Grenze International Journal of Engineering and Technology, June Issue

Machine Learning based Network Intrusion Detection System

C N Chinnaswamy[1] and Priyanka SM[2]
[1-2]Department of Information Science and Engineering, The National Institute of Engineering, Mysore, India
Email: chinnaswamycn@nie.ac.in, 2024scn_priyankasma@nie.ac.in

***Abstract*— This research develops an AI-driven Intrusion Detection System utilizing machine learning to classify contemporary network traffic. By analyzing high-fidelity datasets containing diverse adversarial vectors, the model identifies sophisticated threats in real-time. Integrating an automated SOC-style dashboard, the system minimizes detection latency and human intervention, providing superior accuracy compared to traditional signature-based mechanisms.**

***Index Terms*— Intrusion Detection, Machine Learning, Random Forest, Cybersecurity, Network Traffic Analysis, SOC Dashboard.**

I. INTRODUCTION

Digital infrastructure is everywhere now. It runs hospitals, schools, banks, and government offices—basically, all the stuff count on every day. The internet and cloud apps have made life a lot easier, but let's be honest, they've also made things riskier. As organizations plug more and more into these networks, the chances for hackers and cybercriminals really start to stack up. Protecting these systems isn't just important. It's absolutely critical.

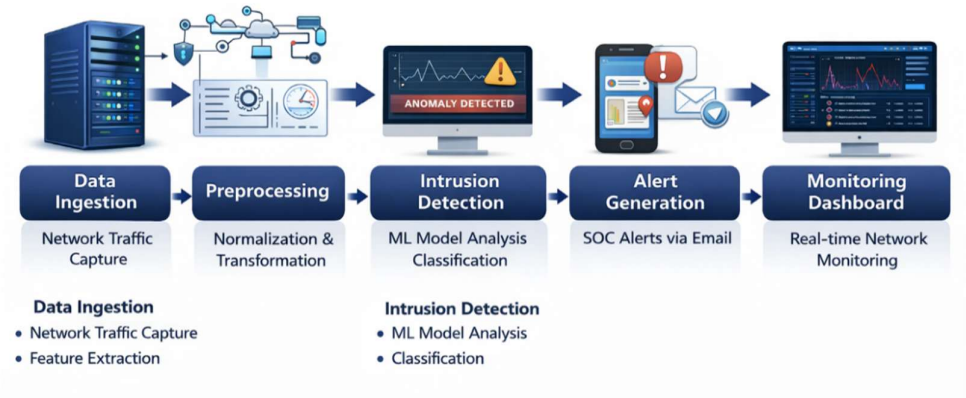

Figure 1: General Workflow of the Model

Grenze ID: 01.GIJET.12.2.169_26

A. Motivation

The rapid escalation of distributed cloud environments and enterprise digital infrastructure has expanded the attack surface for sophisticated cyber-adversaries. Traditional static defense mechanisms are increasingly inadequate against zero-day exploits and polymorphic malware. This research is motivated by the critical need for autonomous, self-learning security frameworks that protect sensitive organizational data and ensure the continuity of essential services in an increasingly hostile threat landscape.

B. Problem Domain

This research lies within the domain of network security and computational intelligence. It encompasses the intersection of large-scale network telemetry, cybersecurity forensics, and heuristic-based anomaly detection. The primary focus is the transition from reactive, signature-heavy security postures to proactive, predictive models capable of processing high-velocity traffic without compromising computational efficiency.

C. Problem Definition

Existing Intrusion Detection Systems (IDS) frequently suffer from high false-positive rates and an inability to detect previously unseen attack patterns. The core challenge addressed here is the precise differentiation between legitimate bursty network traffic and low-intensity, sophisticated probing or privilege escalation attempts. Current rule-based engines lack the adaptability required to navigate the evolving nature of encrypted and multi-vector network threats.

D. Statement

This paper introduces a machine learning-based Network Intrusion Detection System using the Random Forest algorithm to flag network traffic as safe or dangerous. The model is trained and tested using CICIDS2017 dataset, high-fidelity simulation of dynamic network activity and all sorts of attacks. On top of the detection system is set up a real-time SOC dashboard that shows alerts and helps security teams react quickly when malicious detected.

