DEPAETMENT OF CYBER SECURITY
Dawood University OFEngineering And Technology

DATA STRUCTURE
AI-Powered SOC Platform

A Hybrid Framework Integrating ML Detection, LLM Explanations, and Automated Response

GROUP MEMBERS

Muhammad Al i: 24F-CY-031
Syed Shaheer : 24F-CY-036
Rasab Khan : 24F-CY-041
Syed Shamveel Ahmed : 24F-CY-026
Raj Kumar : 24F-CY-045

ABSTRACT

The rapid growth of digital infrastructures, cloud platforms, and interconnected devices has greatly increased the risk of cyberattacks. Modern networks generate massive streams of logs and system events, making manual monitoring difficult and causing traditional rule-based security systems to fall behind. To address these challenges, this project introduces an AI-Driven Real-Time Threat Detection and Automated Incident Response System designed to intelligently analyze live system data, classify threat severity, and take appropriate defensive actions. The core goal is to combine machine-learning-based anomaly detection with Large Language Model (LLM)–assisted reasoning to create an adaptive, scalable, and more reliable cybersecurity framework.

In the preparation stage, the system is trained on large collections of historical cybersecurity logs obtained from multiple sources. These logs may include network activity, login behaviors, system calls, firewall events, and application activity. The data undergoes thorough preprocessing to remove noise, extract relevant features, encode categorical fields, normalize values, and handle class imbalance. Through this process, the machine learning model learns to differentiate normal user behavior from malicious patterns such as brute-force attempts, unusual authentication paths, privilege escalation, suspicious file access, or irregular data movement. Since training data can be extremely large, the system separates training from real-time deployment: once the model is trained, only fresh live data is processed during actual use, making threat classification fast, lightweight, and continuous.

During real-time operation, the system receives live telemetry from network devices, servers, endpoints, and applications. This data is passed through the trained ML model, which assigns a threat level—Low, Medium, or High—based on learned behaviors. Each severity level triggers a predefined action. Low-level alerts reflect minor irregularities; these are logged and sent to the LLM for explanation, context, and recommended next steps. Medium-level alerts indicate stronger signs of malicious behavior and notify the security team, supported by LLM-generated summaries, potential attacker intent, and mitigation strategies. High-level alerts represent confirmed malicious activity. For such cases, the system performs automated defensive actions, such as blocking suspicious IP addresses,

disabling accounts, isolating affected devices, or terminating dangerous processes—followed by detailed natural-language documentation generated by the LLM.

The integration of an LLM significantly enhances explainability. Traditional ML models are often accurate but difficult to interpret. The LLM solves this by reviewing model outputs and producing clear, human-readable insights. It can explain why an alert was triggered, describe the nature of the threat, suggest next steps, and help analysts understand complex system behaviors. This improves response quality and supports junior analysts who may not have deep experience with advanced attack techniques.

Together, the ML model and LLM form a hybrid system that is both fast and intelligent. The ML component continuously monitors system behavior and adapts to new or changing attack patterns, while the LLM supports human decision-making by providing analysis, summaries, and recommendations. This combination reduces false positives, speeds up detection, and enhances automated response. The system is designed with modularity, allowing it to integrate with common security tools such as SIEMs, firewalls, and cloud security solutions.

This project delivers several important benefits for modern cybersecurity operations. It lowers the workload on security teams by automating the initial analysis of alerts and handling high-severity threats without manual involvement. It improves detection accuracy by learning from large-scale historical data and comparing it with real-time system behavior. It enhances situational awareness through natural-language insights, enabling faster and clearer decision-making. It is also scalable and adaptable, making it suitable for different organizational environments—from small offices to large cloud infrastructures.

The project also addresses the limitations of conventional security systems. Signature-based tools struggle with zero-day attacks, whereas anomaly detection models can detect unknown threats by identifying deviations from normal behavior. Security teams often face alert fatigue; automated triage reduces noise by classifying alerts by severity. By adding LLM reasoning, the system overcomes another challenge: the inability of ML models to clearly justify their decisions. Instead of raw outputs, analysts receive meaningful explanations that build trust and support informed action.

