frameworks for enhancing network security in contemporary computing environments.

Keyword: - *Machine Learning, Intrusion Detection System, Cybersecurity and Feature Engineering etc…*

1. INTRODUCTION

With the rapid growth in the use of the internet and activities associated with it, our networks, as well as systems, have become more prone to attacks. As a user one never pays much attention to the challenges that come along with the use of the internet. Since the technology keeps on changing so should the ways to secure a system or a network and make it more stable. Though many security features such as antivirus and firewalls have been a part of our network security, these protective applications can also be exploited. Therefore, the importance of an intrusion detection system became an integral part of any network.

IDS keep a track of any abnormality in the network. The more trained the IDS more is the accuracy. To identify or detect any intrusion it is important to differentiate between a normal user's behaviour and an attacker's activities. It is also important to define a set of rules and train them against a network to develop an efficient IDS. The properties of an efficient IDS are improved detection rate and reduced wrong alarm rate. Currently used IDS cannot still detect the attacks that are unknown because the network environment keeps on changing rapidly. Generally, IDSs are implanted in the proximity of safe network nodes i.e., Switches in bigger networks.

2. LITERATURE SURVEY

The author [1] has studied the network security issues and conducted the experiment using Naive Bayes, Random Forest Support Vector Machine and K-means ML algorithms to detect four types of attacks like DOS, PROBE, U2R and R2L. They concluded that the Random Forest Classifier (RFC) surpasses the other methods and also stated that hierarchical clustering method can be used to improve the performance of the system. In paper [2] the author has done a comparison using supervised machine learning classifiers namely, Random Forest, Support Vector Machine, Gaussian Naive Bayes, and Logistic Regression are compared for an intrusion detection in network. Effective classifying algorithm is identified based on performance matrix namely F1-Score, accuracy,precision, and recall. Based on the observed results they have concluded that the Random forest classifier outperforms other classifiers for the considered data-set and parameters.

A light weight IDS method is proposed here [3] mainly concerned on pre-processing of the data, so that they can use important features of online data. The main step is to remove the redundant data from dataset to standardize the data. This helps the machine learning algorithms to give the unbiased and accurate result. In paper [4] the author proposed intrusion detection system(IDS) using supervised machine learning techniques to detect the online network data as normal or anomaly. The proposed method only identifies the Denial of Service (DOS) and probe attacks, but the other attacks are not taken into consideration. The author proposed Intrusion detection method using Support Vector Machine (SVM) [5]. They also used feature removal method to improve the efficiency of the algorithm. Using the proposed feature removal method, they selected best nineteen features from the KDD-CUP99 data-set. The authors have proposed [6] anomaly intrusion detection using improved Self Adaptive Bayesian Algorithm to process the large amount of data. In this paper [7] authors proposed a novel idea to reduce the dimensionality of the data by using triangle-based K-NN approach.

An Intrusion Detection system using fuzzy logic is tested [8]. This technique uses a set of fuzzy rules which are obtained from the definite rules using frequent items. The classification accuracy of this approach is above 90% for all types of attacks. G.Meera Gandhi et al., [9] examined the performance measure of four supervised machine learning algorithms in detecting the four types of attack such as DoS, R2L, Probe, and U2R. The result shows that the C4.5 decision tree classifier performs best in prediction accuracy compared to Naıve Bayes. The authors [10] have compared the performance of the three machine learning algorithm namely Neural Network, Support Vector Machine and Decision Tree. The algorithms were measured based on false alarm rate, accuracy and detection rate of four categories of attacks classes. From these experiments they found that the Decision tree (J48)algorithm outperformed the other two algorithms.