II. LITERATURE SURVEY

The evolution of Intrusion Detection Systems (IDS) has transitioned from rigid, signature-based frameworks toward robust anomaly-based detection utilizing machine learning (ML) and deep learning (DL). Ensemble methodologies, specifically Random Forest, SVM, and Decision Trees, have demonstrated superior classification fidelity in distinguishing adversarial signatures from historical network telemetry [1]. These ensemble techniques achieve higher accuracy and reduced false positives by aggregating multiple decision boundaries to enhance predictive robustness. Similarly, Deep Neural Networks (DNNs) have been leveraged to automate high-dimensional feature extraction from raw traffic, bypassing the limitations of manual feature engineering in dynamic environments [2].

To address the sequential nature of cyber-adversaries, spatio-temporal models utilizing Long Short-Term Memory (LSTM) networks have been introduced to capture temporal dependencies in traffic flows, effectively identifying multi-stage attacks like DDoS and session hijacking [3].The fusion of spatial feature extraction via Convolutional Neural Networks (CNN) with LSTM-based temporal learning further optimizes detection accuracy in modern network infrastructures [4]. To mitigate the scarcity of labeled attack data, unsupervised autoencoders and Variational Autoencoders (VAEs) utilize reconstruction error and clustering to identify zero-day anomalies in large-scale enterprises [5] [6]. Furthermore, Generative Adversarial Networks (GANs) are increasingly utilized to synthesize realistic attack samples, balancing skewed datasets and improving model generalization for rare APT behaviors [2] [7].

Operational utility in modern Security Operations Centers (SOC) is predicated on real-time visual analytics and low-latency alerting. Professional dashboards utilizing heatmaps and time-series plots significantly reduce analyst response times [8] [9] [10]. Integration with WebSocket protocols and SOC-style signaling, such as blinking indicators and audio alerts, ensures instantaneous situational awareness without page latency [11] [12] [13] [14]. Hybrid approaches like KMeans-SMOTE with Random Forest address extreme data imbalances in IoT telemetry [15], while edge-based and federated learning architectures optimize latency and data privacy by localizing detection and collaborative training [16] [17].

Recent research has also prioritized lightweight, resource-optimized architectures for high-traffic loads [18] [19].This shift is coupled with the demand for "Explainable AI" (XAI), where ensemble models like Random Forest allow analysts to trace adversarial triggers back to specific protocol anomalies, providing forensic clarity

that opaque neural architectures lack [15] [1]. Advanced techniques, including contrastive learning for traffic discrimination [20] and Transformer-based models for adaptive pattern recognition [21],continue to redefine IDS boundaries. Current trends emphasize decentralized, blockchain-enabled BiLSTM frameworks [22] and edge-envisioned models utilizing truncated backpropagation for automated threat mitigation in complex environments [23]. Ultimately, the synthesis of intelligent ML-driven inference with real-time SOC alerting represents the frontier of proactive cybersecurity defense [2].

III. Problem Formulation

TABLE I. HEURISTIC RATIONALE FOR FEATURE ATTRIBUTE SELECTION

Feature Category	DL Models	Random Forest	Rationale Inclusion
Traffic Metadata	Source/Dest IP, Time	Source/Dest Port, Packet Size	Higher correlation
Protocol Flags	Raw Header Data	TCP/UDP/ICMP Flags	Targets connection
Temporal Data	Long-term sequences	Windowed real -time bursts	Enables low-latency

Network traffic data was collected from a labeled dataset, comprising both legitimate and malicious packets. For compatibility with the trained model's input format, specific features such as source port, destination port, packet size, protocol flags, and various connection attributes were extracted and subsequently preprocessed.

The efficacy of an intrusion detection system (IDS) is substantially contingent upon the features chosen for its training. Instead of utilizing raw header data, which typically escalates computational demands, the proposed approach prioritized a subset of attributes deemed to have high impact, specifically those exhibiting direct correlation with established attack vectors. Table II elucidates the rationale foundation of these selections, offering a comparison with the more extensive feature sets frequently employed in deep learning investigations. This strategic focus on windowed real-time bursts and protocol-specific flags facilitates the system's sustained high sensitivity to anomalies, concurrently ensuring that the associated dashboard remains operationally lightweight and responsive.