Safety and reliability are also key design considerations. Automated actions are restricted to high-confidence threats to prevent accidental disruptions. Each automated decision is logged, and the LLM is used to reassess critical events to confirm that mitigation actions align with the detected threat. Continuous monitoring and scheduled retraining ensure that the model remains effective as new attack patterns emerge.

In summary, this project presents a modern, intelligent, and adaptable cybersecurity solution that combines machine learning and language-based reasoning to deliver real-time threat detection, automated incident response, and clear, human-friendly insights. It is designed to keep pace with evolving threats while reducing analyst workload, improving accuracy, and enhancing the overall security posture of organizations.

Index

CHAPTER 1: INTRODUCTION

Modern networks and digital infrastructures generate an enormous volume of log data, with millions of events recorded every day. Expecting security analysts to manually inspect and interpret these logs is unrealistic, which is why a wide range of automated security solutions have been developed. However, traditional rule-based systems often struggle to keep up with evolving attack techniques, leading to missed alerts or excessive false positives. As cyber threats grow increasingly sophisticated, there is a clear need for intelligent, adaptive, and scalable log-analysis systems capable of identifying anomalies in real time.

Cybersecurity has become a critical concern for organizations of all sizes, as attacks such as ransomware, phishing, zero-day exploits, and insider threats continue to evolve. Conventional systems, including signature-based Intrusion Detection Systems (IDS) and Security Information and Event Management (SIEM) platforms, often fall short in detecting new or unknown threats. Analysts face alert fatigue due to high false-positive rates, and manual monitoring cannot keep pace with the rapid generation of network logs. Consequently, organizations require automated, intelligent, and interpretable solutions that can analyze large datasets, detect threats in real time, and assist analysts in responding effectively.

Artificial intelligence (AI) and machine learning (ML) have emerged as promising technologies to address these challenges. ML models can analyze historical and real-time data, recognize patterns, and classify anomalous behavior that may indicate cyberattacks. Meanwhile, Large Language Models (LLMs) provide the ability to generate human-readable explanations, summarize incidents, and recommend mitigation strategies, bridging the gap between automated detection and human understanding. Integrating these technologies allows for a comprehensive security framework capable of real-time detection, analysis, and response.

1.1 Problem Statement

Despite advances in cybersecurity tools, several critical problems remain:

- **Data Overload:** Millions of logs generated daily cannot be effectively analyzed manually.

- **False Positives:** Rule-based and some ML-based systems produce numerous false alerts, overwhelming analysts.
- **Delayed Response:** Manual monitoring and decision-making slow down the response to high-severity threats.
- **Lack of Explainability:** Many automated systems cannot provide human-readable insights, limiting trust and effective decision-making.
- **Integration Challenges:** Few systems combine real-time ML detection, LLM-based analysis, and automated mitigation in a unified framework.

The primary objective of this project is to address these challenges by developing a hybrid AI-driven threat detection and automated incident response system that integrates machine learning classification, real-time monitoring, and LLM-powered explanations, while providing automated responses for high-critical alerts.

1.2 Previous Related Work

Over the past decade, researchers and industry practitioners have explored various approaches to improve threat detection and enhance cybersecurity capabilities. The rapid growth of networks and connected devices has created massive amounts of log and telemetry data, prompting the need for automated detection systems capable of handling high-volume, real-time information. Several methodologies have been proposed, each with their strengths and limitations:

- **Machine Learning-Based IDS:** Early intrusion detection systems relied on supervised and unsupervised machine learning algorithms such as Random Forest, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN) to identify anomalous patterns in network traffic. These methods were effective at recognizing known patterns and improving detection accuracy. However, many ML-based IDS solutions struggled with interpretability; analysts often found it difficult to understand why a particular alert was triggered. Additionally, these approaches were generally designed for offline analysis, limiting their applicability in real-time monitoring scenarios.
- **SIEM and SOAR Platforms:** Security Information and Event Management (SIEM) systems, such as Splunk, IBM QRadar, and Azure Sentinel, centralize logs from multiple sources and enable workflow automation.

