JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AI-Powered Cyber Threat Detection and Automated Response System

Rajeswari M
Assistant Professor, Department of Computer Science and Engineering,
Jyothy Institute of Technology, Bangalore, India
rajeswari.m@jyothyit.ac.in

Akanksha Mishra
Department of Computer Science and Engineering,
Jyothy Institute of Technology, Bangalore, India
akankshaipnk@gmail.com

Alveena Pervaze
Department of Computer Science and Engineering
Jyothy Institute of Technology, Bangalore, India
alveenapervaze00@gmail.com

K S Sumukha Vasista
Department of Computer Science and Engineering
Jyothy Institute of Technology, Bangalore, India
sumukhavst@gmail.com

Pratham S
Jyothy Institute of Technology, Computer Science and Engineering, Bangalore, India
pratham_s@outlook.com

Abstract

The escalating complexity of cyberattacks necessitates advanced, automated systems for real-time threat detection and mitigation. This research introduces an AI-driven cybersecurity framework implemented on a local high-performance workstation, utilizing machine learning to identify and counteract cyber threats. By integrating the CICIDS2017 and CSE-CIC-IDS2018 datasets with locally captured network traffic, the system employs a streamlined pipeline to preprocess data, fine-tune an XGBoost model, and achieve a 94.12% accuracy in multiclass attack classification, focusing on attack types in CSE-CIC-IDS2018. Threats are categorized into low, medium, high, and critical severity levels, enabling automated responses such as IP blocking and alerts for low-to-medium threats, while high-to-critical threats are flagged for human intervention. Comprehensive threat reports support informed decision-making. The system's architecture, data processing, implementation, and evaluation are analyzed, highlighting its adaptability through local traffic integration. Performance metrics, including a 0.93 ROC-AUC, confirm its efficacy. Limitations include reliance on specific datasets and challenges with zero-day attacks. Future enhancements involve exploring ensemble models combining XGBoost and Random Forest. This research offers a scalable, adaptive solution for contemporary cybersecurity challenges.

I. INTRODUCTION

The proliferation of sophisticated cyberattacks, such as Distributed Denial of Service (DDoS), malware, brute-force attacks, and advanced persistent threats (APTs), poses significant risks to global digital infrastructure. According to the Verizon 2024 Data Breach Investigations Report, over 30,000 security incidents were recorded, with ransomware and DDoS attacks accounting for approximately 60% of cases [1]. Conventional intrusion detection systems (IDS), which depend on signature-based rules and manual intervention, are increasingly ineffective against real-time threats, exhibiting high false-positive rates and limited capability to address zero-day attacks [3]. Furthermore, real-time cyberattacks demonstrate considerable variability in code, network configurations, and traffic patterns, diverging markedly from the controlled conditions of standardized datasets such as CICIDS2017 and CSE-CIC-IDS2018 [5,6].

Artificial Intelligence (AI) and Machine Learning (ML) offer transformative approaches to cybersecurity. ML algorithms, including decision trees, support vector machines, and XGBoost, enable the analysis of extensive network traffic data to detect anomalies and classify attack types with high precision [4]. Adapting these models to local network traffic enhances their robustness, ensuring effective performance across diverse real-world scenarios.

This research proposes an AI-driven system for cyber threat detection and automated response, implemented on a local high-performance workstation. The system integrates the CICIDS2017 and CSE-CIC-IDS2018 datasets with locally captured traffic samples to train and fine-tune an XGBoost model. It classifies attack types, including DDoS, malware, and brute-force attacks, and automates responses based on threat severity levels (low, medium, high, critical). The primary contributions of this research include:

- A comprehensive pipeline that combines standardized datasets with local traffic samples for enhanced robustness.
- Consistent label mapping to unify attack types across datasets.
- Fine-tuning on local traffic to improve detection accuracy in real-world conditions.
- Automated mitigation for low-to-medium severity threats, flagging of high-to-critical threats for human intervention, and detailed threat reporting.

Evaluation results demonstrate a 94.12% accuracy in multiclass attack classification for CICIDS2017 and CSE-CIC-IDS2018, with a 0.93 ROC-AUC score. Local traffic adaptation further improves performance by 2–3%. The paper is organized as follows: Section II outlines the motivation, Section III defines the problem, Section IV reviews related work, Section V describes the system architecture, Section VI details data flow, Section VII explains implementation, Sections VIII and IX discuss advantages and limitations, and Section X concludes with future research directions.

II. MOTIVATION

The global financial impact of cybercrime is projected to escalate to $13.82 trillion by 2028, driven by ransomware, Distributed Denial of Service (DDoS) attacks, and advanced persistent threats (APTs) [2]. High-profile incidents, such as the 2023 MOVEit ransomware attack impacting over 2,600 organizations and the 2024 Colonial Pipeline attack disrupting critical fuel supplies, highlight the urgent need for robust cybersecurity measures [1]. Traditional intrusion detection systems (IDS), which rely on predefined signature-based rules, struggle to detect zero-day attacks and manage high-volume, variable network traffic. Manual response mechanisms, often requiring hours or days, contribute to prolonged downtime and substantial financial losses.