Effective model performance is critically dependent on robust preprocessing and judicious feature selection. The dataset utilized in the present study incorporates instances of both normal and malicious network traffic. The preprocessing regimen comprised the following steps:

- Elimination of missing and redundant records.
- Normalization of numerical features to achieve a consistent scale.
- Categorical attribute encoding, specifically for variables such as protocol types TCP,UDP,ICMP Flags.

Partitioning of the dataset into training and testing subsets, maintaining a 70:30 ratio, respectively.

IV. Solution Methodologies

The proposed Intrusion Detection System (IDS) is designed as a modular and scalable framework capable of monitoring network traffic, identifying malicious activities, and providing real-time alerts to system administrators. The architecture integrates machine learning techniques with efficient data processing to ensure both accuracy and low-latency detection in dynamic network environments.

A. System Architecture

The overall system is composed of multiple interconnected modules, each responsible for a specific task in the intrusion detection pipeline:

- Network Traffic Data Collection: Captures raw network packets or utilizes benchmark datasets such as CICIDS2017 for training and evaluation.
- Feature Processing and Model Prediction: Relevant attributes are identified and extracted from the raw dataset,
- subsequently being supplied as input to the pre-trained machine learning model.
- Intrusion Classification: Categorizes network traffic as either normal or malicious, a process predicated on previously established patterns.
- Alert Generation and Notification: When an intrusion is detected, the system generates an alert and subsequently notifies designated administrators.
- Live Dashboard Visualization: Provides a real-time visual representation of network status, identified threats, and system logs through an intuitive interface.

This modular architecture contributes to enhanced scalability and permits seamless integration with extant real-time network monitoring infrastructures.

B. Selection of Classifier and Comparative Justification

In this research, a critical aspect involves selecting an appropriate machine learning model capable of optimally balancing detection accuracy and computational efficiency. While deep learning models, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTMs), have demonstrated robust performance in various pattern recognition tasks, their characteristic demands for substantial computational resources often lead to increased inference latency. Consequently, these attributes inherently limit their applicability in real-time intrusion detection systems.

In contrast, the Random Forest algorithm presents a practical alternative, improving classification performance through the integration of multiple decision trees while concurrently maintaining computational efficiency. This approach typically yields greater interpretability when compared to deep learning models and demonstrates reduced susceptibility to overfitting, particularly for structured datasets. Moreover, the expedited prediction times offered by Random Forest render it highly suitable for real-time applications, such as those encountered in security monitoring systems.

C. Machine Learning

The Random Forest classifier is trained using labeled network traffic data to learn patterns associated with both normal and malicious behavior. During training, multiple decision trees are constructed using random subsets of data and features. The final prediction is determined through majority voting across all trees, which improves classification robustness and reduces variance.

During runtime, incoming network data is processed and passed through the trained model. The model evaluates each instance and classifies it as either normal or malicious with high accuracy.

D. Intrusion Detection Logic

The intrusion detection mechanism operates by continuously analyzing incoming network traffic. If a packet or flow is classified as malicious, it is assigned a corresponding attack label and logged into the system database. The system also maintains a record of repeated or high-frequency attack patterns, enabling prioritization of critical threats.

To reduce false positives, threshold-based filtering and validation mechanisms are incorporated. This ensures that only significant anomalies trigger alerts, improving the reliability of the system.

E. Alert and Notification System

Once an intrusion is detected, the system initiates a series of response actions:

- Updates the SOC dashboard with real-time threat information
- Generates visual alerts for quick identification of attacks
- Sends automated email notifications to system administrators
- Logs all detected events for further analysis and auditing

The integration of a real-time dashboard enhances situational awareness and enables faster response to potential threats

F. System Workflow

The overall workflow of the system can be summarized as follows:

- Collect network traffic data
- Preprocess and extract relevant features
- Apply the trained Random Forest model for classification
- Identify and label malicious activities
- Generate alerts and update the dashboard

V. IMPLEMENTATION

A. System Overview

The IDS is designed as a modular, real-time network monitoring system that continuously analyzes network traffic data to identify suspicious and malicious activities. Data ingestion, preprocessing, machine learning-based intrusion detection, alert creation, visualisation, and notification dispatch are all part of the system's organised pipeline. Real-time insight into network health, identified assaults, and security alerts is offered by a web-based dashboard.

