iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VII **Month of publication:** July 2025

DOI: https://doi.org/10.22214/ijraset.2025.73363

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 VII July 2025- Available at www.ijraset.com

Detection of Threats using Machine Learning Techniques

"To build a system that detects network threats and classifies their attack type using machine learning "

Ramesh Uma Sankar[1], Challa Narasimham[2]
MCA student, IOCL – Chair Professor, Department of Information Technology & Computer Applications, Andhra University College of Engineering, Visakhapatnam, AP

Abstract: ***The rapid increase in cyberattacks poses a significant threat to digital infrastructure, highlighting the need for intelligent and timely threat detection systems. This paper presents a machine learning-based approach for classifying cybersecurity threats using network traffic data. Various supervised learning models, including Logistic Regression, Random Forest, Support Vector Machine (SVM), and XGBoost, are applied and evaluated for their performance in detecting and categorizing network-based attacks. The system processes key features such as packet length, port usage, and protocol behavior to identify malicious patterns and determine attack types. A user-friendly interface is developed using Streamlit to support data upload, preprocessing, prediction, and visualization. Experimental results demonstrate that the proposed models, particularly ensemble methods, provide high accuracy in threat classification, offering an effective and scalable solution for proactive cybersecurity defense.***

Keywords: Cybersecurity, Threat Detection, Machine Learning, Network Traffic Analysis, Attack Classification, Streamlit Interfacer.

I. INTRODUCTION

In the age of digital transformation, organizations and individuals increasingly rely on networked systems for communication, commerce, governance, and everyday operations. As a result, cyber threats have become more pervasive, frequent, and sophisticated, posing significant risks to data privacy, system availability, and overall digital trust. Traditional rule-based or signature-based intrusion detection systems (IDS) often struggle to keep up with the dynamic nature of modern cyberattacks, especially zero-day threats and evolving attack patterns.

To address these limitations, the integration of Machine Learning (ML) techniques into cybersecurity has emerged as a promising solution. ML models can learn patterns from large volumes of network traffic data and detect anomalies or classify malicious behavior based on learned patterns. Such data-driven approaches not only improve the accuracy of threat detection but also enhance the adaptability of security systems against new and previously unseen attacks.

This paper presents a machine learning-based system for the classification of cybersecurity threats using structured network traffic features. The system aims to identify whether an instance of network activity represents a threat and, if so, determine its specific type (e.g., DDoS, Bot, PortScan). The dataset used comprises various features extracted from packet flow metadata, such as packet length, protocol type, port numbers, and byte/packet rates.

Multiple supervised learning algorithms are employed, including Logistic Regression, Random Forest, Support Vector Machine (SVM), and XGBoost, each trained and tested to evaluate their performance in both binary classification (threat vs. no threat) and multiclass classification (type of threat). The models are assessed using standard performance metrics such as accuracy, precision, recall, and F1-score.

To enhance usability, a web-based interface is developed using Streamlit, allowing users to upload traffic datasets, preprocess features, run predictions, and visualize the results in an intuitive environment. This interface bridges the gap between technical models and practical cybersecurity applications, making the system suitable for both researchers and security analysts.

Through experimental validation, the study demonstrates that ML models—especially ensemble-based methods like Random Forest and XGBoost—are highly effective in accurately detecting and classifying cyber threats. The proposed approach contributes to the field by offering a scalable, efficient, and interactive solution that supports proactive cyber defense strategies.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

II. LITERATURE REVIEW

A. Understanding Threat Classification Systems

Threat classification is a key function of modern cybersecurity infrastructure. These systems analyze patterns in network traffic to determine whether the activity is normal or potentially malicious. Effective classification systems not only detect threats but also categorize them into types such as DDoS, Brute force attack, Ransomware and other forms of intrusion. By learning from previously observed behaviors, these systems enhance early detection capabilities and support timely intervention. The use of data-driven classification methods improves the ability to distinguish real threats from benign anomalies, reducing false positives and increasing system resilience.

