iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VIII **Month of publication:** August 2025

DOI: https://doi.org/10.22214/ijraset.2025.73531

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VIII Aug 2025- Available at www.ijraset.com

Cyber Threat Detection Using Machine Learning: A Performance Evaluation Approach

Dr. Levina Tukaram[1], Khadanand Chaudhary[2], Nitesh Jha[3], Bibek Pangeni[4], Vision Gaihre[5]
[1, 3, 4, 5]*Department of Computer Science and Engineering, KNSIT Bengaluru*
[2]*Department of Information Science and Engineering, KNSIT Bengaluru*

Abstract: ***Now a day world has come all dependent on cyberspace for every aspect of daily living. The use of cyberspace is increasing with each day by day. The world is spending most of the time on the Internet than ever ahead. As a result, the pitfalls of cyber pitfalls and cybercrimes are increasing day by day. The term' cyber trouble' is applicable to as the illegal exertion performed using the Internet. Cybercriminals are changing their ways with time to pass through the wall of protection. Conventional ways are not able of detecting zero- day attacks and sophisticated attacks. There fore, far, stacks of machine literacy ways have been developed to descry the cybercrimes and battle against cyber pitfalls. The ideal of this exploration work is to present the evaluation of some of the extensively used machine literacy ways used to descry some of the most threatening cyber pitfalls to the cyberspace. Three primary machine literacy ways are substantially delved, including deep belief network, decision tree and support vector machine. We've presented a brief disquisition to gauge the performance of these machine literacy ways in the spam discovery, intrusion discovery and malware discovery grounded on constantly used and standard datasets.***

Keywords: Cyber Threat; Cybercrime; Performance Evaluation; Machine Learning Application; Intrusion Detection System; Malware Detection; Spam Classification

I. INTRODUCTION

Cyber trouble discovery is one of the most pivotal factors of a comprehensive cybersecurity strategy. It involves relative potentially dangerous conditioning within a system or network before they can begat damage. Effective trouble discovery helps help data loss, fiscal damage, service interruptions, and reputational detriment. Traditional discovery systems frequently calculate on hand-grounded or rule- grounded styles which can only identify known pitfalls and struggle with unknown or evolving attack patterns (e.g. zero- day attacks).

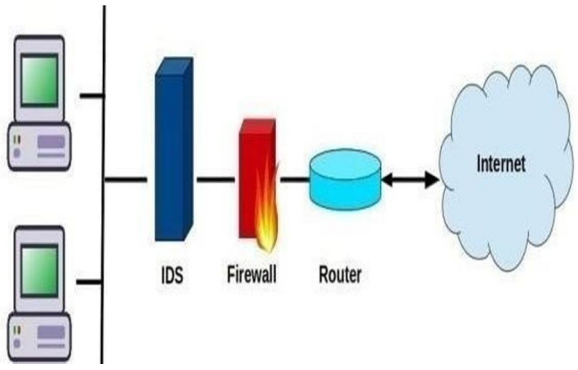


Fig 1: Cyber Threat Detection

II. LITERATURE SURVEY

Several studies have explored machine literacy(ML) models for intrusion discovery systems(IDS).
The following are significant benefactions
Tavallaee et al. (2009) – Proposed the NSL- KDD dataset, an bettered interpretation of the outdated KDD'99 dataset, reducing redundancy and imbalance issues. Thisdataset came the foundation for numerous ML- grounded IDS evaluations.
Amor et al. (2004) – Compared Decision Trees (DT), Support Vector Machines(SVM), and Naïve Bayes for intrusion discovery. They set up that DTs handed better interpretability while SVMs delivered advanced Delicacy for multiclass intrusion discovery.

©IJRASET: All Rights are Reserved

Shone et al. (2018)– Introduced a mongrel deep literacy model using non-symmetric deep auto encoders and arbitrary timbers for point birth and bracket. This approach showed promising performance for detecting sophisticated attacks.

