Email Alert-Enabled Network Intrusion Detection Systems: A Supervised Machine Learning Approach with Recursive Feature Elimination

MOHAMMAD AMAN[1], ASHISHIKA SINGH[2], TUSHAR J MALVIYA[3], ADITI A KALGI[4], SHREYA HEGDE[5], HARSH KUMAR[6]

[1, 2, 3, 4, 5, 6] *Computer Science and Engineering Presidency University Bengaluru, India*

Abstract- ***In the evolving cybersecurity environment, the importance of a robust intrusion detection system (IDS) is paramount. This research explores the integration of supervised machine learning models such as decision trees, support vector machines (SVMs) and random forests to improve the capabilities of network intrusion detection systems (NIDS). The proposed methodology includes data pre-processing, feature selection and model training using the KDD-Cup99 dataset. This research presents a comparative analysis of the performance of a model with 41 features and a reduced set of 15 features obtained by recursive feature elimination (RFE). This research contributes to understanding the effectiveness of machine learning in strengthening email-alert enabled NIDS against cyber threats.***

Indexed Terms- ***Network intrusion, Supervised Machine learning, Network Attack detection, Network Security, Email-Alert, Threat Detection.***

I. INTRODUCTION

In the ever-changing field of cybersecurity, networked systems must be protected from harmful infiltration at all costs. Network Intrusion Detection Systems (NIDS) are like mighty sentinels, using cutting-edge technologies to identify and neutralize possible threats instantly. Robust intrusion detection plays a crucial role as cyber threats become more sophisticated. As a specific tool in the larger intrusion detection framework, network intrusion detection systems (NIDS) are essential because they independently monitor and assess system and network activity. NIDS examines network data and can function as either software or hardware. Its goal is to quickly identify and address security problems. NIDS acts as a front-line defender in the complex dance between attackers and defenders by continuously examining data packets and system activity to spot departures from the norm.

In intrusion detection, Network-based Intrusion Detection Systems (NIDS) and Host-based Intrusion Detection Systems (HIDS) are the two main forms of IDS. While HIDS concentrates on keeping an eye on activity on particular devices, NIDS adopts a more comprehensive approach by looking at each and every incoming network packet. In addition to signature-based detection, machine learning techniques are used to proactively counter new threats. Because of its extensive methodology, NIDS is able to detect unusual activity and offer a strong defense against a wide range of possible threats.

A breakthrough age for NIDS has begun with the inclusion of machine learning, notably supervised learning. Through the use of algorithms like Random Forests, Support Vector Machines (SVM), and Decision Trees, NIDS is able to identify patterns, categorize network activity, and make well-informed judgments. By extending NIDS capabilities beyond signature-based detection, these machine learning models allow for adaptable responses to increasingly complex and sophisticated cyber threats. As we set out on an extensive investigation of NIDS, we will cover the systems' design, the many approaches they use, and the difficulties they encounter. We will specifically examine the vital role that supervised learning models such as Random Forests, SVM, and Decision Trees play in strengthening NIDS against a constantly changing array of cyber threats.

The effectiveness of the machine learning models used determines how adaptable NIDS is to a constantly changing threat scenario. A key

technique in supervised learning, decision trees offer an organized framework for generating decisions based on input information. Decision Trees are useful in network intrusion detection systems (NIDS) because they may identify trends in network traffic. They are useful tools in the complex process of anomaly identification because of their interpretability and flexibility. Another key component of supervised learning, support vector machines (SVM), perform exceptionally well in classification problems by identifying the best hyperplanes to divide data into discrete classes. SVM becomes an effective technique for differentiating between legitimate and malicious network activity when used in conjunction with NIDS. SVM is a vital component in improving the precision and dependability of intrusion detection due to its capacity to manage complicated datasets and adjust to a variety of situations. NIDS is further enhanced by the use of Random Forests, an ensemble learning technique. Random Forests reduce the possibility of overfitting and improve the system's capacity for generalization by fusing several Decision Trees. When managing the varied and dynamic nature of network traffic, this ensemble method shows to be especially beneficial. Random Forests make NIDS more resilient in the complex dance of cyber threats by offering a strong and flexible protection system.