The author [11] suggested using a collective of, Support Vector tor Machines (SVMs), Multivariate Adaptive Regression Splines (MARS), and Artificial Neural Networks (ANNs). Whereas [12] proposed a hybrid approach in which they have used a Support Vector Machine (SVM) and Radial Basis Function (RBF). The sequential search strategy for feature selection or feature extraction through determining the importance of a given attribute by simply removing it and recording the performance [13]. If performance of the algorithm is increased, then the feature is unimportant and thus shall be removed. The author [14] suggested that every attribute in the dataset is not much important andit will not give the accurate result as expected. It is very important to reduce the no of features using feature selection technique and also they concluded that simple Cart algorithms gives accurate result than other five algorithm J4.8 Naive Bayes, NB Tree, Multi-Layer Perceptron, and SVM.

3. EXISTING SYSTEM

The existing system employs Intrusion Detection Systems (IDS) utilizing the Naive Bayes algorithm and Principal Component Analysis (PCA) for feature reduction. Naive Bayes is a probabilistic classifier based on Bayes' theorem, which assumes independence between features. Despite its simplistic assumption, Naive Bayes often performs well in practice, particularly with large datasets and relatively independent features. In the context of IDS, Naive Bayes efficiently categorizes network traffic into normal or intrusive based on learned probabilities from historical data. Principal Component Analysis (PCA) is a dimensionality reduction technique used to transform high-dimensional data into a lower-dimensional space while preserving the most important information. By identifying the principal components capturing the maximum variance in the data, PCA reduces the computational complexity and enhances the efficiency of subsequent analysis. This combination of Naive Bayes and PCA offers a lightweight and computationally efficient solution for intrusion detection. While Naive Bayes provides rapid classification based on probability estimates, PCA streamlines the feature space, mitigating the curse of dimensionality and enhancing the system's scalability.

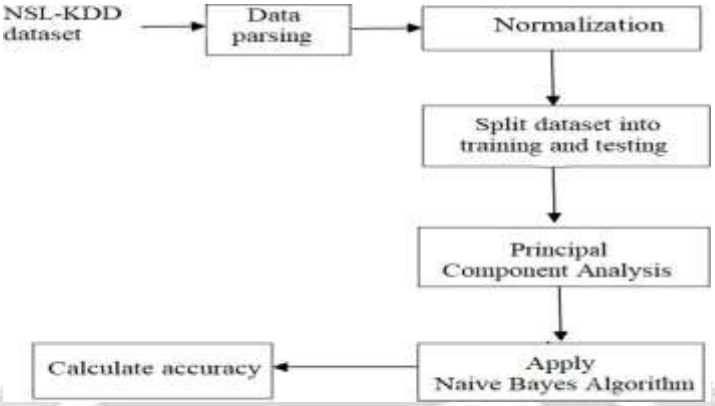

NSL-KDD dataset
Data parsing
Normalization
Split dataset into training and testing
Principal Component Analysis
Apply Naive Bayes Algorithm
Calculate accuracy

Fig -1: Block diagram of Existing System

3.1 Drawbacks

One significant drawback of the existing system is its reliance on the Naive Bayes algorithm. Naive Bayes assumes independence between features, which may not hold true in real-world scenarios, especially in complex network environments where correlations between features exist. This can lead to suboptimal performance, particularly when dealing with correlated or dependent features, as Naive Bayes may overlook important relationships between variables, potentially resulting in decreased detection accuracy. Additionally, while PCA is effective in reducing the dimensionality of high-dimensional data, it does so by transforming the original features into a new set of orthogonal variables. While this reduction in dimensionality can improve computational efficiency, it comes at the cost of interpretability. The transformed features may not directly correspond to the original variables, making it challenging to interpret the results and understand the underlying factors contributing to intrusion detection decisions. This lack of interpretability can hinder the system's ability to provide actionable insights and make informed decisions about network security measures. Overall, while the existing system offers a lightweight and computationally efficient approach to intrusion detection, its reliance on Naive Bayes and PCA may introduce limitations in terms of detection accuracy, interpretability, and adaptability compared to the current system employing feature engineering and the Random Forest algorithm.