Vinayakumar et al. (2019) – Delved convolutional neural networks(CNNs) for network intrusion discovery. Their deep literacy frame demonstrated bettered discovery rates for new attacks over traditional ML models

Kumar and Singhal (2021)– Applied ensemble literacy combining Random Forest and XGBoost for cyber trouble discovery. Their work emphasized the significance of point engineering and tuning for high- delicacy results. following are significant contributions:

III. PROBLEM DEFINITION

Incapability to descry Unknown pitfalls hand-grounded systems can only descry pitfalls that have formerly been linked and entered. They're ineffective against zero- day attacks or new pitfalls with no being hand.

High False Positive Rate Rule- grounded systems may inaptly classify licit conditioning as vicious due to exorbitantly strict or outdated rules, leading to alert fatigue for security brigades. Limited Adaptability Traditional systems don't learn from new data. They can not evolve or acclimatize to changing attack strategies over time.

IV. PROPOSED SYSTEM

The proposed system utilizes machine literacy (ML) algorithms to make an intelligent, adaptive, and automated cyber trouble discovery model. Unlike traditional hand-grounded styles, this system is designed to earn patterns from literal network data, descry both known and unknown pitfalls, and continuously ameliorate as new data becomes available.

A. Crucial Features Of The Proposed System Data Driven Discovery

The system is trained on labeled dataset containing both benign and vicious Network exertion. This allows the model to learn the behavioral patterns associated with cyberattacks.

Bracket Using Random Forest Random Forest is an ensemble fashion that constructs multiple decision trees and summations their labors to make the final decision.

Multiple ML Algorithms Four popular bracket algorithms are enforced and compared — Random Forest, Support Vector Classifier(SVC), K- Nearest Neighbors(KNN), and Logistic Retrogression.

Each model is estimated using performance criteria like delicacy, perfection, recall,andF1- score.

Automated trouble Identification. Once trained, the system can automatically classify incoming network business in real- time as either vicious or benign without homemade rule creation or updates.

1) Scalability and Rigidity: The ML models can be retrained with new data, allowing the system to acclimatize to arising pitfalls and maintain high discovery delicacy.

2) End-to-End Architecture: The system processes raw network business, applies PCA to prize significant features, and utilizes Random Forest to classify business as normal or attack. also, a Beaker- grounded web dashboard is stationed to cover IDS cautions in real- time, furnishing an intuitive visualization of detected intrusions.

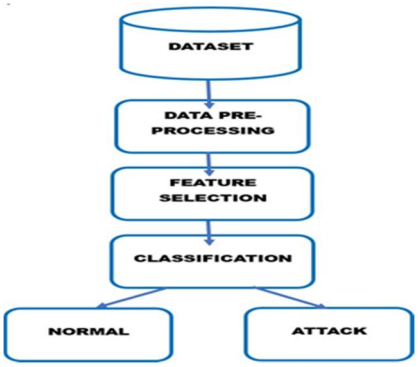


Fig 2: Proposed System Flow Chart

B. Dataset

The NSL- KDD dataset was chosen due to its wide acceptance in IDS exploration and its advancements over the original KDD'99 dataset, addressing redundancy and imbalance issues.

No	Features	Form of value	No	Features	Form of value
1	Duration	Integer	22	is_guest_login	Integer
2	protocol_type	Nominal	23	count	Integer
3	Service	Nominal	24	srv_count	Integer
4	Flag	Nominal	25	serror_rate	Float
5	src_bytes	Integer	26	srv_serror_rate	Float
6	dst_bytes	Integer	27	rerror_rate	Float
7	land	Integer	28	srv_rerror_rate	Float
8	wrong_fragment	Integer	29	same_srv_rate	Float
9	urgent	Integer	30	diff_srv_rate	Float
10	hot	Integer	31	srv_diff_host_rate	Float
11	num_failed_logins	Integer	32	dst_host_count	Float
12	root_shell	Integer	33	dst_host_srv_count	Float
13	num_compromised	Integer	34	dst_host_same_srv_rate	Float
14	roots_hell	Integer	35	dst_host_diff_srv_rate	Float
15	su_attempted	Integer	36	dst_host_same_src_port_rate	Float
16	num_root	Integer	37	dst_host_srv_diff_port_rate	Float
17	num_file_creations	Integer	38	ddst_host_serror_rate	Float
18	num_shells	Integer	39	dst-_host_srv_serror_rate	Float
19	num_access_files	Integer	40	dst_host_rerror_rate	Float
20	num_outbound_cmds	Integer	41	dst_host_srv_rerror_rate	Float
21	Is_host_login	Integer			

Fig 3: NSL-KDD dataset structure

V. SYSTEM ARCHITECTURE

A Data Flow Diagram(DFD) represents how data moves within system, helping fantasize the commerce between colorful factors like external realities, processes, and data stores.