Security Orchestration, Automation, and Response (SOAR) platforms extend SIEM capabilities by automating repetitive tasks and integrating response playbooks. While these platforms are widely used in industry, they rely heavily on predefined rules and correlation logic, making them vulnerable to emerging threats that do not match existing patterns. Moreover, analysts are still required to interpret alerts and make critical decisions, leading to operational inefficiencies.

- **Deep Learning for Anomaly Detection:** With the rise of deep learning, researchers explored the use of autoencoders, Long Short-Term Memory (LSTM) networks, and deep neural networks for detecting complex anomalies in network behavior. These models can capture intricate temporal and structural patterns in network traffic, providing higher accuracy than traditional ML methods. Despite their promise, deep learning solutions are computationally intensive, often requiring significant processing power and large datasets. They also suffer from poor interpretability, making it challenging for security teams to trust automated decisions without additional explanation.
- **Automated Response Systems:** Some studies focused on implementing automated response mechanisms, such as isolating compromised devices, blocking suspicious IP addresses, or terminating malicious processes. These systems improve response times and reduce manual workload. However, existing automated solutions frequently lack intelligent decision-making and human-readable explanations. Without contextual understanding of the threat, analysts may hesitate to trust the system's actions, limiting deployment in sensitive environments.
- **LLM Applications in Cybersecurity:** More recently, Large Language Models (LLMs) have been employed for log summarization, phishing detection, vulnerability assessment, and automated report generation. LLMs excel at providing human-readable explanations, converting complex technical logs into interpretable insights, and assisting analysts in decision-making. Nonetheless, integrating LLMs directly into real-time detection pipelines and automated response frameworks is still in its early stages.

Most current implementations operate offline or as advisory tools rather than integrated security solutions.

While each of these approaches has significantly advanced the field, there remains a gap in end-to-end solutions that simultaneously provide real-time detection, intelligent explanation, and autonomous response. Existing systems often focus on one aspect, such as detection or automation, but fail to address the entire cybersecurity workflow in a unified, adaptive framework.

1.3 Our Work

To address the limitations of existing cybersecurity solutions, this project proposes a hybrid AI-driven system that seamlessly integrates machine learning (ML), Large Language Models (LLMs), and automated mitigation strategies, delivering a complete end-to-end cybersecurity framework. The system is designed to monitor network and device logs in real time, classify threats accurately, provide human-readable analysis, and respond autonomously to critical incidents.

Offline Training: Historical logs and telemetry data are collected from network devices, servers, and endpoints. The data undergoes preprocessing to remove noise, normalize features, and encode categorical information. A machine learning model is then trained on this dataset to classify events into Low, Medium, or High threat levels, allowing the system to effectively distinguish between benign anomalies and malicious activity.

Real-Time Detection: Once deployed, the system continuously ingests incoming logs and telemetry streams. Each event is evaluated by the trained ML model, which assigns a severity level based on learned patterns. This real-time classification ensures that emerging threats are detected immediately, minimizing the time between detection and response.

LLM-Based Analysis: Low and Medium severity threats are analyzed using Large Language Models, which generate interpretable insights. The LLM summarizes anomalous behavior, explains why an event was flagged, and provides recommended mitigation strategies. This functionality enables security analysts to understand the rationale behind alerts and make informed decisions efficiently.

Automated Response: High-severity alerts trigger predefined automatic actions, such as blocking malicious IP addresses, isolating compromised devices, or terminating suspicious processes. Each automated action is recorded, and the LLM generates a detailed, human-readable report for analyst verification. This approach reduces response times, minimizes human intervention for critical incidents, and ensures high-impact threats are mitigated promptly.

The integration of these components establishes a robust, scalable, and intelligent framework. By combining ML-based detection, LLM-powered analysis, and automated mitigation, the system delivers a comprehensive cybersecurity solution capable of addressing the challenges of modern network environments effectively.

1.4 System Significance and Impact

The proposed hybrid AI-driven framework effectively addresses several critical challenges in modern cybersecurity. By combining machine learning with LLM-based analysis, the system mitigates the problem of alert fatigue and false positives, which are common in traditional rule-based systems. Analysts are provided with interpretable insights and contextualized explanations, allowing them to prioritize incidents efficiently. High-severity threats are automatically mitigated, reducing response times and minimizing potential damage.