B. Traditional Intrusion Detection Methods

Conventional cybersecurity systems rely heavily on signature-based detection methods and rule-based filters, which function by identifying predefined patterns of known threats. Tools such as firewalls, antivirus software, and intrusion prevention systems operate using this approach. While effective against well-known attacks, these systems often fail to detect new, sophisticated, or evolving threats. Moreover, their lack of adaptability can lead to high false alarm rates. These limitations have driven a shift toward more intelligent approaches capable of learning from historical data to identify both known and unknown threats.

C. Role of Machine Learning in Threat Classification

Machine learning offers a powerful alternative to traditional static systems by enabling automatic learning and pattern recognition from historical network data. Algorithms such as Random Forest, Logistic Regression, Support Vector Machine (SVM), and XGBoost have been widely applied to cybersecurity tasks due to their ability to generalize across diverse attack types. These models can classify traffic based on features like protocol type, flow duration, packet size, and connection flags. ML algorithms are particularly useful for binary classification (threat vs. no threat) and multiclass classification (type of threat), making them ideal for layered threat analysis in real-time environments.

D. Machine Learning in Cyber Defense

The adoption of machine learning in cyber defense has grown due to its capability to detect anomalies and classify threats without relying on explicit rules. Random Forests, known for their robustness and accuracy, are effective in handling large feature sets and imbalanced data. Logistic Regression provides a baseline approach with high interpretability. SVM models excel in high-dimensional spaces, making them suitable for packet-based traffic analysis. XGBoost, a gradient-boosted decision tree model, has gained popularity for its efficiency and performance in large-scale threat classification tasks. These models, when trained on structured network data, provide accurate threat detection with minimal manual tuning.

E. Review of Related Work

Several studies have explored the application of machine learning to network intrusion detection. Researchers have used benchmark datasets such as NSL-KDD, CICIDS2017, and CyberFedDefender to train and evaluate classification models. Comparative studies show that ensemble methods like Random Forest and XGBoost consistently outperform traditional models in both binary and multiclass scenarios. Prior works have also demonstrated the effectiveness of multi-stage classifiers, where one model detects whether traffic is malicious, and a second model determines the type of attack. This modular approach has shown improved accuracy and scalability, especially in large or heterogeneous network environments.

F. Contribution of Current Work

This project presents a two-stage machine learning system for the classification of cybersecurity threats using structured data from the CyberFedDefender dataset. The first stage performs binary classification to identify threats, while the second stage applies multiclass classification to categorize them into specific attack types. Various ML algorithms including Logistic Regression, Random Forest, SVM, and XGBoost are implemented and compared to determine the most effective approach. The solution is integrated into a Streamlit-based application that enables dataset upload, preprocessing, prediction, and result visualization. The system is designed for ease of use, reproducibility, and real-world applicability, providing a practical tool for cybersecurity research and operations.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

III. EXISTING SYSTEMS

A. Overview of Existing Threat Detection Systems

Traditional threat detection systems have long served as the foundation of network security. These systems include rule-based firewalls, signature-based intrusion detection systems (IDS), and antivirus tools. They operate by scanning incoming traffic or system behavior and triggering alerts when predefined signatures or known attack patterns are matched. Although effective for identifying repeated or well-documented threats, these systems struggle to detect sophisticated or previously unseen attacks such as zero-day exploits.

To address some of these shortcomings, machine learning (ML) techniques have been gradually integrated into modern cybersecurity tools. ML-based systems apply statistical methods to analyze deviations from normal traffic behavior and flag anomalies as potential threats. Some commercial solutions now incorporate behavioral analysis alongside static rules. However, most of these systems still require frequent manual updates, lack dynamic learning capabilities, and often fall short in their ability to classify specific threat types effectively.