For Cyber trouble Discovery using Machine literacy, the DFD outlines the inflow from network business input to trouble vaticination and response.

The following is a step- by- step explanation of the methodology launch – The process begins.

Network Traffic Collection: Real- time data is captured from network packets, including details like IP addresses, harborage figures, packet sizes, and time stamps. This data forms the base for analysis.

1) Data Preprocessing: The raw business is gutted and structured. Preprocessing includes filtering in applicable data, converting packet formats into structured logs, and rooting useful features for analysis.

2) Feature Selection: Important attributes are selected (e.g., source IP, destination IP, protocol, packet size) to reduce noise and focus the model training on significant patterns.

3) Model Selection: A suitable machine learning model is chosen such as SVM, Random Forest, or KNN, depending on the dataset and detection goals.

4) Model Training: The selected algorithm is trained on labeled traffic data (normal vs malicious).

5) The model learns to distinguish patterns that signify threats.

6) Hyper parameter Tuning: The Key parameters like learning rate, depth, and number of estimators are fine-tuned to improve detection performance.

7) If YES → The model is saved for deployment and ready to flag threats in real time.

8) If NO → The process loops back to tuning and retraining the model until

9) Deployment & Real-Time Monitoring – The finalized model is deployed into the network security system. It monitors live traffic and triggers alerts when a potential threat is detected.

10) Alert Generation-- When an anomaly or intrusion is detected, the system generates notifications or logs the event for the security team to investigate.

11) Stop – The detection cycle completes. Monitoring continues in real-time as a continuous loop.

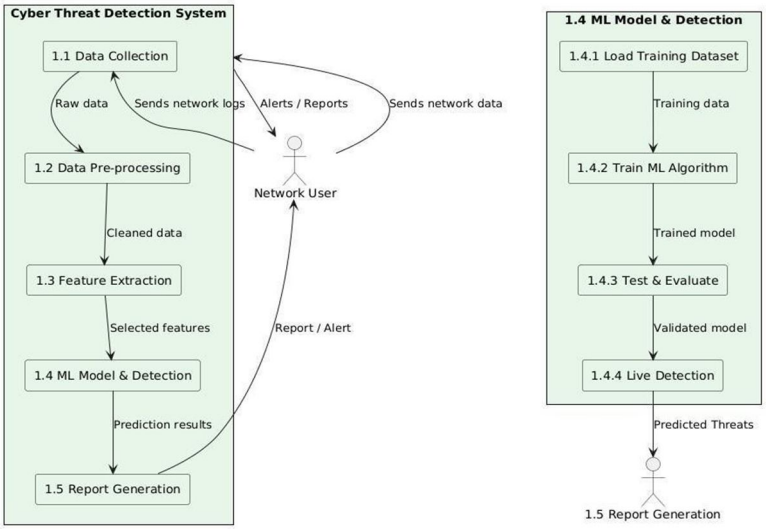


Fig.4: Dataflow diagram

VI. IMPLEMENTATION

The integration of machine learning into IDS has opened new avenues for effective threat detection. Machine learning models such as Decision Trees, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Artificial Neural Networks (ANN) have been extensively used. Decision Trees are appreciated for their interpretability, while SVMs are valued for handling high-dimensional data. KNN provides good results but can be computationally expensive for large datasets. Neural Networks, particularly Deep Learning models, offer high detection capabilities but require significant computational resources. However, the performance of these models depends heavily on the quality and dimensionality of the input features.