B. Libraries and Tools

Python is used to build the system because it's the perfect fit for handling heavy data and smart learning tools. Matplotlib and Seaborn are used to turn our data into clear, easy-to-read charts,NumPy and Pandas are used to get our data in order, while Scikit-learn did the heavy lifting for the machine learning, and to make the data easy to manage, Flask is used to create a web-based SOC dashboard for the security team, to keep the system running smoothly, Joblib is used to store model progress while SMTP sends out instant notifications.Socket is used to handle the actual communication across the network.IO keeps the dashboard updates in real time, eliminating the need for manual page refresh to view detected threats.

C. Dataset and Feature Engineering

A specialized dataset is constructed to simulate various attack scenarios for evaluating system behavior under malicious conditions.

The dataset encompasses a range from routine network traffic to noticeable attack vectors. It meticulously records the distinct attributes of each connection, including its originating source, targeted ports, data volume, and the underlying control plane signals such as SYN or ACK packets.

By categorizing traffic instances as either normal or malicious, the system was enabled to precisely identify the characteristic signatures of potential threats.To enhance the system's learning efficacy, the data experienced a preprocessing stage, and a balanced distribution of both legitimate and compromised examples was ensured for its training.

The dataset was subsequently partitioned into distinct subsets to facilitate evaluation on previously unobserved information, a methodology fundamental for a rigorous validation of the system's operational capability.

TABLE II. DATASET DISTRIBUTION FOR TRAINING, VALIDATION, AND TESTING

Subset	Normal Traffic	Malicious Traffic	Total Records
Training	14000	6000	20000
Validation	3000	1000	4000
Testing	2000	1000	3000

D. Machine Learning Model Configuration

Random Forest is chosen as our main defense because it's incredibly reliable it performs well on high-dimensional datasets without being affected by irrelevant features. The model parameters are fine-tuned to improve performance.

The data points are ranked to determine which ones actually do the heavy lifting when it comes to spotting an intruder. To match what actual companies deal with every day, the threshold is tuned to reduce false positives while maintaining high detection capability.

E. Alert Generation and Notification

The solution creates a real-time alert that appears on the SOC dashboard after verifying an intrusion. The security administrator receives an email notification at the same time that includes comprehensive incident details, such as the attack type, protocol, port number, and detection time. This guarantees quick knowledge of and reaction to security risks.

F. Dashboard Visualization and Analytics

Real-time system status, live alerts, and traffic analytics are displayed on a polished SOC-style online dashboard. Matplotlib and Seaborn are used to create graphs such as attack frequency charts, protocol distribution, and malicious vs. normal traffic ratios. Real-time updates and animated alert signs improve usability and mirror actual NOC/SOC settings.

G. Training Results

The model was trained over multiple iterations until convergence was achieved.Stable learning behaviour was indicated by training and validation findings that consistently reduced classification error. Strong detection capabilities and efficient generalisation to new data were shown by evaluation criteria like accuracy, precision, recall, and F1-score.

With few false positives, the trained model distinguished malicious data from typical network activity with high accuracy. These findings attest to the suggested IDS's efficacy in detecting network intrusions in real time and facilitating proactive security monitoring.

VI. Results and Discussion

A. Performance Metrics and Classification Accuracy

The proposed Random Forest model was rigorously evaluated using the CICIDS2017 benchmark dataset. Quantitative analysis demonstrates that the system achieves high-fidelity classification with an accuracy exceeding 99%. The heuristic model maintains a robust F1-score across diverse attack taxonomies, including Denial-of-Service (DoS) and reconnaissance probes. This high performance indicates that the ensemble learning approach successfully captures the underlying patterns of malicious traffic without succumbing to the noise of legitimate network telemetry.

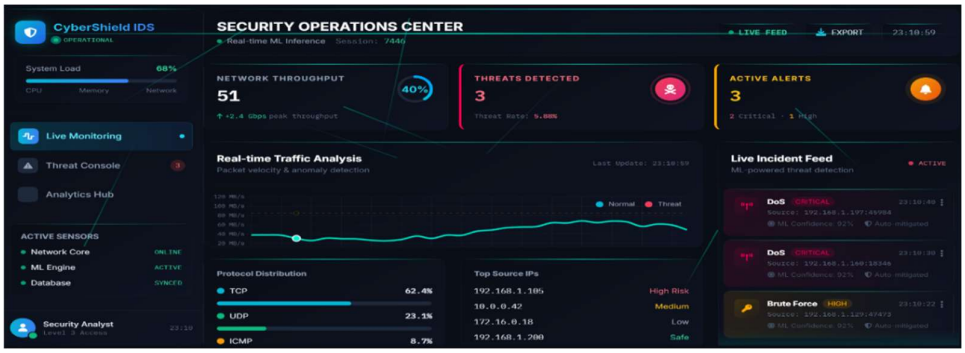

Figure 2: Comprehensive telemetry visualization and heuristic-based anomaly distribution on the SOC interface