Although supervised learning has been successfully included into NIDS, there are still issues. Choosing pertinent features from datasets and finding real-time, labeled traffic data are two challenges in building flexible and efficient intrusion detection systems. The dynamic nature of cyber threats highlights the necessity for adaptive feature selection methodologies. The labor-intensive process of creating labeled datasets continues to be a bottleneck, requiring novel solutions to stay up with new attack scenarios. The incorporation of supervised learning models, including Random Forests, SVM, and Decision Trees, clearly represents a significant advancement in cybersecurity as we navigate the complex landscape of NIDS. These models support NIDS against the tactical moves made by cyber attackers because they have the ability to extract insights from historical data and adapt to new threats. We will go more into the NIDS architecture in the upcoming sections, dissecting its various components and explaining how supervised learning and email-alert enabled intrusion detection work together.

II. LITERATURE REVIEW

By utilizing the Apache Spark Big Data platform, the Spark-Chi-SVM model presented in [1] offers a revolutionary method of intrusion detection in the era of big data. The model implements an SVM classifier and uses the Chi Sq Selector for feature selection, resulting in great performance and shorter training times. The study highlights the necessity for additional investigation into the scalability of the suggested approach while demonstrating feasibility with the KDD99 dataset. In order to differentiate between Signature-Based and Anomaly Detection methods, the review discussed in [2] offers a thorough overview of machine learning techniques in network intrusion detection. Many classification techniques are covered, such as RandomForest, SVM, Hoeffding Tree, Naive Bayes, and Extreme Learning Machine. One significant drawback identified by the evaluation is the difficulty of changing attack scenarios during feature selection. A corporate network traffic dataset is used in [3] to train SVM and RandomForest models for the purpose of detecting network breaches. The research presents a novel feature selection strategy and achieves good accuracy for different sorts of attacks. Nevertheless, there are certain drawbacks, such as the dataset's assumption to cover every scenario of an incursion and its failure to handle encrypted network traffic.

The Isolation Forest technique for network anomaly detection is the subject of the study in [4], which achieves an impressive AUC score of 98.3%. The study highlights the algorithm's efficacy but also recognizes class imbalance as a major drawback and suggests several ways to make it better. [5] presents an ensemble model for intrusion detection based on RandomForest that has excellent accuracy and adaptability. While admitting the common drawback of possible false positives or false negatives, the study assesses the ensemble model's efficiency. K-Means clustering, the XGBoost classification model, and feature selection are combined in [6] to create an anomaly-based intrusion detection system. Although no specific limits are given, the hybrid model shows efficiency and good effectiveness in identifying different types of intrusions. [7] offers a two-phase IoT-specific intrusion detection system that uses an unsupervised elliptic envelope for anomaly detection and Naive Bayes for data classification. The main drawback of the study is its reliance on high-quality data, but it also highlights efficiency, high accuracy,

and IoT adaptability. Deep learning, more specifically deep neural networks (DNNs), is used by a network intrusion detection system in [8] to identify and categorize network threats. The study emphasizes how the system can learn to adapt to changing threats, but it also recognizes that the system's primary weakness is its reliance on the quality of the training data. The emphasis of [9] is on the changing nature of network intrusion attempts and the function of intrusion detection systems (IDS). The text delves into a variety of machine learning methodologies, such as classification, clustering, genetic algorithms, and autoencoder deep learning. Utilizing the KDD Cup'99 dataset, the research highlights the special application of deep learning and genetic algorithms by achieving an astounding 99% accuracy with low false positives. Advantages include a thorough examination of intrusion detection techniques with an emphasis on accuracy improvement; limitations include changing attack scenarios and dataset constraints. Using feature selection, K-Means clustering, and the XGBoost model, [10] offers an anomaly-based intrusion detection system that achieves an accuracy of 84.41%. The model outperforms other models, including deep neural networks, using just 61.47% of the original information. The suggested solution demonstrates competitive performance metrics by combining techniques for effective intrusion detection in an efficient manner. Root to Local, Denial-of-Service, Probe, and Unauthorized to Root incursions are the main topics of the study. The model's advantages and distinct methods make it useful in network security even while its limits aren't addressed directly.