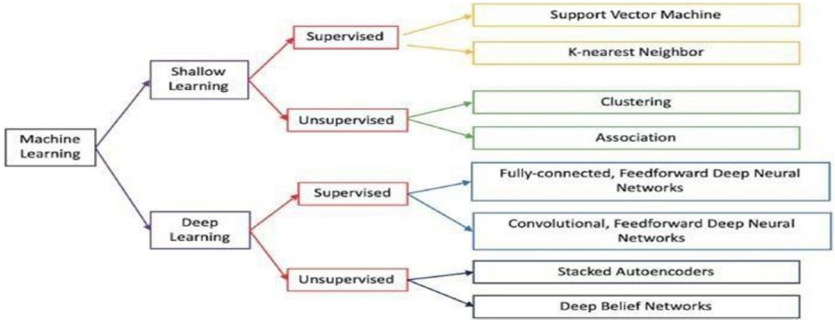


Fig 5: Machine Learning Techniques in IDS

Random Forest is an ensemble machine learning method that constructs multiple decision trees and merges their outputs to make final predictions. It excels in handling high-dimensional datasets and is robust against overfitting. Random Forest classifiers have been successfully applied to intrusion detection tasks, where they outperform many other machine learning algorithms in terms of both accuracy and stability. The model's inherent ability to handle noisy data and its support for feature importance estimation make it an ideal candidate for building scalable IDS.

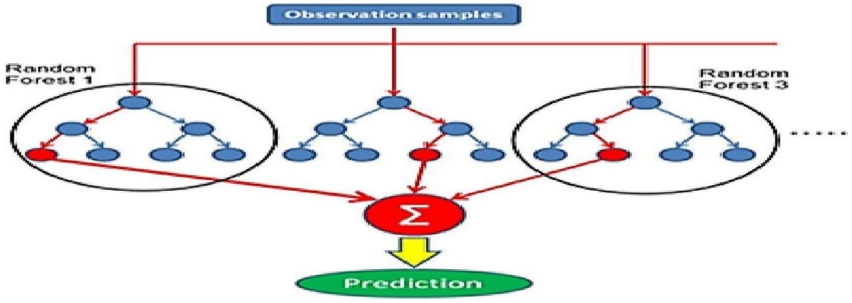


Fig 6: Random Forest Model.

A. Models and Algorithms for IDS

SVM and Naïve Bayes for IDS:
Adaptive Boosting with Feature Selection:
Artificial Neural Networks (ANNs):
Extreme Learning Machine (ELM):
Deep Learning for Feature Extraction:
Fuzzy Rule-Based Feature Selection:

B. Data Preprocessing Algorithm

The raw NSL-KDD dataset required several preprocessing operations:
Step 1: Data Cleaning
The dataset was first cleaned by removing irrelevant columns (like difficulty level) and checking for any missing values. Missing values, if any, were handled appropriately — either by imputing mean values or removing incomplete records.
Step 2: Encoding Categorical Variables
Several features, such as protocol type, service, and flag, are categorical. To enable machine learning models to process them, these were transformed into numerical values using Label Encoding and One-Hot Encoding techniques, preserving the semantic meaning.
Step 3: Feature Scaling
Feature scaling ensures that all features contribute equally to the model's performance. Standardization (zero mean and unit variance) was applied to the numerical attributes to bring them onto a common scale.
Step 4: Dataset Splitting
The cleaned and pre-processed dataset was then split into an 80:20 ratios — 80% used for training the model and 20% reserved for testing and evaluation.
This comprehensive preprocessing pipeline ensured that the dataset was well-prepared for dimensionality reduction and subsequent classification.

C. Feature Selection in Cyber Threat Detection System

Feature selection is a critical step in building an effective cyber threat detection system using machine learning. It involves identifying the most relevant features from the dataset that contribute significantly to distinguishing between benign and malicious network activities. Proper feature selection enhances model performance, reduces overfitting, shortens training time, and improves interpretability.

D. Model Building using Random Forest Algorithm

After dimensionality reduction, a Random Forest Classifier was employed to perform the classification task.
Step 1: Model Initialization
A Random Forest model was initialized with an initial set of parameters, such as:
Number of trees (n_estimators) = 100
Maximum tree depth = None (allow trees to grow fully)
Minimum samples split = 2

Step 2: Hyperparameter Tuning
To optimize the classifier's performance, GridSearchCV was used to perform hyperparameter tuning. Parameters such as max_features, n_estimators, and min_samples_split were systematically varied and evaluated using cross-validation.