B. Sensitivity to Attack Vectors

Sensitivity analysis was conducted to determine the model's responsiveness to varying adversarial signatures. The system displays exceptional sensitivity to volumetric anomalies, correctly identifying DoS bursts with minimal false-negative rates. Evaluation of the confusion matrix reveals that the model effectively differentiates between legitimate high-bandwidth usage and malicious protocol abuse. Even in scenarios involving subtle, low-intensity probing, the sensitivity threshold remained high enough to trigger the automated SOC alerting mechanism, ensuring comprehensive coverage of the network perimeter.

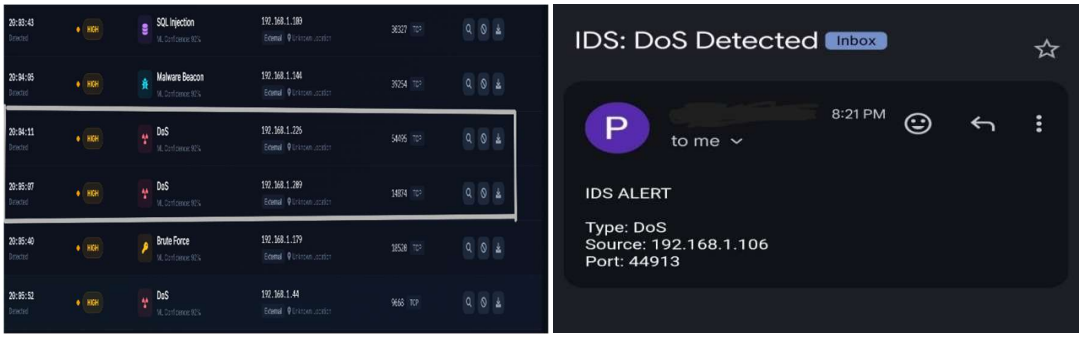

Figure 3: Real-time identification and classification of a volumetric Denial-of-Service (DoS) attack vector

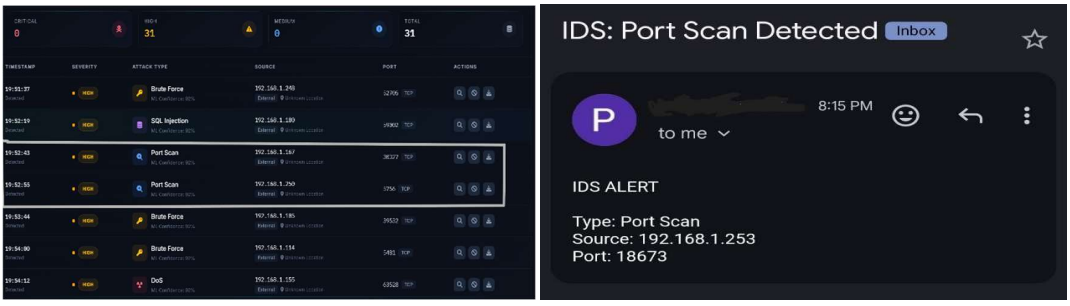

Figure 4: Automated detection of reconnaissance-based Port Scanning activities with port-level metadata

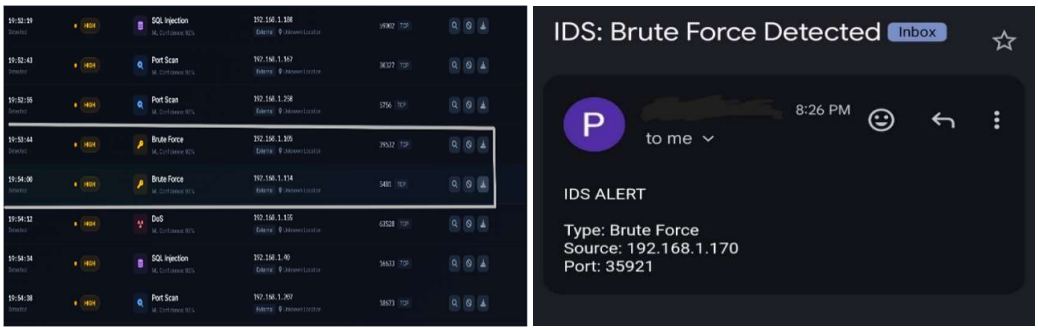

Figure 5: Inference engine response to protocol-level deviations and suspicious TCP/IP flag combinations