III. METHODOLOGY

Our program consists of three main parts that work together cohesively: a sophisticated machine learning (ML) model, a robust Flask server, and a user-friendly web interface. As the user's entry point, the web interface offers a simple way for them to engage with and manage the application. The Flask server serves as the main backend orchestrator, coordinating interactions between the ML model and the web interface. One of the main parts, the machine learning model, uses sophisticated algorithms for interpreting and assessing data in order to generate predictions or judgments. These elements work in unison to provide a strong and user-focused application experience.

A. Conceptual Framework

The proposed network security system design includes a multi-layer process that starts after the firewall. Once the firewall is up and running, the next step is to deploy a network scanner. This scanner carefully inspects every data packet passing through the network, capturing relevant information and compiling it into a comprehensive file. The generated files serve as input to the data preprocessor, a key component of the system.

Data preprocessors transform raw data to produce machine-understandable files. This process extracts real-time features from scanned data packets to create a structured representation suitable for machine learning analysis. The machine-readable file contains all relevant functionality derived from the scanning process. This refined data is passed to the web interface, which acts as a bridge to subsequent stages of the system.

Once on the web interface, the processed data is seamlessly transferred to the machine learning component. Here, advanced algorithms such as decision trees, support vector machines, and random forests analyze the data and detect potential intrusions and security threats. The machine learning phase is critical to adaptively and intelligently respond to emerging cyber threats and represents a paradigm shift from traditional rule-based systems.

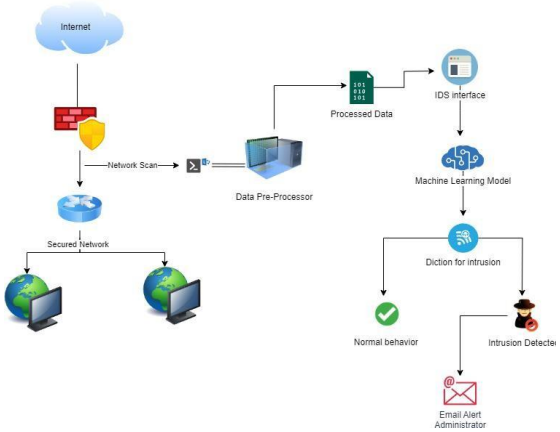

Fig.1. Flow of the proposed methodology

After the machine learning analysis, the system enters the detection phase, where it evaluates the analyzed data for evidence of security breaches or anomalies. When a potential threat is detected, the system initiates an immediate response, which involves generating an email notification. This

automated alert acts as a proactive measure, notifying relevant stakeholders or security managers of identified security risks.

B. Machine Learning Workflow

In our machine learning approach, we integrated three different models for training and testing purposes: Decision Tree (DT), Random Forest (RF), and Support Vector Machine (SVM). Firstly, the model was trained using the entire set of 41 features. To streamline the model and increase efficiency, we implemented recursive feature elimination (RFE), an iterative approach that systematically removes unimportant features while maintaining consistent accuracy throughout each iteration. The goal was to obtain a reduced model with comparable accuracy to the original 41-function configuration.

A machine learning flow starts with raw data containing labels representing 20 to 25 different types of attacks. To simplify these diverse attacks, we introduce the concept of attack classes and classify them into probe attacks, user-to-root attacks (U2R), root-to-local attacks (R2L), and distributed denial of service attacks. This step involves assigning attacks to their respective attack classes and optimizing the dataset. This comprehensive data preprocessing phase laid the foundation for subsequent analyses.

While preprocessing the data, the three statistical properties which were found to be important included services, flags, and protocols. These functions were split into training and testing data sets to provide targeted input to the subsequent modelling phase. Each of these datasets enriched with 41 features was then fed separately to three machine learning models.