4. PROPOSED SYSTEM

The current model for intrusion detection leverages advanced techniques such as feature engineering (RFE) and the Random Forest algorithm to enhance accuracy and efficiency in identifying potential threats within network traffic. RFE systematically selects the most relevant features from the dataset, thereby reducing noise and focusing on the most discriminative characteristics for intrusion detection. This feature selection process not only improves the effectiveness of subsequent analysis but also enhances the interpretability of the model by highlighting the most influential factors in identifying intrusions. Complementing RFE, the Random Forest algorithm, an ensemble learning method, harnesses the power of multiple decision trees to classify network data accurately. Its ability to handle nonlinear relationships and interactions between features makes it well-suited for capturing complex intrusion patterns. By integrating these techniques, the current system not only boosts detection accuracy but also ensures scalability and adaptability to evolving threats, thereby offering a robust solution for safeguarding computer networks against malicious activities.

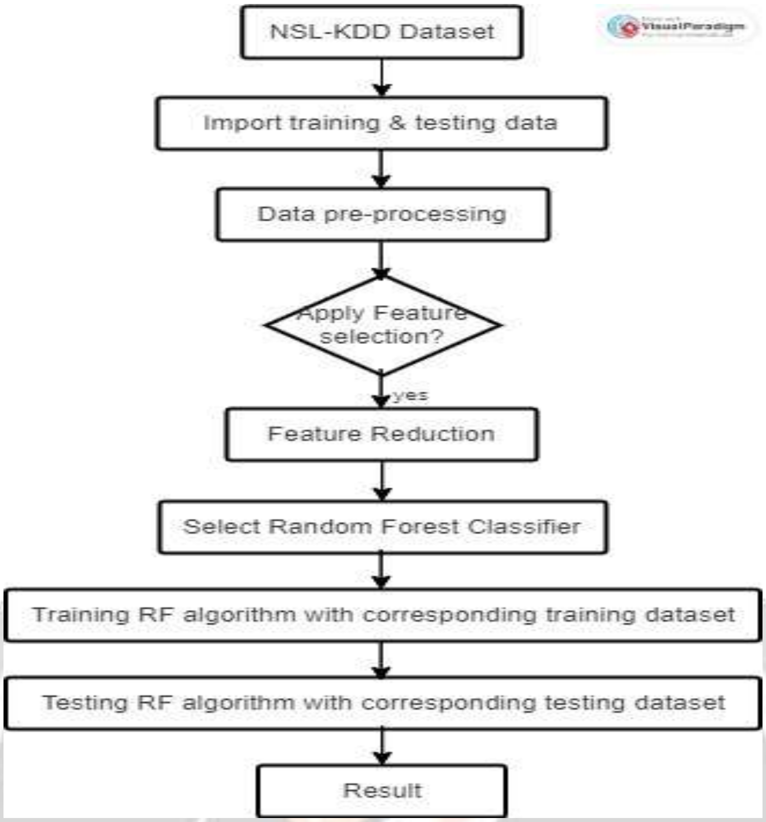

NSL-KDD Dataset
Import training & testing data
Data pre-processing
Apply Feature selection?
yes
Feature Reduction
Select Random Forest Classifier
Training RF algorithm with corresponding training dataset
Testing RF algorithm with corresponding testing dataset
Result

Fig -2: Block Diagram of Current System

4.1 Prerequisites

- **Anaconda**: Anaconda is a distribution of Python and R languages tailored for data science and machine learning. It comes with pre-installed packages like Jupyter notebooks and Spyder IDE, simplifying setup and management. Anaconda streamlines workflows for developers and data scientists, providing a robust platform for building data-driven applications.

- **Jupyter Notebook**: Jupyter Notebook is an open-source web application that allows users to create and share documents containing live code, equations, visualizations, and narrative text. Originally developed for the Python programming language, Jupyter Notebook supports various programming languages through kernels, including but not limited to Python, R, Julia, and Scala.