B. Functional Scope of Traditional Systems

Traditional and early ML-based cybersecurity systems are generally designed for basic threat monitoring and intrusion detection. Their core components often include:

- Signature matching against known attack patterns
- Static rule sets for packet filtering and port blocking
- Threshold-based alerts for anomalous activity (e.g., excessive data flow)
- Manual feature selection and tuning
- Event logging and basic threat notifications

While useful in controlled environments or as educational tools, these systems tend to provide limited insight beyond binary threat identification. They are primarily focused on detecting known threats and triggering general alerts, without offering further analysis or classification of attack types.

C. Limitations of Existing Lightweight Systems

Despite their foundational role in cybersecurity, both traditional and lightweight ML-based systems have several limitations:

1) Static Detection Logic: These systems cannot adapt to new or evolving attack strategies without manual updates, reducing their effectiveness in dynamic threat environments.
2) High Maintenance Overhead: Constant rule revisions, signature library updates, and manual tuning are required to keep the system functional.
3) Lack of Attack Categorization: Most existing models only detect that a threat exists without identifying what type of attack it represents, which limits actionable response.
4) High False Positive Rates: Legitimate activity may be misclassified as malicious, leading to unnecessary alerts and operational inefficiencies.
5) Limited Learning and Modularity: These systems often lack the ability to improve over time or operate in a modular fashion, making scalability difficult in larger networks.

D. Academic Utility vs Practical Deployment

While traditional systems and basic ML classifiers are valuable for learning and preliminary threat detection, they fall short in real-world scenarios that demand adaptability, automation, and detailed threat intelligence. Their limited classification scope and dependence on static configurations make them less viable in dynamic environments where threats evolve constantly.

The system proposed in this project addresses these gaps by implementing a two-stage machine learning pipeline: one model performs binary classification to determine if traffic is malicious, and the second performs multiclass classification to identify the specific attack type. Combined with a Streamlit-based user interface, the system supports real-time dataset upload, preprocessing, prediction, and result visualization. This design enhances usability, accuracy, and modularity, making it suitable for both academic research and deployment in operational cybersecurity workflows.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

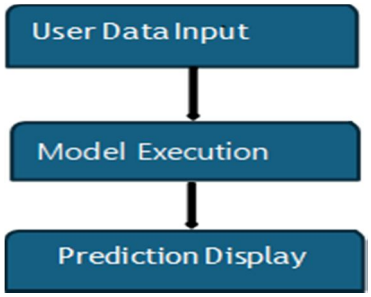
IV. PROPOSED METHODOLOGY

A. Interactive Interface Using Streamlit

The proposed threat classification system begins with a user-friendly interface created using Streamlit, an open-source Python framework for building interactive web applications. This interface is designed to facilitate:

- Uploading CSV files that contain network traffic records.
- Viewing preprocessing status and prediction outcomes.
- Resetting the session to clear previous inputs and results.

The interface is built with accessibility in mind, enabling both cybersecurity professionals and students to easily interact with the system and perform threat classification tasks.



B. Backend Workflow Management

Once the input data is submitted through the interface, the backend initiates a structured workflow consisting of:

- Reading and validating uploaded CSV data.
- Executing preprocessing operations, including:
  - Label encoding of categorical values such as protocols and flags.
  - Feature scaling using a pre-trained scaler.
- Passing the cleaned data to machine learning models for classification.
- Displaying the output as predictions with corresponding class labels.

This workflow ensures the data is consistently formatted and compatible with the trained models for accurate and efficient prediction.

C. Machine Learning Classification Pipeline

The core engine of the system is a two-phase machine learning classification pipeline:

- Phase 1: Binary Threat Detection In this stage, a trained binary classifier (e.g., Random Forest or Logistic Regression) evaluates each record to determine whether it is benign or malicious.
- Phase 2: Attack Type Identification Records identified as threats are forwarded to a second model that classifies the specific type of attack. Potential categories include:

Denial of Service (DoS)
Distributed Denial of Service (DDoS)
Port Scanning
Others