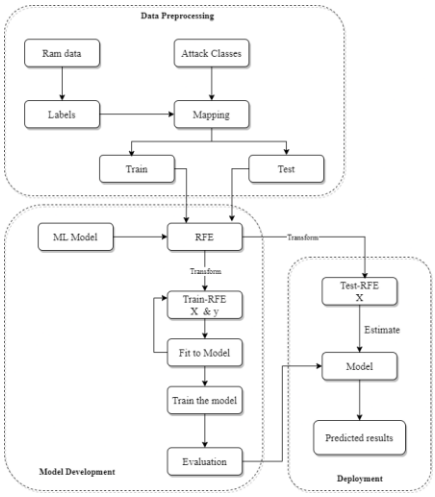

Fig.2. Machine learning work flow

The next step was to apply Recursive Feature Elimination (RFE) to each model to systematically narrow down the features to 15. The transformed features were then integrated into the training set (X and Y), and the test data was scaled down accordingly considering only X, as Y is to be predicted. This careful feature selection process was performed on all the three models.

Once feature selection was complete, the 15 feature datasets went through a final data preprocessing phase and were integrated into their respective models for training. The model was evaluated against the corresponding model based on the original 41 features. We combined the performance metrics of the 41-feature model and the 15-feature model to produce comparable results. This comprehensive approach enabled the optimized model with reduced functionality to maintain similar performance to the original configuration.

Finally, the model was used for basic predictions and produced results containing meaningful insights for intrusion detection.

C. Key Components

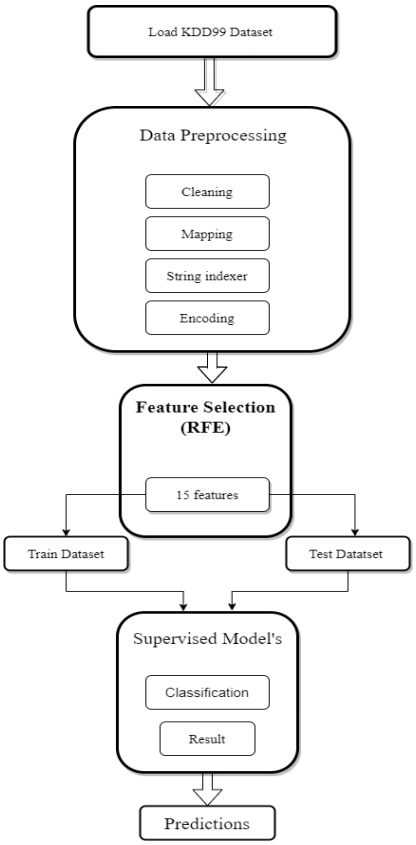

Fig.3. Data preparation, RFE and model training

The initial stage of the machine learning model workflow uses the KDD-cup 99 dataset. The first phase of data preparation consists of three steps: mapping, string indexing, and encoding. We focus on encoding categorical data, especially these two important features: flags and services. This thorough preprocessing phase ensures that the dataset is ready for further exploration. After processing, the data goes through an important step called feature selection, which carefully selects 15 features from the dataset. Selecting the most relevant attributes for your investigation will optimize the efficiency and accuracy of your model. After selecting the features, the dataset is split into separate training and testing sets. This separation is necessary to train a machine learning model on a subset of data and evaluate its performance on an independent subset. A core component of the machine learning workflow is the use of supervised machine learning classification. This method provides pre-processed, feature-selected data to the model, allowing it to identify patterns, correlations, and classifications within the dataset. This phase performs a thorough analysis that reveals how the data was classified based on the specified criteria.

D. Threat Analysis and Notification

The user's interaction with the system begins with a login process that validates access credentials. After approval, the user provides input data that goes through the critical stages of feature removal. During this phase, a machine learning model is trained to identify the 15 most relevant features, which are carefully selected and stored by the system.

After the features are removed, the input data is transformed into a structured data frame, which is the format that the machine learning model needs to understand. For categorical data, this transformation process is particularly important as it ensures smooth integration into later phases of the system. If the encoding is successful, the encoded data frame is reindexed.