- **NSL-KDD Dataset**: The NSL-KDD dataset is a widely used benchmark dataset for intrusion detection systems (IDS). It is an improved version of the KDD Cup 1999 dataset, designed to address some of the limitations and biases present in the original dataset. The NSL-KDD dataset contains network traffic data collected from a simulated computer network environment, including features such as packet headers, protocol types, service types, and flag values. Each record in the dataset is labeled as either normal or one of the predefined attack types, allowing researchers to train and evaluate IDS algorithms on realistic data.

4.2 Libraries

- **Pandas:** It is a popular python library for data manipulation and analysis. It provides data structures and functions for efficiently working with structured data, such as tables and time series. Pandas simplifies tasks like data cleaning, transformation, and aggregation, making it a versatile tool for data wrangling in data science, machine learning, and other analytical workflows.

- **numpy:** It is a powerful python library for numerical computing, providing support for arrays, matrices, and mathematical functions. It offers efficient operations on large datasets, making it essential for scientific computing, data analysis, and machine learning tasks. Numpy's array-oriented computing capabilities enable faster execution of mathematical operations compared to traditional Python lists, making it a cornerstone library in the Python ecosystem for numerical computing tasks.

- **Scikit learn:** It is a widely-used Python library for machine learning, offering a simple and efficient interface for various classification, regression, clustering, and dimensionality reduction algorithms. It provides tools for data preprocessing, model selection, and evaluation, making it an essential tool for building and deploying machine learning models. With scikit-learn, users can easily implement and experiment with different machine learning techniques to solve a wide range of real-world problems.

4.3 Attacks

- **DoS:** Denial of Service attacks aim to disrupt the availability of services by overwhelming the target system with a flood of illegitimate traffic or requests. The goal is to consume the resources of the target system such as bandwidth, CPU, memory, or disk space, rendering it unable to respond to legitimate requests. DoS attacks can take various forms including flooding attacks (e.g., SYN flood, UDP flood), resource depletion attacks (e.g., bandwidth exhaustion), or application layer attacks (e.g., HTTP flood targeting web servers).

- **Probe:** Probe attacks involve attempts by an attacker to gather information about a target network or system in order to identify potential vulnerabilities or weaknesses. These attacks typically involve scanning activities such as port scanning, network mapping, or reconnaissance to identify open ports, available services, and potential entry points into the target network. Probe attacks do not involve direct exploitation of vulnerabilities but rather focus on reconnaissance and information gathering.

- **R2L**: Remote-to-Local attacks involve unauthorized attempts by an attacker to gain access to a target system from a remote location. These attacks typically target vulnerabilities in network services or applications running on the target system. Common R2L attacks include brute force attacks, where the attacker attempts to guess login credentials, or exploitation of known vulnerabilities in network services to gain unauthorized access. Once access is gained, the attacker may escalate privileges to gain further control over the system.

- **U2R:** User-to-Root attacks involve attempts by an unauthorized user to escalate their privileges on a target system from a standard user level to a root or administrator level. These attacks typically target vulnerabilities in local services or applications running on the target system. Common U2R attacks include buffer overflow attacks, where the attacker exploits vulnerabilities in software to execute arbitrary code with elevated privileges, or privilege escalation attacks, where the attacker exploits misconfigured permissions or vulnerabilities to gain root access.