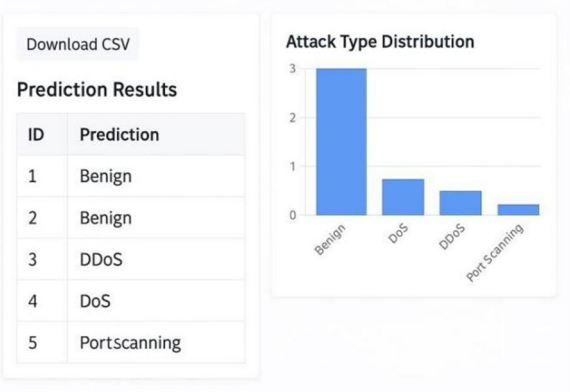
Both models utilize structured input features such as packet statistics, protocol types, flow durations, and byte counts. These inputs are standardized to match the training data format.

D. Prediction Output and Visualization

After classification is complete, the interface provides:

- A results table showing predicted outcomes for each record.
- A download option to export predictions as a CSV file.
- Visual charts (e.g., bar graphs) summarizing the distribution of threat categories.

This comprehensive feedback loop enables users to analyze results efficiently and use them for reporting or further investigation.



V. DESIGN METHODOLOGY

A. *Method Overview*

The proposed system utilizes a structured and modular design to classify cybersecurity threats from network traffic data using multiple machine learning techniques. The architecture supports two stages of classification: one to detect whether the traffic is malicious, and the other to classify the type of attack. The interface is developed using Streamlit, allowing users to upload datasets, trigger preprocessing, execute predictions, and view results seamlessly.

Once the data is uploaded, the system performs:

- Label encoding for categorical variables (e.g., protocol types).
- Feature scaling for numerical attributes (e.g., packet size, byte count).
- Binary classification to detect the presence of malicious traffic.
- Multiclass classification to identify the attack type if a threat is found.

This modular pipeline ensures ease of deployment, reusability of models, and adaptability for real-world scenarios in cybersecurity analytics.

Algorithm Used

Algorithm Name: *Two-Stage Machine Learning-Based Threat Classification using RF, SVM, XGBoost, and Logistic Regression*

The system applies multiple supervised machine learning algorithms for improved accuracy and robustness in threat classification. These include:

- Logistic Regression: Used for its simplicity and interpretability in binary classification.
- Random Forest: An ensemble-based method that handles overfitting and works well with high-dimensional data.
- Support Vector Machine (SVM): Effective in cases where clear margins of separation exist and works well for both binary and multiclass tasks.
- XGBoost: A powerful gradient boosting algorithm known for its accuracy, efficiency, and performance on structured data.

If an instance is classified as malicious, it is passed to a second model which categorizes the specific attack type, such as:

  - Denial of Service (DoS)
  - Distributed Denial of Service (DDoS)
  - Ransomware
  - Other types

Workflow Steps:

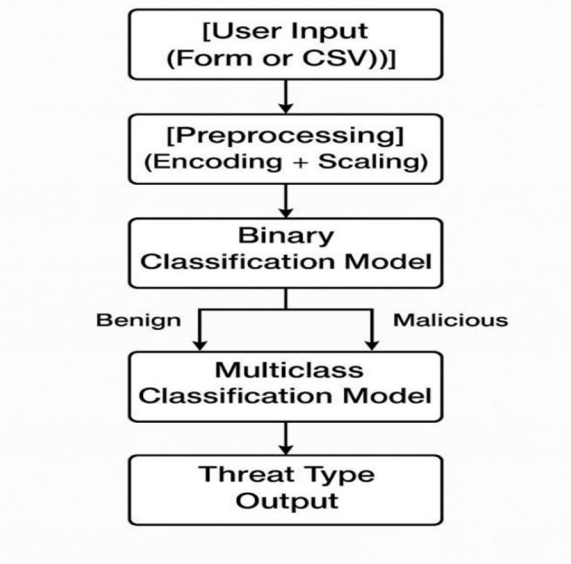
1. Upload structured network traffic data through the interface.
2. Apply preprocessing (label encoding and scaling).
3. Use one of the binary classifiers to detect whether the record is malicious.
4. If malicious, use a multiclass classifier to identify the type of attack.
5. Display results on-screen and provide download options.