Moreover, the system's scalable design ensures it can handle massive volumes of log data generated by modern networks and digital infrastructures, while maintaining real-time performance. The integration of detection, explanation, and response in a single framework establishes a closed-loop cybersecurity workflow, enhancing both operational efficiency and the overall security posture of an organization. By providing real-time actionable intelligence, the system enables organizations to proactively defend against evolving cyber threats and adapt quickly to emerging attack patterns.

CHAPTER 2: Methodology

2.1 Introduction to Methodology

This chapter explains the complete methodological framework used to develop the proposed hybrid AI-driven cybersecurity system. It covers all essential steps—from data acquisition, preprocessing, feature engineering, model development, pipeline creation, LLM integration, real-time monitoring, automated incident response, and finally, the technologies required to implement the entire system. The methodology ensures accuracy, scalability, reliability, and real-time operational capability suitable for modern cybersecurity environments.

2.2 Data Collection

Data is collected from diverse cybersecurity-related sources, including firewalls, IDS/IPS systems, servers, authentication logs, endpoint telemetry, and network traffic monitors. These logs contain timestamps, events, user identity information, IP addresses, port activities, process-level data, and behavioral sequences. The purpose of collecting this wide range of data is to train the system to differentiate between normal activity, unusual anomalies, and malicious threats.

2.3 Data Preprocessing

The raw collected logs often contain irregularities such as missing values, corrupted entries, inconsistent formats, and duplicated timestamps. Preprocessing includes cleaning malformed logs, standardizing timestamps, encoding categorical fields, normalizing numerical attributes, and merging logs from multiple sources. Noise reduction, outlier handling, and time alignment are performed to ensure data quality and consistency. This creates a structured dataset suitable for deeper analysis and model training.

2.4 Feature Engineering

Feature engineering transforms raw logs into meaningful attributes that reflect security-relevant behavior. This includes extracting frequency-based features (e.g., number of failed logins per minute), behavioral metrics (e.g., unusual outbound connections), statistical features (e.g., packet size distribution), and temporal correlations between related events. Additional contextual features are generated to

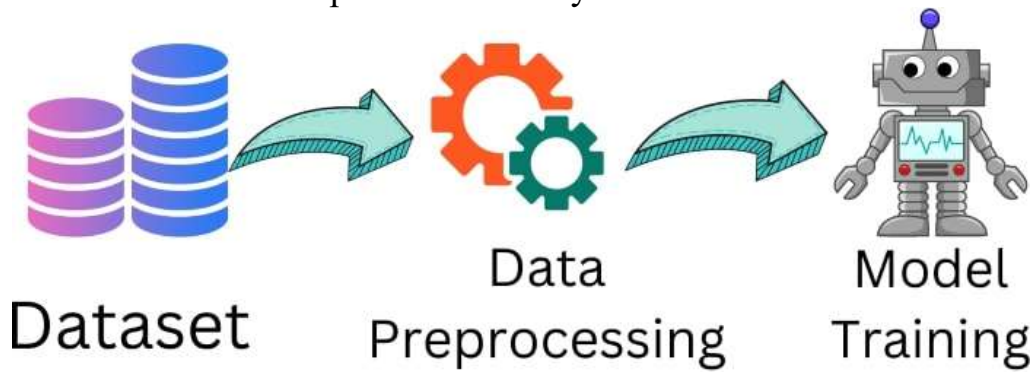
capture user-level or device-level behavior patterns. Effective feature engineering ensures the model receives clear indicators of benign vs. malicious activity.

2.5 Model Training and Ensemble Learning

The machine learning component of the system is trained on the engineered dataset to classify each event into Low, Medium, or High threat levels. Model selection focuses on accuracy, interpretability, computational efficiency, and real-time performance. Algorithms such as Random Forest, Gradient Boosting Machines, Logistic Regression, or Support Vector Machines may be used depending on the final dataset characteristics.