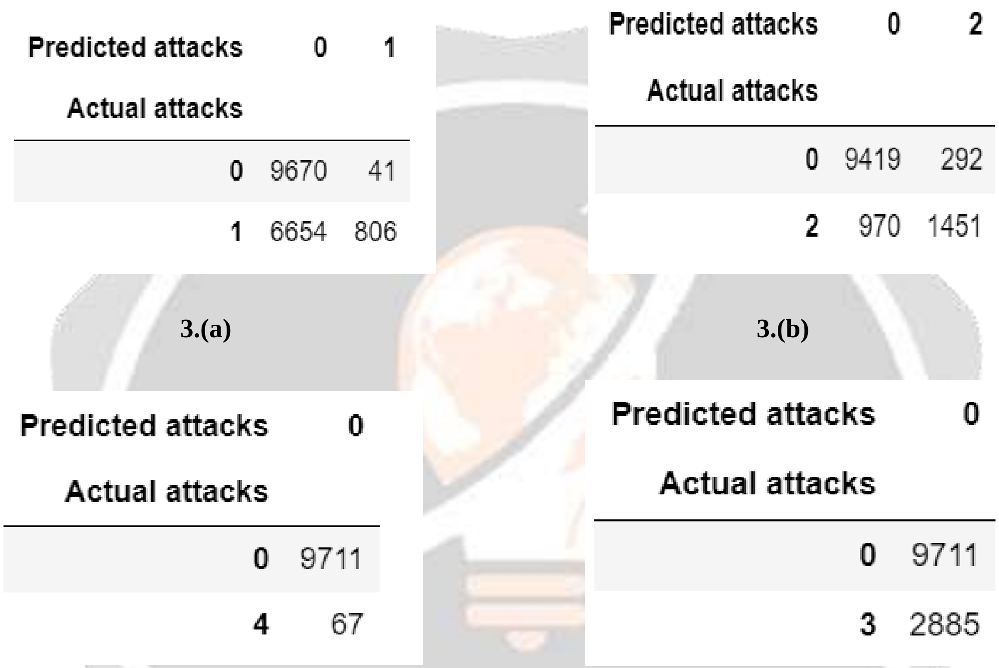
4.4 Methodology

In the current system, the first step involves preprocessing the dataset to clean and transform the raw network traffic data. Feature engineering techniques like RFE are then applied to select the most relevant features from the dataset using a Feature Engineering algorithm called Recursive Feature Elimination, thereby reducing dimensionality and improving the efficiency of subsequent analysis. Once the feature selection process is complete, a Random Forest classifier is trained on the preprocessed dataset using the selected features. Random Forest is an ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy and robustness. The end result of this process is an Intrusion Detection System (IDS) capable of effectively identifying and classifying various types of network intrusions, including Denial of Service (DoS) attacks, probing attempts, unauthorized access (R2L), and privilege escalation (U2R). Also, it provides Confusion Matrices and Cross-Validation graphs. By employing feature engineering with RFE and Random Forest, the current system achieves a high level of accuracy in detecting and mitigating security threats, thereby enhancing the overall security posture of the network environment. The main steps involved in building the model are as follows:

- Download the NSL-KDD Dataset which are in .csv format from the website.

- Install Anaconda in your computer and open Jupyter Notebook
- Pre-process the data by handling missing values, encoding categorical variables, and scaling numerical features.
- Implement RFE to select the most relevant features for intrusion detection.
- Initialize a Random Forest classifier and train it on the dataset.
- Evaluate the model's performance using metrics such as accuracy, precision, recall, and F1-score.
- Obtain confusion matrices and cross-validation graphs for each attack.

5. RESULTS



Predicted attacks	0	1
Actual attacks		
0	9670	41
1	6654	806

3.(a)

Predicted attacks	0	2
Actual attacks		
0	9419	292
2	970	1451

3.(b)