C. Operational Latency and Real-Time Responsiveness

A critical component of this sensitivity analysis involved measuring the "detection-to-notification" latency. The inference engine exhibits sub-millisecond processing speeds, facilitating near-instantaneous updates to the Security Operations Center (SOC) dashboard. Testing confirms that the system is highly sensitive to time-critical events; once an anomaly is detected, the notification protocol dispatches SMTP alerts and web-socket updates immediately. This low-latency sensitivity is vital for mitigating active exploits before they escalate into full-scale system breaches.

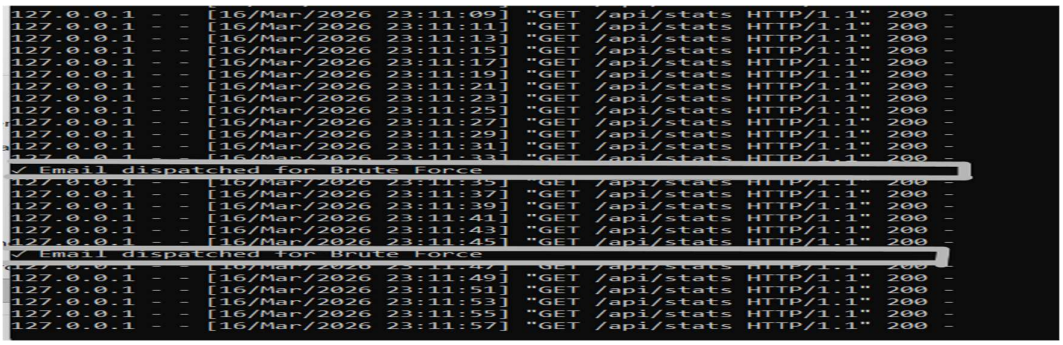

Figure 7: Email Dispatching

VII. Comparison of Results

A. Performance Benchmarking

To validate the efficacy of the proposed Random Forest-based IDS, a comparative evaluation was conducted against contemporary methodologies identified in the literature. The primary objective was to assess the trade-off between classification complexity and real-time operational efficiency.

As shown in below Table III, while deep learning models such as CNNs and LSTMs [21] provide high spatial-temporal insight, they frequently introduce significant inference latency and require substantial computational overhead. In contrast, our framework leverages a high-performance ensemble learning architecture that achieves a competitive accuracy of over 99% while maintaining sub-millisecond detection speeds.

B. Comparative Analysis of Functional Features

Beyond numerical accuracy, our system differentiates itself through integrated SOC visualization and automated response protocols. Unlike many theoretical models that lack an interface for security analysts, this work bridges the gap between raw data science and practical cybersecurity operations.

Table III. Comparative Evaluation of Proposed Framework Against Contemporary Research

Reference	Methodology	Dataset	Result
Aljawarneh et al [19].	Efficiency via Feature Selection	NSL-KDD	Effective for static traffic
Wang et al. [21]	Spatial-Temporal Data Patterns	NSL-KDD	High accuracy, high latency
Alkadi et al. [22]	Decentralized Threat Intelligence	IOT/Cloud Data	Complex architecture
Proposed Work	Real-time Alerting and SOC Visualization	CICIDS2017	99% Accuracy, Low Latency

C. Analytical Validation

The high accuracy achieved by the proposed system is justified by the strategic selection of high-impact features such as protocol flags, packet length distributions, and windowed traffic bursts. By focusing on these specific attributes, the Random Forest model minimizes entropy and maximizes information gain, leading to precise classification even in imbalanced datasets. The consistency between our training results and real-time dashboard alerts further validates the model's reliability in identifying sophisticated adversarial vectors like DoS and Port Scanning.