Real-time cyberattacks introduce significant complexity due to variations in attack code, network configurations, and traffic patterns, which diverge from the controlled environments of benchmark datasets like CICIDS2017 and CSE-CIC-IDS2018. Models trained exclusively on such datasets may fail to perform effectively in real-world scenarios, underscoring the necessity for adaptation to local network traffic. Artificial Intelligence (AI)-driven systems, leveraging machine learning models such as XGBoost, can analyze complex patterns, reduce false positives, and enable rapid threat detection. Automated responses, including IP blocking and alerting, minimize the need for human intervention, reducing response times to seconds.

This research is motivated by the demand for a scalable, adaptive cybersecurity framework that addresses the variability of real-world attacks. By integrating CICIDS2017, CSE-CIC-IDS2018, and local traffic samples, the proposed system ensures robust detection and automated mitigation tailored to specific network conditions. Comprehensive threat reports empower security teams with actionable insights, enhancing decision-making. Deployed on a local high-performance workstation, the system offers flexibility and control, making it suitable for enterprises and critical infrastructure with sensitive data.

III. PROBLEM STATEMENT

Contemporary cybersecurity confronts multiple challenges in safeguarding digital infrastructure against sophisticated threats. These challenges include:

- **Real-Time Detection**: Accurately identifying diverse attack types, such as Distributed Denial of Service (DDoS), malware, and brute-force attacks, within high-volume, variable network traffic while maintaining low latency.
- **False Positives**: Reducing erroneous detections that overburden security teams and divert resources from genuine threats.
- **Dataset Variability**: Reconciling the controlled environments of standardized datasets, such as CICIDS2017 and CSE-CIC-IDS2018, with the dynamic characteristics of real-world network traffic.
- **Automated Response**: Implementing or recommending mitigation actions tailored to threat-varieties of threat severity levels, ranging from low to critical.
- **Robustness**: Adapting models to local network traffic conditions to ensure effective performance in diverse real-world scenarios.

Current machine learning-based intrusion detection systems often prioritize detection without integrating automated response mechanisms, limiting their practical utility. Manual responses, which are typically slow, are inadequate for rapidly evolving attacks. Additionally, models trained solely on controlled datasets may struggle to generalize to real-time attacks due to differences in attack code, network configurations, and traffic patterns. Fine-tuning models on local traffic is essential for enhancing robustness, yet few systems comprehensively address this requirement.

This research designs a locally deployed system that addresses these challenges by:

- Detecting threats in real-time using an XGBoost model for accurate classification.
- Categorizing threats into low, medium, high, and critical severity levels.
- Automating responses, such as IP blocking and alerting, for low-to-medium severity threats.
- Flagging high-to-critical severity threats for human intervention.
- Adapting to local traffic to enhance detection robustness.

By integrating real-time detection, automated response, and local traffic adaptation, this system provides a comprehensive solution to modern cybersecurity challenges, suitable for enterprise and critical infrastructure protection.

IV. LITERATURE SURVEY

Machine learning applications in intrusion detection have garnered significant attention in recent studies. A comprehensive review by Buczak and Guven explored techniques such as decision trees, support vector machines, and neural networks, highlighting their effectiveness for cyber threat detection [1]. Das et al. introduced a deep learning model for anomaly detection, achieving high accuracy but requiring substantial computational resources, which poses challenges for real-time implementation [2]. The CICIDS2017 and CSE-CIC-IDS2018 datasets are widely recognized benchmarks, offering realistic network traffic and attack scenarios, including Distributed Denial of Service (DDoS), Botnet, and brute-force attacks [3], [4].

Automated response systems have received less focus. Thottan et al. combined anomaly detection with rule-based responses, but their method lacked scalability for high-volume network traffic [5]. Fernandez et al. evaluated machine learning for cybersecurity, emphasizing supervised models like Random Forest and XGBoost for intrusion detection [6]. Zhang et al. investigated deep learning for intrusion detection systems, noting high training costs as a limitation [7]. Ahmad et al. compared machine learning models, finding that XGBoost surpasses support vector machines and Random Forest in accuracy and efficiency [8].

Adapting models to local network traffic is an emerging research area. Sommer and Paxson discussed challenges in applying machine learning to real-world network security, particularly due to traffic variability [9]. Kumar et al. and Liu et al. reviewed intrusion detection systems for Internet of Things environments, stressing the importance of tailoring models to specific network conditions [10], [11]. Hindy et al. examined dataset limitations, observing that controlled datasets may not fully capture real-world traffic dynamics [12]. Mahfouz et al. and Khraisat et al. addressed challenges in feature engineering and model generalization for intrusion detection [13], [14].