Step 3: Model Training
The Random Forest model was trained on the PCA-transformed training dataset. The ensemble method allowed the model to average the predictions from multiple decision trees, thereby improving generalization and robustness.

Step 4: Feature Importance Analysis
Random Forest naturally provides feature importance scores. This information was used to further interpret the model and verify that the most significant principal components corresponded to the key intrusion patterns.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

E. Observed values

High classification accuracy (96.78%).
Low error rate (0.21%).
Robustness against overfitting.
Capability to handle class imbalances effectively.
malicious activity.
User Authentication: To ensure that only authorized personnel can access IDS alerts and reports.
Real-Time Alerts: Displayed on a dynamic dashboard, highlighting detected attacks immediately.
Scalability: Designed in such a way that it can be connected to live packet capture tools for real-time monitoring.

F. Experimental Setup

The experimental setup for evaluating the Cyber Threat Detection model involved the following configuration:
Dataset: NSL-KDD dataset (KDDTrain+ for training and KDDTest+ for testing).
Training and Testing Split: 80% of the data used for training 20% used for testing.
Cross-Validation: 5-fold cross-validation was employed to validate the model and prevent overfitting.
Evaluation Metrics: Accuracy, Precision, Recall, F1-Score, Confusion Matrix, ROC-AUC.
The experiments were conducted in a controlled offline environment. However, the system architecture was designed with scalability in mind, enabling future extension to real-time, live network traffic monitoring environments.
Distribution of flow duration:

G. Observations

The histogram shows a fairly even distribution of flow duration across various intervals.
There are noticeable peaks around 25,000–40,000 and 75,000–85,000, indicating higher occurrences in those ranges.
The blue curve represents a smooth density estimate, showing a slight bimodal trend.
Overall, the flow durations are consistently distributed with no extreme outliers

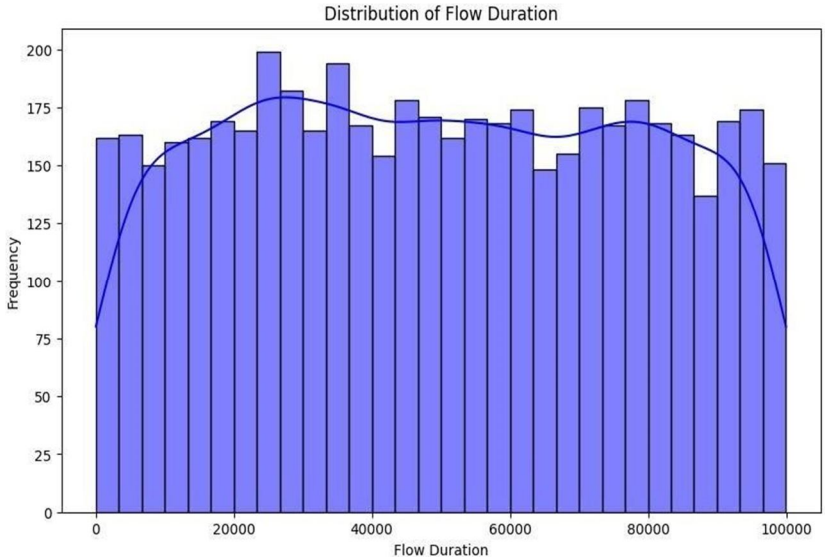


Fig 7: Distribution of flow duration

H. Based on distribution of flow duration

Both BENIGN and MALICIOUS traffic have a similar median Idle Mean, around the 500 mark.
The spread (interquartile range) is also quite similar for both labels, indicating comparable variability.
Both categories show a wide range, from near 0 to over 1000, suggesting possible idle time bursts.
There is no significant difference in Idle Mean between the two labels, making it less effective alone for classification.