Reindexing ensures that the input data matches the continuous patterns discovered when training the machine learning model. The newly indexed data frame is forwarded to the prediction engine to be synchronized with the patterns learnt by the model. The system carefully scans the data provided by this module to detect potential cyber threats. Based on predetermined attack classifications, machine learning models record these threats and perform a complete analysis. When a risk is detected, the system will immediately trigger an alert mechanism and an email notification will be sent to the user's registered email address. If no threats are detected, the system does not send email notifications, eliminating unnecessary user intervention.

Email alerts serve as comprehensive reports and provide detailed information about the types of attacks identified. By systematically presenting attack types and classifications, it provides users with a comprehensive picture of their security situation. Additionally, email notifications provide specific information such as threat volume, attack type, and attack classification.

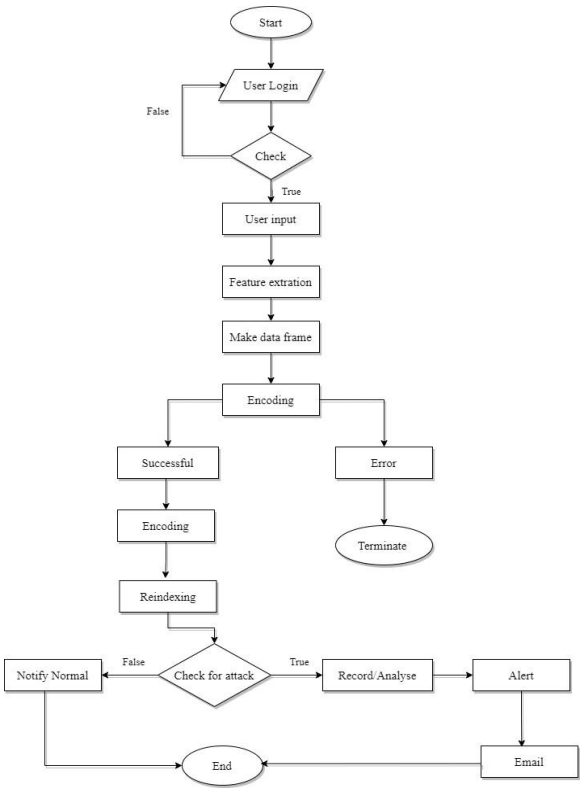

Fig.4. Analysis for threats and email-alerting

In addition to serving as a useful documentation tool, the comprehensive data provides users with actionable insights, allowing them to quickly and intelligently respond to potential security threats.

E. Automated Threat Detection and Reporting

After a successful login, the user will be taken to the home page with clear options in the main header. The 'Home' option allows the user to return to the home page, 'Detect' starts the network analysis process, and 'Logout' securely ends the session. Selecting “Detect” or launching from the home page ” Start” takes the

user to the Network Analysis page, which is a hub for input and analysis.

Users are prompted to enter their entries manually or via a CSV file. For CSV file input, the network scanner generates a comprehensive file containing network details that is seamlessly integrated into the interface. Triggering the Detect button starts a complex backend process. The collected data is passed to a machine learning (ML) model from manual input or a CSV file. This ML model has robust algorithms and performs careful analysis of your dataset. For CSV files, the model classifies each device's data and corresponding records, subjecting manual input to targeted analysis.

The core functionality of ML models is to detect potential intruders in the provided data. If the analysis results do not indicate a threat, the system remains in its normal state.

Conversely, if a threat exists, the system quantifies and categorizes the identified threat or attack. This detailed information is used to create comprehensive reports. The generated reports serve as valuable output and provide insight into the type and number of threats and attacks identified.

This report is automatically sent to the user and provides a clear understanding of the security situation based on the analysis performed by the system. Automated reporting mechanisms allow users to quickly receive critical information so they can take informed action against potential security threats.