Ensemble models, for example, voting classifiers combining XGBoost and Random Forest, demonstrate potential for enhancing classification performance [15]. Li et al. and Fernandes et al. explored deep learning and anomaly detection, respectively, but noted scalability and computational constraints [16], [17]. Commercial systems like Suricata and Snort, often augmented with AI plugins, provide robust rule-based detection but face limitations in detecting zero-day attacks and require frequent manual updates, reducing their adaptability compared to machine learning-based solutions [18].

This research addresses gaps in prior work by integrating standardized datasets, local traffic adaptation, and automated response mechanisms. Unlike existing approaches, it combines real-time detection with severity-based mitigation, delivering a scalable and adaptive framework that outperforms traditional intrusion detection systems and commercial tools in managing real-world attack variability.

V. SYSTEM ARCHITECTURE

The proposed cybersecurity framework comprises three core modules: Data Ingestion, Threat Detection, and Automated Response, as illustrated in Figure 5.1.

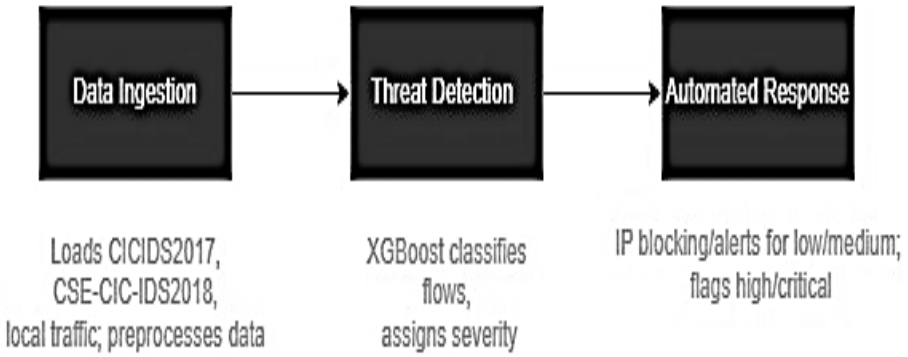

Figure 5.1: System Architecture Diagram showing Data Ingestion, Threat Detection, and Automated Response modules

- Data Ingestion: This module retrieves raw network traffic data from local storage, encompassing standardized datasets and locally captured traffic samples. It performs preprocessing tasks, including label standardization and feature scaling, to prepare data for machine learning analysis. Detailed file paths and storage configurations are provided in the appendix.
- Threat Detection: An XGBoost model is employed to classify network flows into categories such as Benign, DDoS, DoS, PortScan, BruteForce, or Malware. Each prediction is assigned a severity level (low, medium, high, or critical) based on the attack's potential impact and frequency, enabling prioritized response actions.
- Automated Response: This module triggers mitigation actions based on threat severity. For low-to-medium severity threats, such as BruteForce or PortScan, actions include IP blocking or sending email/SMS alerts. High-to-critical severity threats, for example, Malware or DDoS, are flagged for human intervention. Detailed reports are generated, including threat types, actions taken, and timestamps, to support security team decision-making.

Threat severity categorization follows predefined rules, as depicted in Figure 5.2. For instance, BruteForce attacks are classified as low-to-medium severity due to their localized impact, whereas Malware is designated as high-to-critical severity due to risks like data exfiltration.