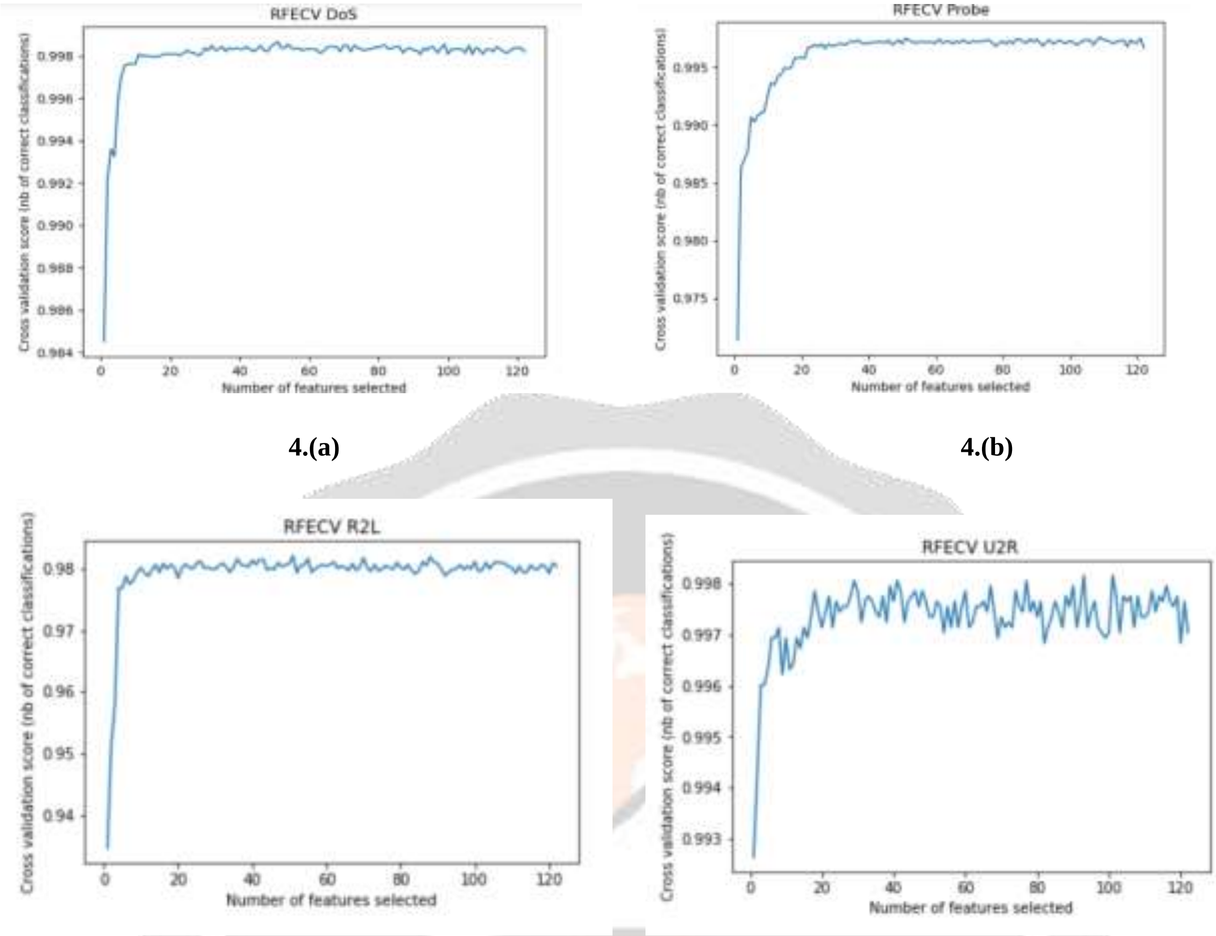
Predicted attacks	0
Actual attacks	
0	9711
4	67

3.(c)

Predicted attacks	0
Actual attacks	
0	9711
3	2885

3.(d)

Fig -3: Confusion matrices of (a) DoS, (b) Probe, (c) R2L, (d) U2R



4.(a) **4.(b)**

4.(c) **4.(d)**

Fig -4: Cross-Validation graphs of (a) Dos,(b) Probe, (c) R2L, (d) U2R

6. CONCLUSION

In conclusion, the project on building an Intrusion Detection System (IDS) using Random Forest with Feature Engineering (RFE) has yielded promising results. By leveraging advanced techniques in data pre-processing, feature selection, and machine learning, we have successfully developed an effective IDS capable of detecting and mitigating various types of network intrusions. Through the use of the NSL-KDD dataset and the implementation of RFE, we were able to identify the most relevant features for intrusion detection, thereby reducing dimensionality and improving model efficiency. The Random Forest algorithm, with its ensemble learning approach, further enhanced the accuracy and robustness of the IDS by effectively combining the predictions of multiple decision trees.