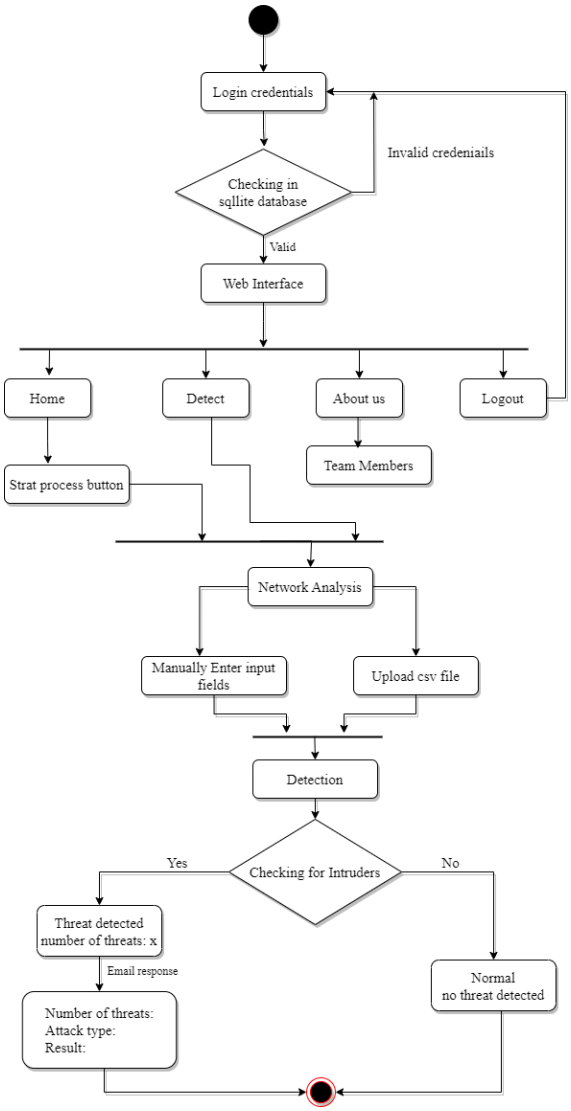

Fig.5. User interface

IV. RESULTS AND DISCUSSION

The email-alert enabled supervised machine learning model, integrated with Network Intrusion Detection System (NIDS), has exhibited notable performance optimization through recursive feature elimination (RFE). After the application of RFE, the model achieved a commendable accuracy of 94%, demonstrating its ability to efficiently discern and prioritize relevant features. Notably, the feature count was successfully reduced to 15, showcasing the model's adaptability in focusing on the most significant aspects of network traffic analysis. Although the initial model boasted a slightly higher accuracy of 99% with 41 features, the trade-off of reduced complexity and enhanced interpretability

achieved through RFE makes the 94% accuracy with 15 features an attractive choice. This streamlined feature set not only contributes to computational efficiency but also facilitates a clearer understanding of the identified threats, reinforcing the model's overall effectiveness in threat detection and email notification generation.

The 15 Selected Features are as follows

Index(['src_bytes','wrong_fragment','hot',
'lnum_compromised','count',
'srv_count', 'same_srv_rate',
'dst_host_diff_srv_rate',
'dst_host_same_src_port_rate',
'dst_host_srv_diff_host_rate',
'dst_host_serror_rate','service_eco_i',
'service_ftp_data',
'service_private','flag_RSTR'],
dtype='object')

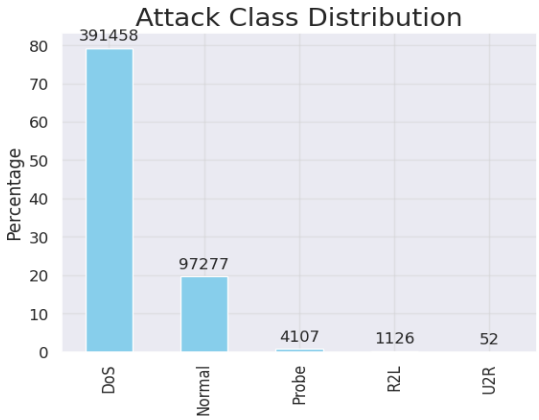

Fig.6. Graphic visualization to analyze the percentage and categories of attacks detected.