I. Performance Comparison

Table 1: Performance Comparison

Model	Accuracy (%)	Precision (%)	Recal (%)	F1 Score (%)
Random Forest	96.2	94.8	95.3	95
SVM	93.5	92.1	90.4	91.2
KNN	89.7	88.3	87.9	88.1
Naïve Bayes	85.2	83.1	84.5	83.8

VII. RESULTS

The home screen of the Cyber Threat Detection system web app, allowing Network Packets to generate results



Fig 8: Home Page



Fig 9: Result Page

Benign – Indicates that the input network packet data is classified as non-malicious based on the user's input.
Malicious – Indicates that the input network packet data is classified as potentially harmful or malicious based on the user's input.
Real-time Prediction: The final model was successfully deployed to accept custom user inputs for key features such as flow duration, total packets, and idle mean, and was able to classify the input as benign.
Improved Accuracy: Based on the entered values, the system correctly identified the network traffic as benign, indicating that the model's predictions align well with expected behavior and demonstrate practical accuracy.
Model Reliability: This successful classification confirms that the trained algorithms are not only theoretically sound but also practically reliable for basic real-time detection tasks.

VII. CONCLUSION

The increasing sophistication and frequency of cyberattacks have made network security a top priority. This Paper focused on detecting cyber threats by classifying network traffic as *benign* or *malicious* using various machine learning algorithms, including Logistic Regression, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). Through data pre-processing and feature selection, relevant network traffic features such as flow duration, packet counts, and statistical measures were used to train and evaluate the models. Among the algorithms tested, Random Forest emerged as the most effective, offering high accuracy and robustness in classifying threats. The Paper demonstrated that machine learning models, when properly trained and tuned, can play a significant role in enhancing the early detection of cyber threats. It also emphasized the importance of using a balanced and comprehensive dataset to improve detection reliability and reduce false positives/negatives.
While the results are promising, the Paper also has its limitations. Real-time implementation, adaptability to new and unknown threats, and scalability in high-speed networks remain challenges. Future improvements may include integrating deep learning models, using real-time data streams, and expanding the feature set for more accurate threat detection. In conclusion, this Paper provides a foundational approach to applying machine learning in cybersecurity and highlights the potential of data-driven models in strengthening network defense systems.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