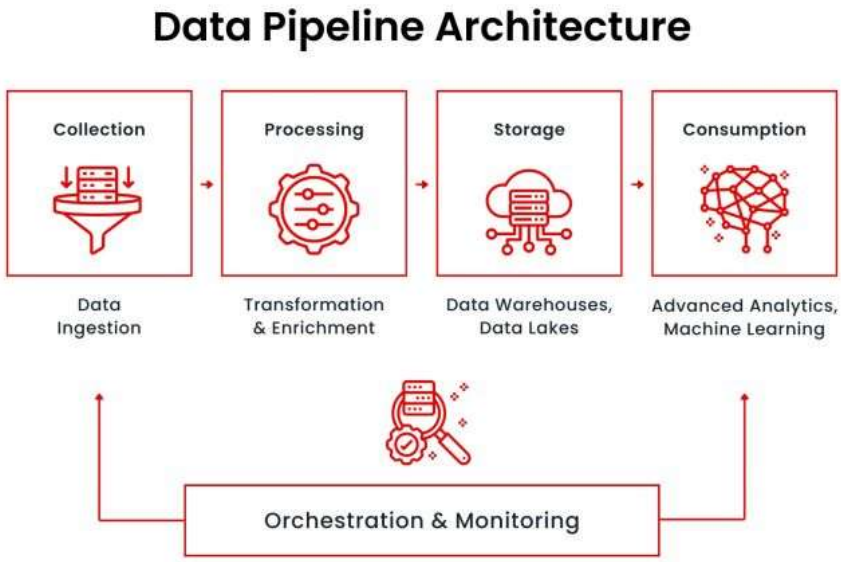
Ensemble learning is used to further improve accuracy and robustness. By combining multiple models—such as Random Forest, Gradient Boosting, or XGBoost—the system reduces noise sensitivity, improves generalization, and handles complex, high-dimensional patterns typical in cybersecurity data. Ensemble methods also help reduce false positives and false negatives, providing more stable predictions under varying network behaviors.

The final model undergoes hyperparameter tuning, cross-validation, and class imbalance handling to ensure optimal performance. The output is a low-latency, real-time classifier that can operate effectively under live network conditions.



2.6 Pipeline Construction

Pipelines ensure that the entire process from incoming logs to classified alerts runs smoothly and automatically. Each component (preprocessing, feature extraction, classification, and LLM interpretation) is integrated into a modular pipeline. This approach ensures scalability, consistent execution, and easier maintenance.



Pipelines also support real-time streaming integration using connectors such as *WebSockets, Kafka,* or *SIEM* platforms. Any new data entering the system automatically flows through the pipeline without requiring manual processing. This design is crucial for deploying the solution in production environments.

2.7 LLM-Based Analysis Layer

Large Language Models (LLMs) are incorporated as an interpretive layer on top of the machine learning classifier. When the ML model assigns a threat level to an event, the LLM analyzes the event context, explains the reasoning in human-readable language, and recommends remediation steps.

For *Low and Medium-severity* alerts, the LLM provides summaries, possible causes, and recommended actions for analysts. For *High-severity* alerts, the LLM supports automated reporting, generating detailed explanations of the issue and the subsequent automated actions executed by the system.

This layer ensures transparency, auditability, and improved decision-making by converting complex log data into understandable narratives.

2.8 Real-Time Monitoring

Real-time monitoring is achieved by connecting the system to continuous telemetry streams from network devices and servers. The system receives events instantly, processes them through the pipeline, classifies them using the ML model, and forwards them for LLM interpretation and automated response.

Streaming frameworks such *as Kafka, WebSockets,* or *SIEM* connectors enable high-throughput and low-latency monitoring. The system ensures that anomalies and threats are detected the moment they occur, minimizing risk and reducing the time between detection and mitigation.