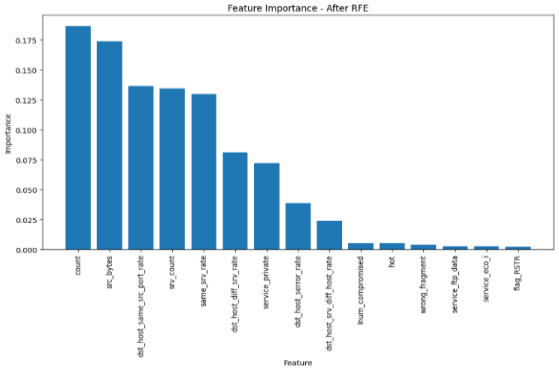

Fig.7. Graph featuring the importance of various features after RFE

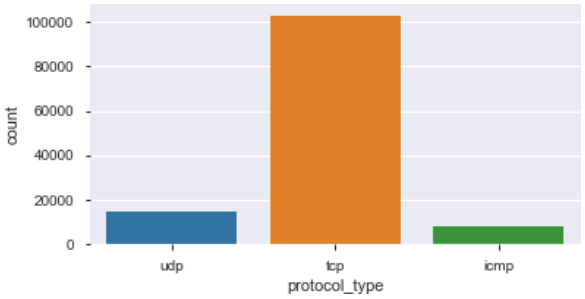

Fig. 8. Visualization depicting the protocols that most attacks are susceptible to.

attack_class	
DoS	391458
Normal	97277
Probe	4107
R2L	1126
U2R	52

Fig. 9. Count of the attack class distributions

Selected Features	Description
src_bytes	numberofdatabytesfromsourceto destination
wrong_fragme nt	numberof``wrong"fragments
hot	numberof``hot"indicators
lnum_compro mised	numberof``compromised"conditions
count	numberofconnectionstosamehostin past two seconds
srv_count	numberofconnectionstosameservice in past two seconds
same_srv_rate	%ofconnectionstothesameservice
dst_host_diff_s rv_rate	%ofconnectionstodifferentservices
dst_host_same _src_port_rate	%ofconnectionsservedbythe destinationhost
dst_host_srv_d iff_host_rate	%ofconnectionstodifferenthosts
dst_host_serror _rate	%ofconnectionsthathave``SYN"errors
flag	normalorerrorstatusoftheconnection

service	networkserviceonthedestination,e.g., http, telnet, etc.
attack_class	mappingofdifferentattackintoclasses of R2L, U2R, Probe, DoS, Normal

Table1. Selected Features Description

ModelEvaluation				
		ecision Tree	Random Forest	Support Vector Machine
lidationmean Score		0.9996	0.9997	0.9991
ModelAccuracy		0.9999	0.9999	0.9993
With41 Feature Accuracy	Train	0.9995	1	1
	Test	0.9989	0.9999	0.9963
With 15 Feature Accuracy	Train	0.9985	0.9993	0.9994
	Test	0.9939	0.9999	0.9981
F1Score		0.9945	0.9294	0.9981

Table2. Model Evaluation Report

Fig.10. Start button after log into kick start the process

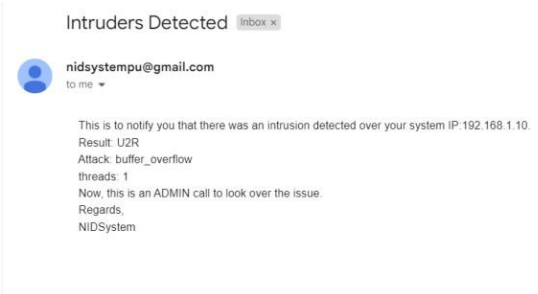

Fig.11. Email-alert received by the user-admin after analysis of their network

V. LIMITATIONS

Our NIDS system has limitations that include inability to detect intrusions in real-time as it operates on a static dataset. This hinders immediate threat response, impacting the system's ability to address dynamic and evolving network security challenges effectively.