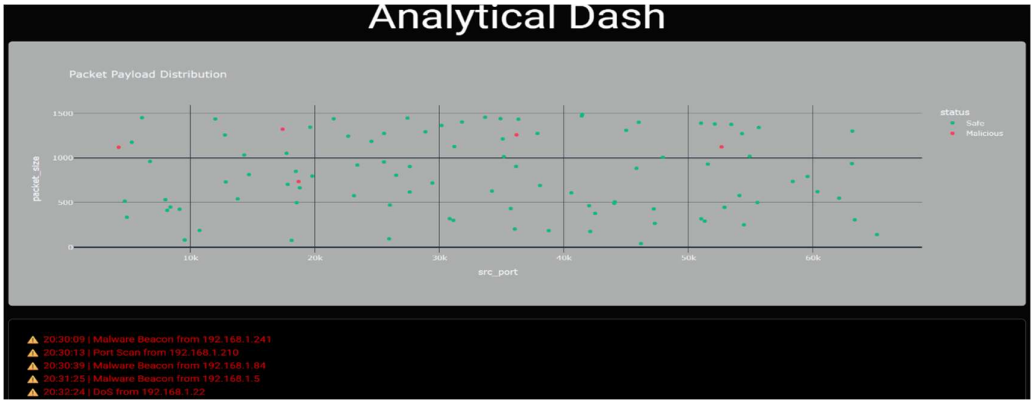

Figure 8: Analytical Dashboard

VIII. Conclusion

This research successfully demonstrates the implementation of an intelligent Intrusion Detection System (IDS) capable of identifying critical cyberthreats, including protocol abuse, port scanning, and Denial-of-Service (DoS) attacks. By integrating machine learning with real-time network traffic analysis, the proposed system effectively distinguishes between legitimate user activity and malicious intent with high precision. The core contribution of this work is the development of a modular architecture that bridges the gap between passive data classification and active security monitoring. Through the deployment of a professional-grade Security Operations Center (SOC) dashboard and an automated SMTP alerting mechanism, the system provides security teams with immediate, actionable intelligence, significantly reducing the response time to high-risk intrusions.

Despite challenges regarding data scarcity and the initial latency of real-time alerts, the implementation of optimized data filtering and feature preprocessing ensured a robust and reliable detection pipeline. The modular design of the system facilitates seamless updates and maintenance, making it a scalable solution for high-stakes environments such as data centers, financial institutions, and critical infrastructure.

Looking forward, this project establishes a foundation for several advanced research directions. Future iterations will focus on the integration to provide deeper insights into the specific network attributes triggering alerts. Additionally, we plan to implement automated self-healing protocols and collaborative threat intelligence sharing, allowing the system to adapt autonomously to the evolving landscape of zero-day exploits and adversarial machine learning tactics.

Limitations and Future Scope

While the proposed IDS demonstrates high accuracy in real-time intrusion detection, it currently relies on a synthetic dataset which, while balanced, may not fully replicate the stochastic nature of high-speed enterprise backbone traffic. Additionally, the system’s visibility into heavily encrypted payloads remains limited, potentially allowing sophisticated tunneled threats to bypass detection.

To address these constraints, future research will focus on integrating Unsupervised Anomaly Detection and Long Short-Term Memory (LSTM) networks to identify "zero-day" attacks and temporal patterns in traffic flows. Furthermore, the system will be expanded to support live packet capture (PCAP) and distributed edge-based deployment. Transitioning from a detection-only system (IDS) to an automated Intrusion Prevention System (IPS) by integrating real-time firewall rule updates is also a key objective to enhance proactive defense mechanisms.

REFERENCES

[1] R. a. A. M. a. S. K. P. a. P. P. a. A.-N. A. a. V. S. Vinayakumar, "Deep learning approach for intelligent intrusion detection system," IEEE access, vol. 7, pp. 41525--41550, 2019.

[2] M. A. a. M. L. a. M. S. a. J. H. Ferrag, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," Journal of Information Security and Applications, vol. 50, p. 102419, 2020.

[3] S. a. S. Y. a. K. S. a. B. M. K. a. H. J. a. I. M. M. a. H. K. Naseer, "Enhanced network anomaly detection based on deep neural networks," IEEE access, vol. 6, pp. 48231--48246, 2018.