2.9 Automated Incident Response

Automated response is a central component of the system designed to automatically contain or mitigate high-severity threats. Playbooks are defined to specify the proper response actions for each type of critical threat. These actions may include:

- *Blocking malicious IP addresses or domains*
- *Isolating affected devices from the network*
- *Terminating malicious or suspicious processes*
- *Disabling compromised accounts*
- *Resetting authentication tokens*
- *Updating firewall or access control rules*
- *Triggering MFA or session revalidation*
- *Uploading response logs into a centralized evidence database*

These response operations can be executed using orchestration tools, *SOAR* platforms, or custom *automation scripts*. After execution, the *LLM* generates a detailed explanation for analysts and records the entire incident for auditing and compliance.

2.10 Required Technologies

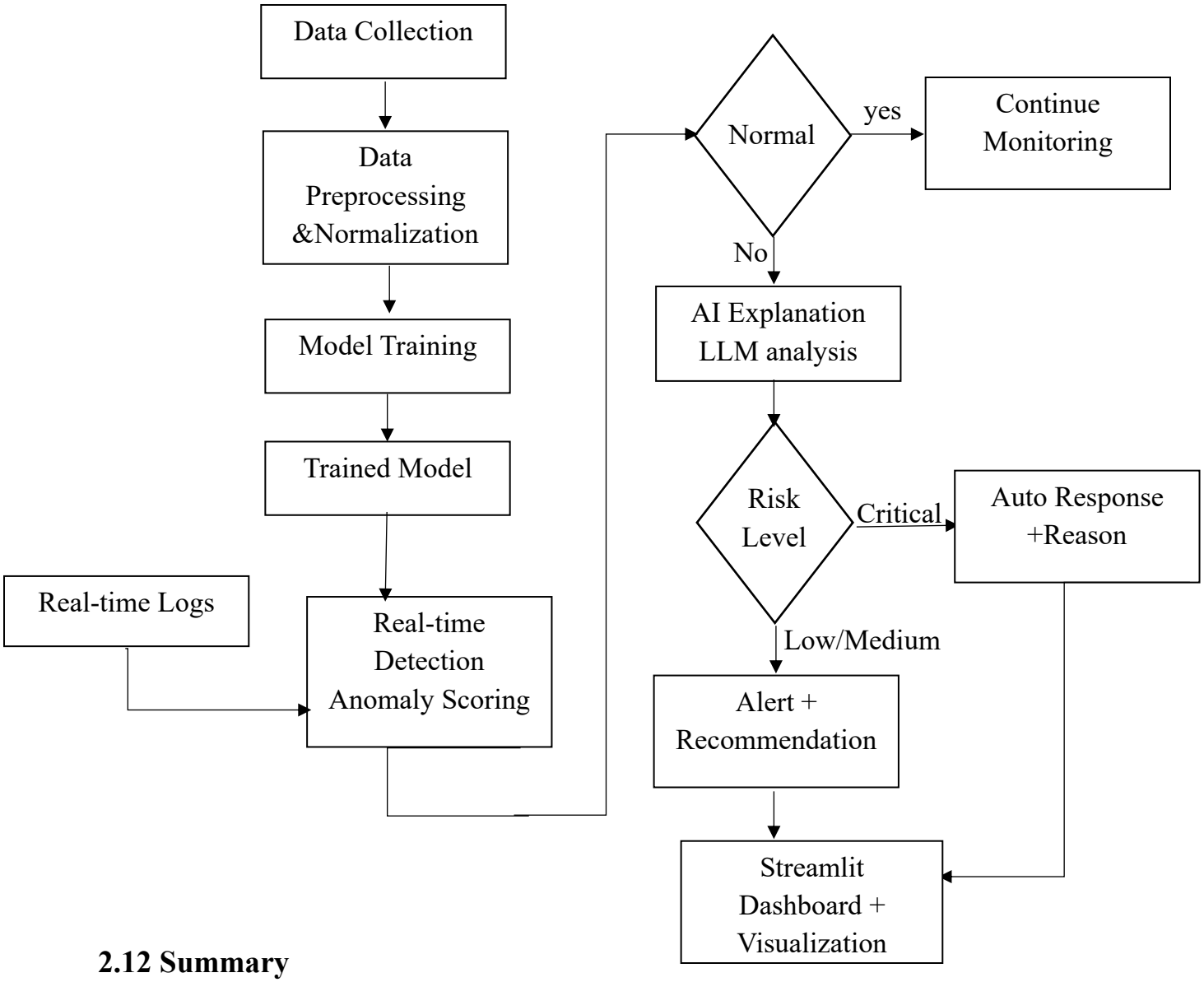
For development environments (IDEs): Jupyter Notebook for data exploration, PyCharm and VS Code for backend development and integration
For data ingestion: Python log collectors
For preprocessing & ML: Python, Pandas, NumPy, Scikit-Learn
For streaming & real-time: Apache Kafka, WebSockets, SIEM connectors (e.g., Splunk, Elastic)
For LLM integration: OpenAI API, local LLM frameworks (Llama, GPT-based solutions), Hugging Face
For automated response: SOAR platforms (Cortex XSOAR, Shuffle), Python automation scripts
For storage: MySQL
For interface/front-end: Streamlit