A. Scope for Further Optimization

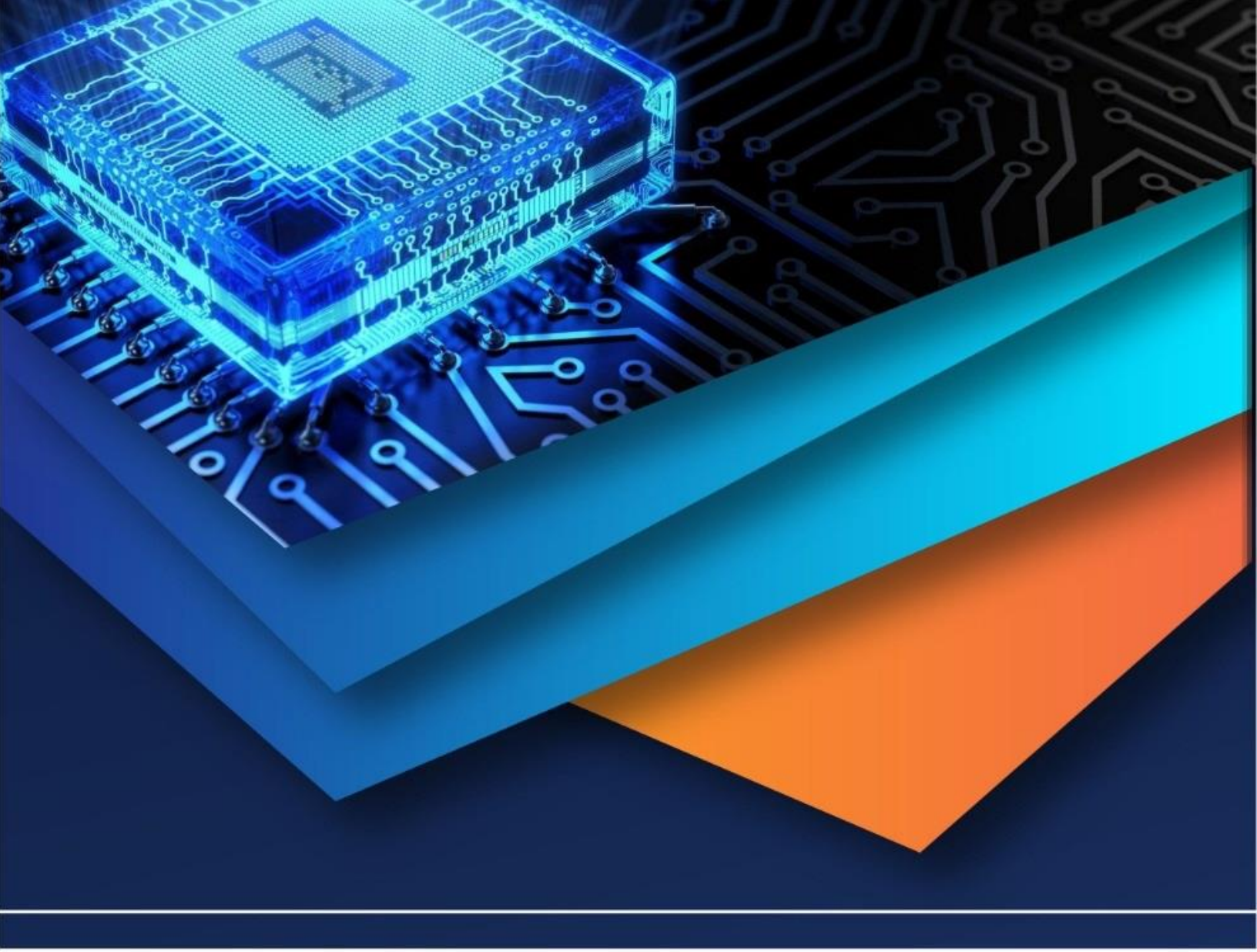
There is room for improvement through real-time detection, deep learning techniques, and more granular feature extraction to enhance threat identification accuracy. Despite preprocessing and feature selection efforts, most models struggled with recall for the minority class (malicious), highlighting the need for further refinement. Additionally, visualizations such as flow duration distribution, ideal mean analysis, and feature importance graphs helped identify patterns, but more advanced techniques may be needed to improve model performance.

Handling Class Imbalance: Implementing resampling techniques like SMOTE or ADASYN to better represent malicious traffic during training. Model Enhancement: Exploring advanced algorithms such as XGBoost, ensemble learning, or deep learning approaches for improved classification. Real-Time Detection: Optimizing the model for faster inference and deploying it in real-time environments for live traffic monitoring.

REFERENCES

[1] S.Ke, "Cyber Threat Detection Using Machine Learning Techniques: A Performance Evaluation Perspective," Journal of Physics: Conference Series, vol. 2113, no. 1, pp. 012074, 2021.
[2] Bipin Kumar Singh, "A Review on Machine Learning Based Cyber Threat Detection Techniques," Journal of Pharmaceutical Negative Results, 2024..
[3] G.Kim, S. Lee, and S. Kim, "A Novel Hybrid Intrusion Detection Method Integrating Anomaly Detection with Misuse Detection," Expert Systems with Applications, vol. 41, no. 4, pp. 1690-1700, 2014.
[4] I.Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," ICISSP, pp. 108-116, 2018.
[5] KDD Cup 1999 Dataset: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[6] D. Gunawan, R. F. Rahmat, A. Putra, and M. F. Pasha, "Filtering Spam Text Messages by Using Twitter-LDA Algorithm," in 2018 IEEE International Conference on Communication, Networks and Satellite (Comnetsat), 2018: IEEE, pp. 1-6
[7] P. Mishra, V. Varadharajan, U. Tupakula, E. S. J. I. C. S. Pilli, and Tutorials, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," vol. 21, no. 1, pp. 686-728, 2018.
[8] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in 2015 International Conference on Signal Processing and Communication Engineering Systems, 2015: IEEE, pp. 92-96.
[9] M. Pradhan, C. K. Nayak, and S. K. Pradhan, "Intrusion Detection System (IDS) and Their Types," in Securing the Internet of Things: Concepts, Methodologies, Tools, and Applications: IGI Global, 2020, pp. 481-497.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)