During the evaluation phase, the IDS demonstrated impressive performance metrics, including high accuracy, precision, recall, and F1-score, as well as insightful visualizations of the model's behavior. Additionally, the feature importance analysis provided valuable insights into the underlying patterns of network traffic data, aiding in the interpretation and validation of the model's effectiveness. Moving forward, the deployed IDS will play a crucial role in enhancing the security posture of the network environment by continuously monitoring for suspicious activities and adapting to evolving threats. Furthermore, ongoing monitoring and periodic retraining of the model will ensure its effectiveness over time, contributing to the overall resilience and security of the system.

7. REFERENCES

[1] A. T. Yousef El Mourabit, AnouarBouirden and N. E. Moussaid, " Intrusion Detection Techniques in Wireless Sensor Network Using Data Mining Algorithms:Comparative Evaluation Based on Attacks Detection", International Journal of Advanced Computer Science and Applications, vol. 6, no. 9, pp. 164–172, (2015).

[2] J. Manjula C. Belavagi and Balachandra Muniyal, "Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection" TwelfthInternational Multi-Conference on Information Processing- 2016.

[3] S. S. S. Sindhu, S. Geetha and A. Kannan, "Decision Tree Based Light Weight Intrusion Detection Using a Wrapper Approach, Expert Systems with Applications",vol. 39, no. 1, pp. 129–141, (2012).

[4] P.Sangkatsanee, N. Wattanapongsakorn and C. Charnsripinyo,, " Practical Real-Time Intrusion Detection Using Machine Learning Approaches, ComputerCommunications" , vol. 34, no. 18, pp. 2227–2235, (2011).

[5] Y.Li, J. Xia, S. Zhang, J. Yan, X. Ai and K. Dai, "An EfficientIntrusion Detection System Based on Support Vector Machines and Gradually Feature Removal Method,Expert Systems with Applications", vol. 39, no. 1, pp. 424430, (2012).

[6] D. M. Farid and M. Z. Rahman, "Anomaly network intrusion detection based on improved self-adaptive Bayesian algorithm" Journal of computers, vol. 5, no. 1,pp. 23–31, 2010.

[7] JB. Luo and J. Xia, "A novel intrusion detection system based on feature generation with visualization strategy", Expert Systems with Applications, vol. 41, no. 9,pp. 4139– 4147,2014.

[8] R. Shanmugavadivu and Dr. N. Nagarajan, "Network intrusion detection system using fuzzy logic", Indian Journal of Computer Science and Engineering, Vol. 2, pp.101-111.

[9] G. Meera Gandhi, "Machine learning approach for attack prediction and classification using supervised learning algorithms", International Journal of ComputerScience & Communication, Vol. 1, No. 2, pp. 247-250. July-December 2010.

[10] K. AbdJalil, and S. Mara, "Comparison of machine learning algorithms performance in detecting network intrusion", In Proceedings of Networking and InformationTechnology (ICNIT), pp. 221 – 226, Manila 2010.

[11] S. Mukkamala, A. H. Sung, and A. Abraham, "Intrusion detection using an ensemble of intelligent paradigms" J. Netw. Comput. Appl., vol. 28, no. 2, pp. 167–182,2005.

[12] M. Govindarajan and R. Chandrasekaran, "Intrusion Detection using an Ensemble of Classification Methods," In Proceedings of World Congr. Eng. Computer. Sci.,vol. I, no. October, 2012

[13] Karan Bajaj, Amit Arora, "Improving the Intrusion Detection using Discriminative Machine Learning Approach and Improve the Time Complexity by Data MiningFeature Selection Methods", International Journal of Computer Applications (0975 – 8887) Volume 76– No.1, August 2013.

[14] C.Ingre B. and Yadav A., " Performance analysis of NSL-KDD dataset using ANN," 2015 International Conference on Signal Processing and CommunicationEngineering Systems, Guntur, 2015, pp. 92-96.

[15] NSL-KDD dataset [online] available: http://nsl.cs.unb.ca/nsl-kdd/.