2.11 Algorithm:

1. **Collect historical and real-time logs** from sources (firewalls, Sysmon, Suricata, Zeek, endpoints, authentication logs, github, kaggle).
2. **Preprocess logs**: clean, normalize, parse, time-align, and encode categorical fields.
3. **Perform feature engineering**: extract traffic, behavioral, temporal and correlation features; produce the final feature set.
4. **Split dataset** into training, validation and test sets; apply class-balancing if required.
5. **Train base ML models** (e.g., Random Forest, XGBoost, LightGBM) on the training set.
6. **Apply ensemble strategy** (bagging/boosting/stacking) to combine base models and obtain the final classifier.
7. **Optimize and validate** the ensemble using cross-validation and hyperparameter tuning; measure metrics (precision, recall, F1, AUC).
8. **Serialize and deploy** the trained ensemble model for real-time inference (export model artifacts).

9. **Ingest real-time events** into the pipeline; run preprocessing and feature extraction on each incoming event or micro-batch.
10. **Invoke model inference** to assign a threat level: Low, Medium, or High.
11. **Route result for handling**:
    - If **Low** → Log and store; generate an LLM summary for analyst reference.
    - If **Medium** → Notify analyst with LLM-generated explanation and recommended actions.
    - If **High** → Execute automated incident response playbook and record the action.
12. **LLM post-processing**: generate human-readable reports for every handled alert (explanation, evidence, suggested mitigations, action log).
13. **Record feedback**: accept analyst feedback and store it for retraining/active learning.
14. **Periodic retraining & evaluation**: schedule model retraining using new labeled data and feedback to improve detection over time.

SYSTEM ARCHITECTURE



2.12 Summary

This chapter presented the complete methodological framework for building the hybrid AI-driven cybersecurity system. Through structured data collection, preprocessing, feature engineering, ensemble-based model training, pipeline construction, LLM-based interpretation, real-time monitoring, and automated incident response, the system provides an end-to-end approach to modern cybersecurity defense. The integration of learning algorithms and automation ensures fast, accurate, and intelligent mitigation of threats in real operational environments.

CHAPTER 3: CODE AND OUTPUT

This chapter is in progress

3.1 Dataset Description and Analysis

- 3.1.1 Dataset Overview and Statistics
- 3.1.2 Data Distribution and Imbalance Analysis
- 3.1.3 Exploratory Data Analysis Findings

3.2 Data Preprocessing Implementation

- 3.2.1 Data Cleaning and Normalization
- 3.2.2 Categorical Feature Encoding
- 3.2.3 Temporal Feature Extraction
- 3.2.4 Handling Missing Values and Outliers

3.3 Feature Engineering Process

- 3.3.1 Network-based Feature Extraction
- 3.3.2 Behavioral Pattern Features
- 3.3.3 Statistical and Temporal Features
- 3.3.4 Feature Selection and Dimensionality Reduction

3.4 Model Development and Training

- 3.4.1 Algorithm Selection and Justification
- 3.4.2 Ensemble Model Architecture
- 3.4.3 Hyperparameter Tuning Process
- 3.4.4 Cross-Validation Strategy
- 3.4.5 Class Imbalance Handling Techniques