B. Pseudocode

```
def classify_network_traffic(record):
    processed_record = preprocess(record)
    is_malicious = binary_model.predict(processed_record)

    if is_malicious == "Benign":
        return "Benign"
    else:
        attack_type = multiclass_model.predict(processed_record)
        return attack_type
```

C. Conceptual Flow Diagram



D. Modularity and Extensibility

This system is designed to allow future upgrades without major architectural changes. Possible extensions include:

1) Replacing the current deep neural network models with more complex ones like CNNs or Transformers for better accuracy.
2) Integrating with real-time packet capture tools to enable live monitoring instead of static file uploads.
3) Adding feedback mechanisms to retrain and improve model performance with new data over time.
4) Deploying the system on embedded hardware for use in environments with limited computing resources.
5) Linking with security tools like firewalls or alerting systems to trigger automated responses when threats are detected.

This flexible design ensures that the system is not only effective today but can evolve as cybersecurity challenges become more complex.

VI. IMPLEMENTATION

This project implements a machine learning-based classification system for detecting and categorizing cyber threats from structured network traffic data. The system is designed to be efficient, user-friendly, and adaptable, combining traditional machine learning algorithms with a lightweight web interface. Python is used as the primary development language due to its strong ecosystem for machine learning and web development.

The system follows a two-stage classification architecture:

1) Binary classification determines whether a given network traffic record is benign or malicious.
2) Multiclass classification further identifies the type of malicious activity (e.g., DoS, DDoS, Port Scan).

An interactive interface built using Streamlit enables users to upload traffic datasets, process input, and view results with clear outputs and visualization support.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

A. Software Environment

- Programming Language: Python 3.8+
- Libraries Used:
  - scikit-learn – for training and applying ML models (Random Forest, SVM, Logistic Regression, XGBoost)
  - pandas, numpy – for data manipulation and numerical operations
  - Streamlit – for building the web interface
  - joblib, pickle – for saving/loading models and encoders
  - matplotlib, seaborn – for data visualization
- Operating System: Compatible with Windows, Linux, and macOS

B. Project Structure

The system is structured into three main components to promote modularity, maintainability, and scalability.

1) Model Inference Logic (Backend)

This component is responsible for executing the classification logic using pre-trained ML models saved in .pkl format. The backend performs the following steps:

- Loads input features along with encoders and a scaler.
- Preprocesses the input by applying label encoding and normalization.
- Uses a binary classifier to check if the record is a threat.
- If malicious, uses a multiclass classifier to predict the specific type of cyberattack.
- Returns the final threat label for each record.

All models are trained offline using structured data and optimized using appropriate hyperparameters.

2) Web Interface (Streamlit App)

The web application serves as the user interface, providing the following features:

- Uploading .csv files with network traffic records
- Performing automated preprocessing (scaling and encoding)
- Predicting and displaying results in a structured table
- Offering CSV download of the output predictions
- Displaying visual summaries using bar charts

The interface ensures ease of use for researchers, students, and cybersecurity professionals.

3) Preprocessing Assets

To maintain consistency between training and prediction:

- A scaler is used for normalizing numerical values (e.g., packet counts, durations)
- Label encoders convert categorical features (e.g., protocol types) into numerical form
- A stored feature list ensures proper feature alignment during prediction

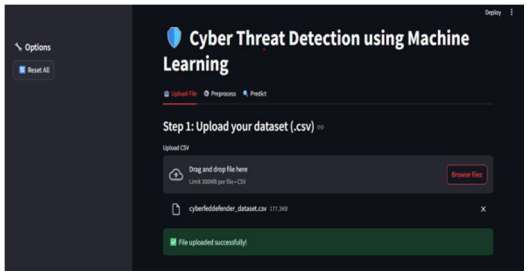
All assets are generated during model training and reused during inference.