VI. FUTURE WORKS

a) In the future, we aim to enhance our NIDS system to detect intrusions in real-time. This improvement ensures a prompt response to dynamic network security challenges, addressing the limitations of the current static dataset operation.

b) We plan to integrate a dashboard for project analysis, providing a user-friendly interface to enhance monitoring and assessment capabilities in our NIDS system upgrade.

c) Our future work entails extending the NIDS project beyond supervised algorithms. We plan to integrate unsupervised and semi-supervised algorithms, enhancing the system's adaptability and enabling more comprehensive intrusion detection. This evolution will result in a more robust and versatile network security solution.

CONCLUSION

In summary, this study highlights the important role of supervised machine learning models in enhancing network intrusion detection systems (NIDS). Examining decision trees, support vector machines (SVMs), and random forests demonstrates their effectiveness in adapting to the dynamic nature of cyber threats. A careful methodology using data pre-processing and recursive feature elimination (RFE) highlights the possibility of optimizing the feature set without compromising recognition accuracy. This adaptability is critical when facing an ever-evolving cybersecurity environment. Additionally, a comparative analysis between a model using 41 features and a model using a reduced set of 15 features provides valuable insights. This study suggests that the model maintains a commendable level of performance even with fewer features. This finding is promising for real-world applications where resource efficiency is important. This research will contribute to the optimization of NIDS and provide a nuanced understanding of how machine learning algorithms can be used for effective

intrusion detection. As the threat landscape becomes more complex, the need for a sophisticated and adaptable NIDS becomes more apparent. This study serves as a starting point in this direction by highlighting the strengths and limitations of different models of supervised learning. The continued evolution of NIDS through advances in machine learning promises to proactively defend against emerging cyber threats and ultimately improve the security posture of network systems. In an ever-expanding digital ecosystem, the implications of the study go beyond the specific models discussed and highlight the broader paradigm of the use of machine learning in cybersecurity. NIDS's continued development is based on a collaborative effort, incorporating new technologies and innovative methods to stay one step ahead of adversaries in complex cybersecurity environments.

REFERENCES

[1] S.M. Othman, F.M. Ba-Alwi, N.T. Alsohybe, and A. Y. Al-Hashida, "Intrusion detection model using machine learning algorithm on Big Data environment," J Big Data, vol. 5, no. 1, Dec. 2018, doi: 10.1186/s40537-018-0145-4.

[2] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, Dec. 2019, doi: 10.1186/s42400-019-0038-7.

[3] T. Rakshe, V. Jijabai, and V. Gonjari, "Anomaly based Network Intrusion Detection using Machine Learning Techniques." [Online]. Available: www.ijert.org

[4] Institute of Electrical and Electronics Engineers and PPG Institute of Technology, *Proceedings of the 5th International Conference on Communication and Electronics Systems (ICCES 2020): 10-12, June 2020.*

[5] M.A. Hossain and M.S. Islam, "Ensuring network security with a robust intrusion detection system using ensemble-based machine learning," *Array*, vol. 19, Sep. 2023, doi: 10.1016/j.array.2023.100306.

[6] J. Han and W. Pak, "High Performance Network Intrusion Detection System Using Two-Stage LSTM and Incremental Created Hybrid Features," *Electronics (Switzerland)*, vol. 12, no. 4, Feb. 2023, doi: 10.3390/electronics12040956.

[7] M. Vishwakarma and N. Kesswani, "A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection," *Decision Analytics Journal*, vol. 7, Jun. 2023, doi: 10.1016/j.dajour.2023.100233.

[8] L. Ashiku and C. Dagli, "Network Intrusion Detection System using Deep Learning," in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 239–247. doi: 10.1016/j.procs.2021.05.025.

[9] Galgotias University. School of Computing Science and Engineering, Institute of Electrical and Electronics Engineers. Uttar Pradesh Section, and Institute of Electrical and Electronics Engineers, *2018 4th International Conference on Computing Communication and Automation (ICCCA).*

[10] Institute of Electrical and Electronics Engineers, *Proceedings, 2018 Sixteenth International Conference on ICT and Knowledge Engineering, November 21-23, 2018, Bangkok, Thailand.*