3.5 LLM Integration for Explainability

- 3.5.1 LLM Configuration and Setup
- 3.5.2 Prompt Engineering for Threat Analysis
- 3.5.3 Natural Language Explanation Generation
- 3.5.4 LLM Response Formatting and Structuring

3.6 Real-Time Pipeline Development

- 3.6.1 Streaming Data Ingestion Implementation

Chapter 4: Future Work and Conclusion

4.1 Future Work

- **Performance Optimization**: While the current ML and LLM integration is effective, real-time processing of high-volume logs can be computationally intensive. Future work can focus on optimizing model inference speed and memory consumption, possibly through GPU acceleration or model compression.
- **Extended ML Models**: Expanding support to deep learning models such as LSTM networks, Graph Neural Networks (GNNs), or Transformers may improve detection of complex temporal and relational patterns in network data.
- **Hybrid Security Approaches**: Integrating this system with other security tools, such as SIEM, SOAR, and endpoint protection platforms, could create a comprehensive security ecosystem. The system can later support cross-platform threat correlation and multi-source analytics.
- **Adaptive Learning**: Implementing continual learning or online learning would allow the ML component to adapt to emerging threat patterns without requiring full retraining.
- **Usability and Explainability**: Enhancing the LLM layer with domain-specific threat knowledge can improve alert explanations, recommended actions, and analyst trust. Future research may explore automated alert prioritization using risk scores or dynamic playbooks.
- **Edge and Cloud Deployment**: Optimizing the system for deployment on cloud and edge environments can enable organizations to perform distributed, low-latency threat detection across multiple locations and devices.
- **Dataset Expansion and Real-World Testing**: Applying the system on diverse real-world datasets, including industrial control systems and IoT telemetry, would further validate its effectiveness and robustness.

4.2 Conclusion

The proposed hybrid AI-driven cybersecurity system successfully combines machine learning, LLM reasoning, and automated incident response to address modern cyber threats. By integrating anomaly detection with human-readable analysis, it mitigates alert fatigue, improves threat detection accuracy, and accelerates response to critical events.

The modular design ensures scalability and adaptability for diverse organizational environments. High-severity threats are automatically mitigated, while low- and medium-severity alerts are supplemented with LLM-generated explanations, enhancing analyst understanding and decision-making.

Future enhancements, including performance optimization, hybrid deployment, extended ML models, and adaptive learning, can further strengthen the system, making it a reliable solution for real-time cybersecurity operations.

REFERENCES

[1] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, pp. 1-22, 2019.

[2] M. Ring, S. Wunderlich, D. Grüdl, D. Landes, and A. Hotho, "Flow-based benchmark data sets for intrusion detection," in Proceedings of the 16th European Conference on Cyber Warfare and Security, Dublin, Ireland, 2017, pp. 361-369.

[3] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proceedings of the 4th International Conference on Information Systems Security and Privacy, Funchal, Portugal, 2018, pp. 108-116.

[4] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525-41550, 2019.

[5] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in Proceedings of the National Aerospace and Electronics Conference, Dayton, OH, USA, 2015, pp. 339-344.

[6] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in Proceedings of the International Conference on Platform Technology and Service, Jeju, South Korea, 2016, pp. 1-5.

[7] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, "On the effectiveness of machine and deep learning for cyber security," in Proceedings of the International Conference on Cyber Conflict, Tallinn, Estonia, 2018, pp. 371-390.

[8] OpenAI, "GPT-4 Technical Report," OpenAI, 2023. [Online]. Available: https://cdn.openai.com/papers/gpt-4.pdf

[9] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou et al., "A survey of large language models," arXiv preprint arXiv:2303.18223, 2023.

[10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 1877-1901.

[11] M. H. G. E. Din, A. S. A. Ghani, and S. M. Shamsuddin, "Review of anomaly detection techniques in network intrusion detection system," in Proceedings of the International Conference on Soft Computing and Data Mining, Johor, Malaysia, 2018, pp. 267-276.

[12] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 2018.

[13] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," Computers & Security, vol. 45, pp. 100-123, 2014.

[14] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," IEEE Access, vol. 5, pp. 21954-21961, 2017.

[15] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 2017, pp. 1285-1298.