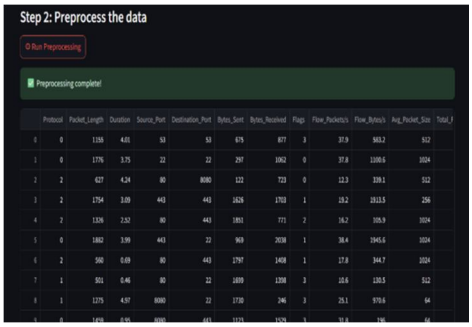
C. Initial Interface View

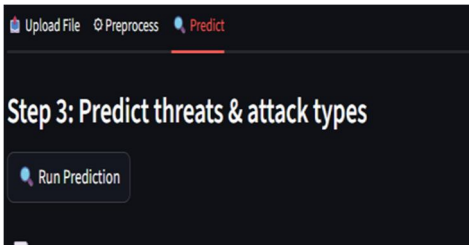
Upon launching the application, users are greeted with a welcome screen containing project instructions. Users can upload a CSV dataset, preprocess it, and begin predictions. A reset option is also provided to start a new session cleanly.

D. File Upload and Preview

Once a dataset is uploaded, the app displays its contents for verification. It checks for feature compatibility, alerts the user in case of missing columns, and prevents errors during classification.





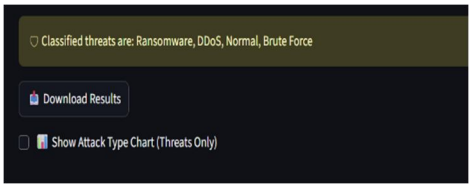


E. Prediction Output and Feedback

After classification:

- A data table displays each record with its predicted class (e.g., Benign, DoS, Port Scanning).
- A download option allows exporting predictions in .csv format.
- A bar chart visually summarizes the types and frequency of detected threats.

This output format ensures transparency and supports quick analysis for cybersecurity decision-making.