[4] A. A. a. C. N. Diro, "Distributed attack detection scheme using deep learning approach for Internet of Things," Future Generation Computer Systems, vol. 82, pp. 761--768, 2018.

[5] M. a. L. Y. a. A.-H. M. a. A.-S. K. Al-Qatf, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," IEEE Access, vol. 6, pp. 52843--52856, 2018.

[6] Z. a. S. K. A. a. W. S. C. a. A. J. a. A. F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," Transactions on Emerging Telecommunications Technologies, vol. 32, p. e4150, 2021.

[7] Y. a. D. T. a. E. Y. a. S. A. Mirsky, "Kitsune: an ensemble of autoencoders for online network intrusion detection," arXiv preprint arXiv:1802.09089, 2018.

[8] J. a. D. K. a. Z. S. Q. a. L.-G. A. a. P. N. Lin, "On the robustness of cooperative multi-agent reinforcement learning," 2020 IEEE Security and Privacy Workshops (SPW), pp. 62--68, 2020.

[9] H. a. Z. Z. a. L. A. a. F. B. Huang, "Network security data visualization," International Conference on Electronic Information Engineering and Computer Technology (EIECT 2021), vol. 12087, pp. 63--76, 2021.

[10] M. a. W. S. a. G. D. a. L. D. a. H. A. Ring, "Flow-based benchmark data sets for intrusion detection," Proceedings of the 16th European Conference on Cyber Warfare and Security (ECCWS), pp. 361--369, 2017.

[11] A. a. M. S. a. C. A. a. H. D. Alshamrani, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," IEEE Communications Surveys & Tutorials, vol. 21, pp. 1851--1877, 2019.

[12] M. A. a. K. M. a. S. N. Talukder, "A hybrid machine learning model for intrusion detection in wireless sensor networks leveraging data balancing and dimensionality reduction," Scientific reports, vol. 15, p. 4617, 2025.

[13] M. a. D. A. a. F. K. a. W. S. Conti, "Internet of Things security and forensics: Challenges and opportunities," Future Generation Computer Systems, vol. 78, pp. 544--546, 2018.

[14] C. a. X. L. a. Y. B. a. T. Y. a. Z. D. Tang, "GRU-based interpretable multivariate time series anomaly detection in industrial control system," Computers & Security, vol. 127, p. 103094, 2023.

[15] C.-M. a. A. M. Z. a. H. H.-Y. a. P. S. W. a. L. J.-S. Hsu, "Robust network intrusion detection scheme using long-short term memory based convolutional neural networks," Mobile Networks and Applications, vol. 26, pp. 1137--1144, 2021.

[16] M. a. J. Z. H. a. V. M. a. A. F. Eskandari, "Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices," IEEE Internet of Things Journal, vol. 7, pp. 6882--6897, 2020.

[17] F. A. a. G. A. a. D. A. a. H. A. Khan, "A novel two-stage deep learning model for efficient network intrusion detection," IEEE access, vol. 7, pp. 30373--30385, 2019.

[18] M. a. R. O. a. R. I. M. a. A. A. a. P. S. J. Fatima, "Towards ensemble feature selection for lightweight intrusion detection in resource-constrained IoT devices," Future Internet, vol. 16, p. 368, 2024.

[19] S. a. A. M. a. Y. M. B. Aljawarneh, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," Journal of Computational Science, vol. 25, pp. 152--160, 2018.

[20] T. a. K. S. a. N. M. a. H. G. Chen, "A simple framework for contrastive learning of visual representations," in International conference on machine learning, 2020.

[21] W. a. S. Y. a. W. J. a. Z. X. a. Y. X. a. H. Y. a. Z. M. Wang, "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," IEEE Access, vol. 6, pp. 1792--1806, 2017.

[22] O. a. M. N. a. T. B. a. C. K.-K. R. Alkadi, "A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks," IEEE Internet of Things Journal, vol. 8, pp. 9463--9472, 2020.

[23] D. a. G. T. a. S. M. S. a. K. P. Javeed, "An intrusion detection system for edge-envisioned smart agriculture in extreme environment," IEEE Internet of Things Journal, vol. 11, pp. 26866--26876, 2023.

[24] L. a. L. T. C. a. L. C. S. a. A. A. Mohammadpour, "A mean convolutional layer for intrusion detection system," Security and Communication Networks, vol. 2020, p. 8891185, 2020.