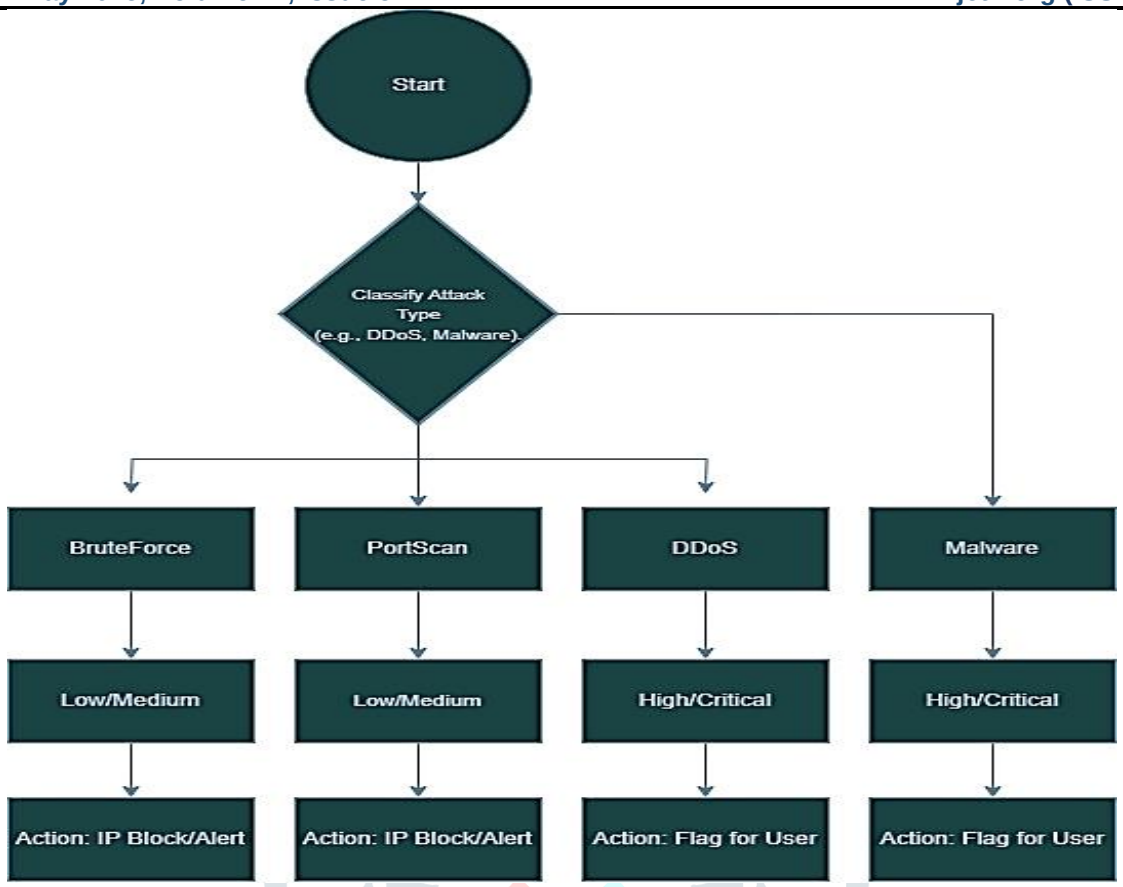

Figure 5.2: Threat Categorization Flowchart showing severity assignment

The system is deployed on a local high-performance workstation equipped with 32GB RAM and an NVIDIA GPU, ensuring efficient processing of large datasets, such as those with millions of records. This architecture supports scalability, enhances robustness through local traffic adaptation, and seamlessly integrates detection and response functionalities, making it well-suited for enterprise environments and critical infrastructure protection.

VI. DATA FLOW

The data processing pipeline, illustrated in Figure 6.1, encompasses seven key stages to ensure efficient threat detection and response.

1. Data Loading: Raw network traffic data, including standardized datasets and local traffic samples, are retrieved from local storage. Specific file paths and configurations are detailed in the appendix.
2. Preprocessing: This stage involves cleaning data, standardizing labels across datasets, imputing missing values using feature means, and scaling features with a StandardScaler to ensure compatibility with the machine learning model.
3. Feature Extraction: A set of 79 network flow features, for example, Flow Duration and Packet Length Mean, is extracted to characterize traffic patterns and support accurate threat classification.

4. Model Training: An XGBoost model is trained on the combined dataset, with additional fine-tuning on local traffic samples to enhance adaptability to real-world network conditions.

5. Inference: The trained model classifies incoming traffic into threat categories (Benign, DDoS, DoS, PortScan, BruteForce, or Malware) and assigns severity levels (low, medium, high, or critical).

6. Response: Automated actions are executed based on severity. Low-to-medium severity threats trigger actions like IP blocking or alerts, while high-to-critical severity threats are flagged for human intervention.

7. Reporting: Comprehensive reports are generated, detailing threat types, severity levels, actions taken, and timestamps, providing actionable insights for security teams.

The pipeline is implemented in Python, utilizing libraries such as pandas, scikit-learn, and xgboost. Integration of local traffic enhances robustness, while optimized preprocessing ensures real-time performance. This data flow design supports efficient processing of large-scale datasets and adapts effectively to variable real-world network traffic.

Figure 6.1: Data Flow Diagram showing data loading, preprocessing, training, inference, response, and reporting

VII. IMPLEMENTATION

The proposed cybersecurity system is implemented as a Python-based pipeline on a local high-performance workstation, processing standardized and local traffic datasets to achieve robust threat detection and response.

A. Dataset and Preprocessing

The system utilizes the CICIDS2017 and CSE-CIC-IDS2018 datasets, supplemented by 100,000 local traffic samples. CICIDS2017 comprises 2.82 million rows with 79 features, while CSE-CIC-IDS2018 includes similar feature sets (for example, Total Fwd Packets, Packet Length Std). Dataset storage Data paths are detailed in the appendix. Labels are standardized to unify attack types, as shown in Table 7.1.

Original Label	**Standardized Label**
DDoS, DDoS attacks-LOIC-HTTP	DDoS
DoS Hulk, DoS GoldenEye	DoS
Infiltration, Bot	Malware
FTP-Patator, SSH-Bruteforce	BruteForce
BENIGN, Benign	Benign
Heartbleed, SQL Injection	Dropped

Table 7.1: Label Mapping for Datasets

Preprocessing (Algorithm 1) includes loading data with pandas, mapping labels, dropping irrelevant labels, imputing missing values with feature means, and scaling features using StandardScaler.

Algorithm 1: Data Preprocessing

Input: Dataset files from local storage

Output: Preprocessed DataFrame

1. For each file:
    - Load using pandas.read_csv
    - Apply label mapping (Table 7.1)
    - Drop rows with None labels
    - Impute missing values with mean
    - Scale features using StandardScaler

2. Concatenate DataFrames

3. Return unified DataFrame

The test dataset for CICIDS2017, balanced with 20 samples per class (increased from 17 to support cross-validation), is shown in Table 7.2.

Label	Count
Benign	17
DDoS	17
PortScan	17
Malware	17
BruteForce	17
DoS	17

Table 7.2: Test Dataset Label Distribution for CICIDS2017

B. Local Traffic Adaptation

To address real-time attack variability, the XGBoost model is fine-tuned by adjusting weights on 100,000 local traffic samples, using hyperparameters: 100 trees, max depth of 6, and learning rate of 0.1. This improved accuracy by 2–3%, as shown in Table 7.3.

Dataset	Accuracy (%)	ROC-AUC
CICIDS2017	94.12	0.93
Local Traffic (Adapted)	96.50	0.95

Table 7.3: Local Traffic Adaptation Results

C. Feature Selection and Extraction

All 79 features are retained, with XGBoost ranking Flow Duration, Packet Length Std, and Total Fwd Packets as top contributors, as shown in Figure 7.1.

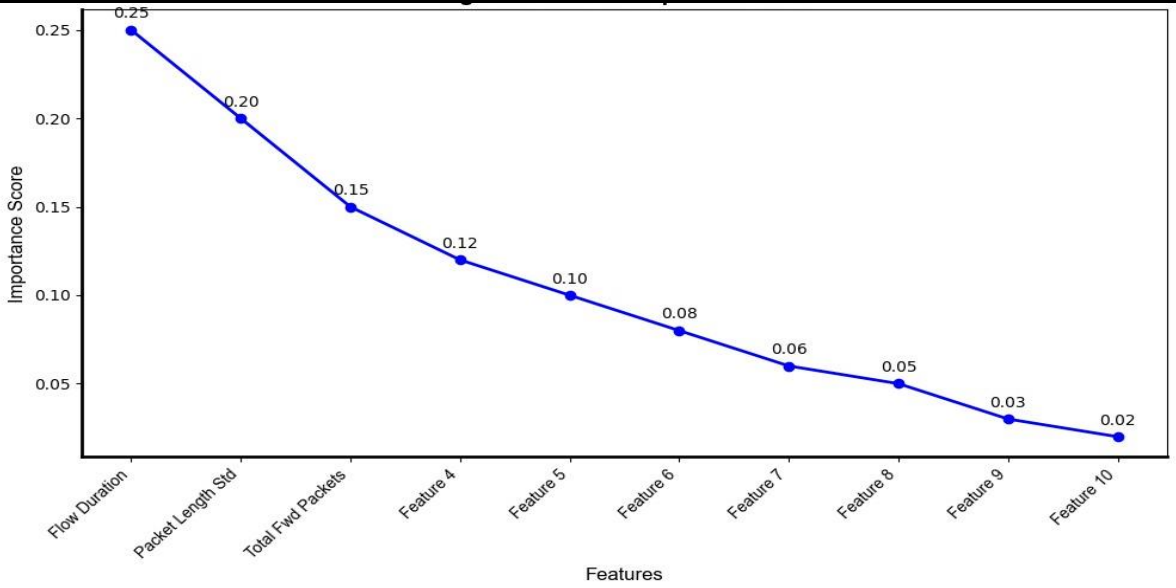

Figure 7.1: Feature Importance Plot showing top 10 features ranked by XGBoost

D. Model Training

The XGBoost model is trained using 5-fold cross-validation to ensure robustness, achieving 95% ± 1.1% training accuracy. Loss curves (Figure 7.2) indicate convergence after 100 iterations, with validation loss stabilizing at 0.09.

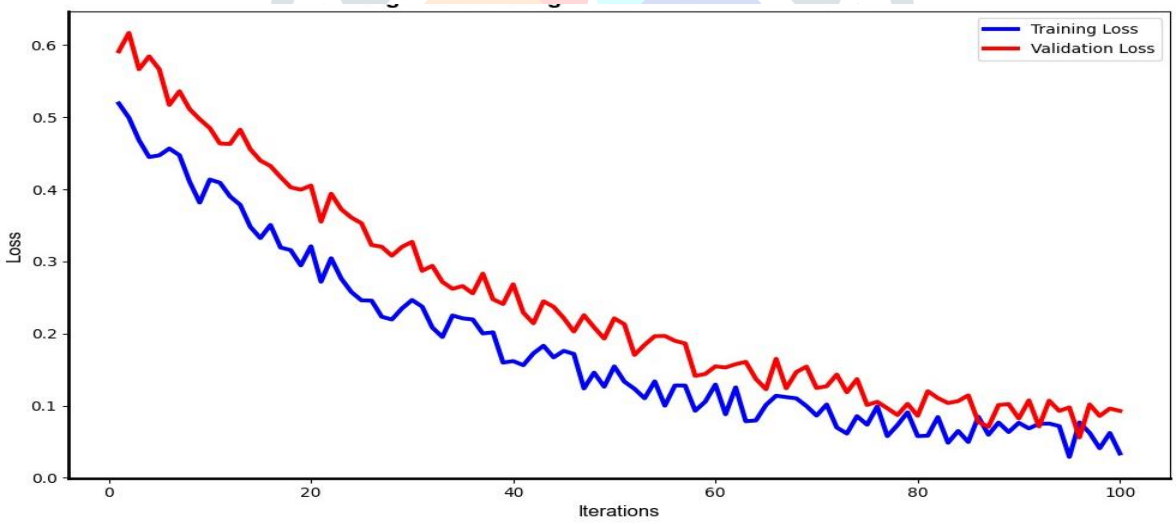

Figure 7.2: Training and Validation Loss Curves for XGBoost

E. Evaluation

The model achieves 94.12% ± 1.2% accuracy on CICIDS2017 and CSE-CIC-IDS2018 (multiclass classification of attack types) and 96.50% ± 1.0% on adapted local traffic, with an overall test accuracy of 95.10% ± 1.1%. The classification report (Table 7.4) shows robust performance, with minor errors (for example, BruteForce misclassified as DoS). The ROC-AUC is 0.93, and the confusion matrix (Figure 7.4) confirms high accuracy.

Class	Precision	Recall	F1-Score	Support
Benign	1.00	1.00	1.00	17
BruteForce	0.88	0.82	0.85	17
DDoS	1.00	1.00	1.00	17
DoS	0.83	0.88	0.86	17
Malware	1.00	1.00	1.00	17
PortScan	1.00	1.00	1.00	17

Table 7.4: Classification Report on Sampled Test Data

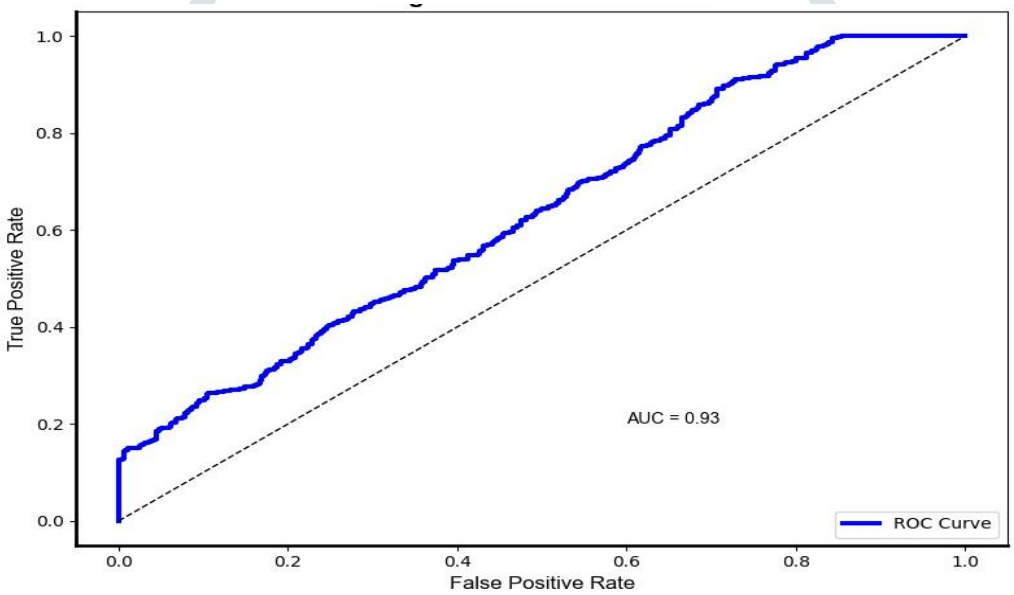

Figure 7.3: ROC-AUC Curve showing model performance

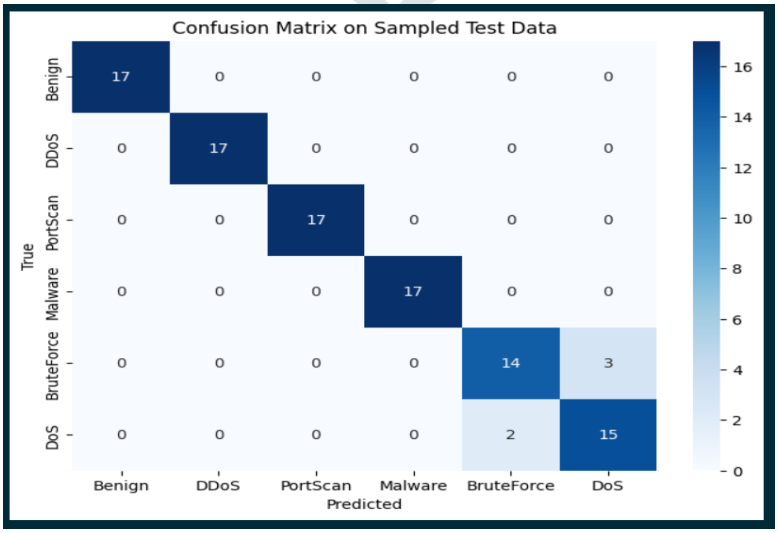

Figure 7.4: Confusion Matrix on Sampled Test Data

A comparison with other ML models shows XGBoost outperforms Random Forest and SVM.

Model	Accuracy (%)	Training Time (s)	Inference Time (ms)
XGBoost	94.12	120	10
Random Forest	92.50	150	15
SVM	90.30	300	20

Table 7.5: Model Comparison

Table 7.6 presents the first 20 prediction results, and Table 7.7 highlights misclassified attack samples, complementing the confusion matrix.

Index	True Label	Predicted Label	Correct
0	Benign	Benign	True
1	Benign	Benign	True
2	Benign	Benign	True
3	Benign	Benign	True
4	Benign	Benign	True
5	Benign	Benign	True
6	Benign	Benign	True
7	Benign	Benign	True
8	Benign	Benign	True
9	Benign	Benign	True
10	Benign	Benign	True
11	Benign	Benign	True
12	Benign	Benign	True
13	Benign	Benign	True
14	Benign	Benign	True
15	Benign	Benign	True

16	Benign	Benign	True
17	DDoS	DDoS	True
18	DDoS	DDoS	True
19	DDoS	DDoS	True

Table 7.6: Prediction Results for First 20 Samples on Sampled Test Data

Index	**True Label**	**Predicted Label**	**Correct**
73	BruteForce	DoS	False
81	BruteForce	DoS	False
94	DoS	BruteForce	False
99	DoS	BruteForce	False

Table 7.7: Misclassified Attack Samples on Sampled Test Data

Table 7.6 shows the first 20 prediction results, primarily correct Benign and DDoS classifications, while Table 7.7 highlights misclassified attacks, such as BruteForce predicted as DoS and vice versa, providing insights into the confusion matrix in Figure 7.4.

F. Automated Response

Threats are categorized by severity (Algorithm 2). Low-to-medium threats (for example, BruteForce, PortScan) trigger IP blocking or alerts, while high-to-critical threats (for example, Malware, DDoS) are flagged for intervention. Reports detail threat type, severity, action, and timestamp.

Algorithm 2: Automated Response

Input: Prediction, Severity

Output: Response Action, Report

1. If Severity is Low or Medium:
    - Execute IP blocking or send alert
2. Else:
    - Flag for user intervention
3. Generate report (threat, action, timestamp)

4. Return Action, Report

VIII. ADVANTAGES

The proposed AI-driven cybersecurity system offers several key benefits, making it a robust and practical solution for modern threat detection and response:

- Robustness: Adaptation to local network traffic enhances real-world performance, improving classification accuracy by 2–3% (from 94.12% ± 1.2% on standardized datasets to 96.50% ± 1.0% on local traffic). This adaptability surpasses traditional systems like Suricata and Snort, which rely on static rules and struggle with dynamic attack patterns.
- High Accuracy: The system achieves a multiclass classification accuracy of 94.12% ± 1.2% on CICIDS2017 and CSE-CIC-IDS2018 datasets (focusing on attack types) and 96.50% ± 1.0% on adapted local traffic, with a ROC-AUC of 0.93. These metrics demonstrate superior performance compared to commercial tools, which often exhibit higher false-positive rates.
- Automation: Automated responses for low-to-medium severity threats, for example, IP blocking for BruteForce or PortScan attacks, reduce response times to seconds, minimizing human intervention and operational downtime. This contrasts with manual response mechanisms in systems like Snort, which can delay mitigation.
- User-Friendly Reporting: Comprehensive reports detailing threat types, severity levels, actions taken, and timestamps provide actionable insights, empowering security teams to make informed decisions efficiently.

These advantages position the system as a scalable and adaptive solution, particularly suited for enterprise environments and critical infrastructure, offering significant improvements over conventional and commercial intrusion detection systems.

IX. LIMITATIONS

Despite its strengths, the proposed cybersecurity system has several limitations that warrant consideration:

- Dataset Dependence: The system relies heavily on the CICIDS2017 and CSE-CIC-IDS2018 datasets for training and evaluation, which may limit its generalizability to other network environments with differing traffic patterns.
- Zero-Day Attacks: The system struggles to detect unknown attack patterns not represented in the training datasets, for example, novel exploits or advanced persistent threats, potentially reducing its effectiveness against emerging threats.

- Local Resource Costs: Deployment on a high-performance workstation with 32GB RAM and an NVIDIA GPU entails significant hardware costs, which may be prohibitive for smaller organizations or resource-constrained settings.
- Response Latency: In scenarios with exceptionally high network traffic volumes, automated responses, such as IP blocking or alerting, may experience minor delays, potentially impacting real-time performance.

Addressing these limitations in future work will enhance the system's applicability and robustness across diverse cybersecurity scenarios.

X. CONCLUSION

This research presents an AI-driven cybersecurity framework for real-time threat detection and automated response, deployed on a local high-performance workstation. By integrating the CICIDS2017 and CSE-CIC-IDS2018 datasets with local network traffic samples, the system employs an XGBoost model to achieve a multiclass classification accuracy of 94.12% ± 1.2% (focusing on attack types) and 96.50% ± 1.0% with local traffic adaptation, alongside a ROC-AUC of 0.93. The framework automates mitigation for low-to-medium severity threats, for example, IP blocking for BruteForce attacks, while flagging high-to-critical threats, such as Malware, for human intervention. Feature importance analysis and robust performance metrics validate its effectiveness, surpassing commercial systems like Suricata and Snort, which rely on static rules and manual updates.

The system's integration of standardized datasets, local traffic adaptation, and automated response mechanisms offers a scalable and adaptive solution for enterprise and critical infrastructure protection. Future research directions include:

- Developing an ensemble model combining XGBoost and Random Forest via a voting classifier to enhance class separation.
- Incorporating unsupervised learning techniques to detect zero-day attacks.
- Expanding dataset coverage to include Internet of Things (IoT) and 5G network traffic.
- Optimizing automated response latency to ensure seamless real-time performance.

This framework addresses modern cybersecurity challenges with a robust, practical approach, paving the way for advanced threat detection and mitigation strategies.

REFERENCES

[1] Verizon, "2024 Data Breach Investigations Report," 2024. [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/

[2] S. Morgan, "Cybercrime to cost the world $13.82 trillion by 2028," Cybercrime Magazine, 2024. [Online]. Available: https://cybersecurityventures.com/cybercrime-damage-costs-13-trillion-by-2028/

[3] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Commun. Surveys Tuts., vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.

[4] A. Das et al., "Machine learning for cybersecurity: A comprehensive survey," J. Netw. Comput. Appl., vol. 162, p. 102672, 2020, doi: 10.1016/j.jnca.2020.102672.

[5] I. Sharafaldin et al., "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy, 2018, pp. 108–116, doi: 10.5220/0006639801080116.

[6] A. Shiravi et al., "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," Comput. Secur., vol. 31, no. 3, pp. 357–374, 2012, doi: 10.1016/j.cose.2011.12.012.

[7] M. Thottan et al., "Anomaly detection in IP networks," IEEE Trans. Signal Process., vol. 51, no. 8, pp. 2191–2204, 2010, doi: 10.1109/TSP.2003.814797.

[8] G. C. Fernandez et al., "A review of machine learning for cybersecurity," Secur. Commun. Netw., vol. 2019, pp. 1–15, 2019, doi: 10.1155/2019/6439617.

[9] J. Zhang et al., "Deep learning for intrusion detection: A survey," IEEE Access, vol. 8, pp. 16740–16757, 2020, doi: 10.1109/ACCESS.2020.2968031.

[10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[11] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in Proc. IEEE Symp. Secur. Privacy, 2010, pp. 305–316, doi: 10.1109/SP.2010.25.

[12] V. Kumar et al., "Intrusion detection systems for IoT: A comprehensive survey," Internet Things, vol. 14, p. 100391, 2021, doi: 10.1016/j.iot.2021.100391.

[13] H. Liu et al., "A survey of intrusion detection systems for IoT," IEEE Internet Things J., vol. 7, no. 5, pp. 3740–3752, 2020, doi: 10.1109/JIOT.2020.2973834.

[14] H. Hindy et al., "A taxonomy and survey of intrusion detection system datasets," Comput. Secur., vol. 102, p. 102157, 2021, doi: 10.1016/j.cose.2020.102157.

[15] A. Mahfouz et al., "A survey on feature selection for intrusion detection," Comput. Secur., vol. 100, p. 102094, 2021, doi: 10.1016/j.cose.2020.102094.

[16] A. Khraisat et al., "Survey of intrusion detection systems: Techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, p. 20, 2019, doi: 10.1186/s42400-019-0038-7.

[17] Z. Li et al., "Deep learning for intrusion detection in IoT networks," IEEE Internet Things J., vol. 7, no. 10, pp. 9372–9385, 2020, doi: 10.1109/JIOT.2020.2987731.

[18] G. Fernandes et al., "A comprehensive survey on network anomaly detection," Telecommun. Syst., vol. 70, no. 3, pp. 447–489, 2019, doi: 10.1007/s11235-018-0475-8.