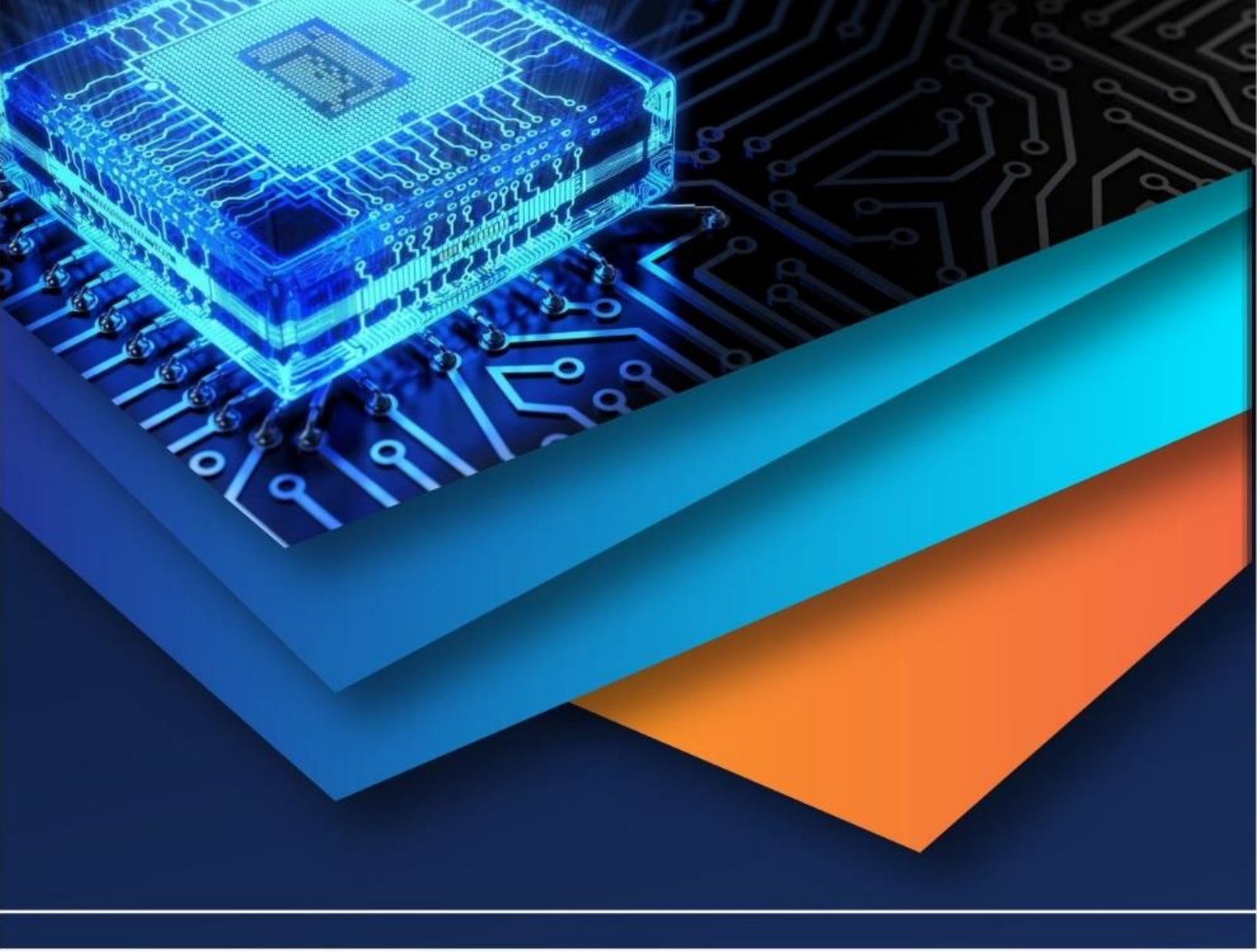


VII. CONCLUSION

This project presents a machine learning-based approach for classifying cyber threats using structured network traffic data. By implementing a two-stage classification process—first identifying whether the traffic is malicious, and then categorizing the type of attack—the system improves the accuracy and reliability of threat detection. Machine learning algorithms such as Random Forest, Support Vector Machine, Logistic Regression, and XGBoost were evaluated for performance and integrated into a Streamlit-based interface for ease of use. The results demonstrate that machine learning models can effectively enhance cybersecurity measures, offering a scalable and accessible solution for detecting a wide range of network threats.

REFERENCES

[1] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009, pp. 1-6.
[2] A. Javaid, Q. Niyaz, W. Sun and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies, 2016, pp. 21–26.
[3] M. R. Ahmed, A. Naser Mahmoud and H. A.Mahmoud,"Machine Learning Approaches for Detecting Cyber Attacks in IoT Systems: A Survey," IEEE Access, vol. 10, pp. 18478–18494, 2022.
[4] S. M. Bridges and R. B. Vaughn, "Fuzzy data mining and genetic algorithms applied to intrusion detection," Proceedings of the National Information Systems Security Conference, 2000, pp. 13–31.
[5] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems," 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 2015, pp. 1–6.
[6] W. Wang, M. Zhu, J. Wang, X. Zeng and Z. Sheng, "Malware Traffic Classification Using Convolutional Neural Network for Representation Learning," 2017 International Conference on Information Networking (ICOIN), 2017, pp. 712–717.
[7] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153–1176, 2016.
[8] H. Hindy et al., "A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems," IEEE Access, vol. 8, pp. 104650–104675, 2020.
[9] S. Shone and Q. N. Ng, "A Deep Learning Approach to Network Intrusion Detection," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41–50, 2018.
[10] N. Doshi, D. Apte, and A. Merchant, "Intrusion Detection Using Machine Learning: A Comparative Study," 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), 2019, pp. 1–6.
[11] T. T. Nguyen and G. Armitage, "A Survey of Techniques for Internet Traffic Classification using Machine Learning," IEEE Communications Surveys & Tutorials, vol. 10, no. 4, pp. 56–76, 2008.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY