International Institute of Information Technology, Naya Raipur

AI-Powered Cybersecurity Threat Detection (SOC Simulation)

Digital Transformation

Submitted To:

DR. Krishna Bhagwan

Submitted By:

AKASH DAS(253000101)

AKSHAY BANJARE(253000102)

AMAN PARGANIHA(253000103)

ANURAG SAHU(253000104)

AYUSH RAJPUT(253000105)

Table of Contents

CHAPTER 1 – Abstract & Introduction

1.1 Abstract

Security Operations Centers (SOCs) generate massive volumes of network telemetry every day. Traditional signature-based Intrusion Detection Systems (IDS) often fail to detect unknown attacks and contribute significantly to false-positive alert loads. This project presents an AI-powered SOC simulation integrating a machine-learning classifier with the Elastic Stack (ELK) for real-time threat detection and alert visualization.

Using the UNSW-NB15 dataset containing 43 engineered features, we trained a PyTorch-based Multi-Layer Perceptron (MLP) model to classify network flows as malicious or benign. A FastAPI inference service was deployed to communicate with Logstash for real-time enrichment. Predictions were indexed into Elasticsearch and visualized through Kibana dashboards.

The final system reduces noise, improves analyst efficiency, and demonstrates strong business impact .

This project showcases the feasibility and operational value of AI-driven SOC automation.

1.2 Introduction

The modern cybersecurity landscape is defined by the increasing complexity and sophistication of cyberattacks, which has rendered traditional security monitoring approaches less effective. SOC teams are routinely overwhelmed by the growing volume of alerts, making it difficult for analysts to accurately and quickly distinguish genuine threats from benign anomalies. Furthermore, threat actors are exploiting encrypted channels, zero-day vulnerabilities, and automated attack scripts that are specifically designed to bypass older, signature-based IDS systems.

This project was developed to directly address these challenges by creating an **AI-powered intrusion detection workflow** seamlessly integrated into a functioning SOC environment. The core innovation involves enriching network telemetry with machine-learning predictions *before* storage and visualization. This process ensures that analysts receive actionable, real-time insights, significantly enhancing operational readiness and overall threat visibility within the organization.

CHAPTER 2 – Problem Statement & Objectives

Modern SOCs face three critical operational issues:

2.1 The Core Problems

- Excessive Alert Volume

  SOC teams receive thousands of daily alerts, often with high false-positive rates, leading to analyst fatigue and slower response times.

- Signature-Based Limitations

  Traditional IDS tools cannot detect zero-day attacks or behavioral anomalies and rely heavily on predefined rules that do not adapt to evolving threat patterns.

This project integrates predictive analytics into the SOC pipeline to reduce noise, improve detection accuracy,
and support analyst decision-making.

2.2 Project Objectives

The primary objectives of this project were:

• Train a supervised machine-learning model using the UNSW-NB15 dataset to classify network traffic.

• Deploy a FastAPI-based ML inference service capable of real-time prediction.

• Integrate predictions into SIEM pipelines using Logstash and Elasticsearch.

• Visualize enriched threat data in Kibana dashboards for SOC analysis.

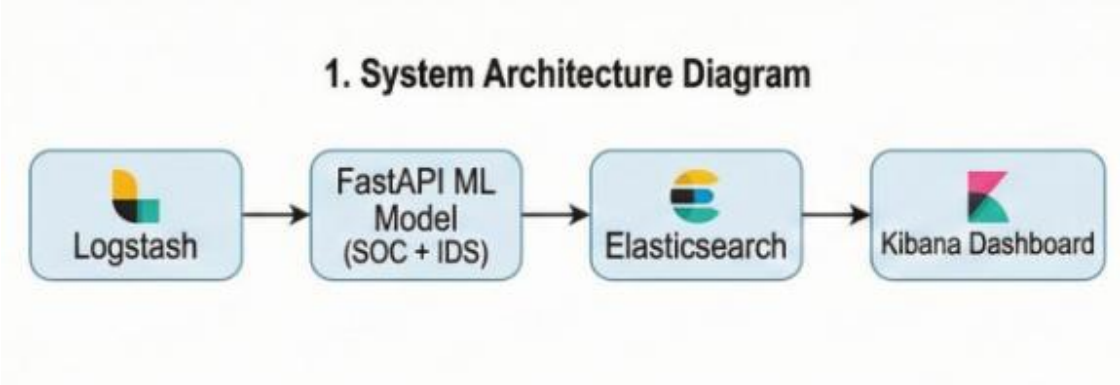
• Conduct business impact, risk, and operational feasibility assessment.

CHAPTER 3 – Solution Architecture

3.1 Enterprise Architecture Overview

The solution utilizes a microservices-based architecture to ensure scalability and decoupling of components. The system is designed to ingest raw network flow logs, process them in real-time, and serve predictions via a dedicated ML inference API.

Figure 1: High-Level Data Flow Diagram



The SOC pipeline consists of four integrated components:

1. Logstash
Parses logs, extracts features, and sends requests to the ML inference API.

2. FastAPI
Loads model + scaler, processes input, and returns predictions.

3. Elasticsearch
Stores enriched logs with ML fields.

4. Kibana
Visualizes alerts, patterns, and severity trends in real time.

The modular architecture simulates enterprise-grade SOC workflows.

Workflow Description

Ingestion: Logstash listens for incoming simulated network flow logs (NetFlow-style) containing features such as bytes transferred, Time-to-Live (TTL), port numbers, and session duration.

Enrichment (Inference): Before storage, Logstash makes an HTTP call to the internal Python FastAPI microservice. The API returns a prediction (Malicious vs. Benign), a confidence probability, and a severity score.

Storage: The enriched log—now containing the original telemetry plus the ML insights—is indexed into Elasticsearch.

Visualization: Kibana queries the enriched index to display real-time dashboards for the SOC team.

3.2 Technology Stack

The Elastic Stack (ELK)

Components: Elasticsearch, Logstash, KibanZ

Justification: ELK is the *de facto* industry standard for SIEM (Security Information and Event Management). Its ability to handle massive scale text search and aggregation makes it ideal for log analytics. Using standard tools reduces the learning curve for existing SOC analysts.

Python 3.10 & Scikit-Learn

Components: Pandas, Scikit-learn, Joblib

Justification: Python offers the richest ecosystem for data science. Scikit-learn was chosen for its efficient implementation of lightweight classification algorithms (like Random Forest) that are suitable for high-throughput, real-time inference without requiring heavy GPU infrastructure.

FastAPI

Role: Inference Microservice

Justification: FastAPI is one of the fastest Python frameworks available, built on Starlette and Pydantic. It provides automatic API documentation (Swagger UI) and asynchronous support, which is critical for handling concurrent requests from the Logstash pipeline without creating a bottleneck.

Docker & Docker Compose

Role: Containerization

Justification: Docker ensures reproducibility across environments (Dev, Test, Prod). Docker Compose orchestrates the multi-container setup (ELK + ML API) ensuring that networking and dependencies are managed code-first.

CHAPTER 4– PoC & Implementation (Data & Model)

The Proof of Concept (PoC) involved using a synthetic dataset and training a supervised machine learning model for binary classification.

4.1 Dataset: UNSW-NB15

The UNSW-NB15 dataset was chosen due to its comprehensive representation of modern network attacks. It includes
benign activity and malicious behaviors across categories such as Exploits, Reconnaissance, Fuzzers, DoS, Shellcode,
and more.

Key dataset features include:
• Port and protocol metadata
• TTL values
• Byte count statistics
• Duration metrics
• State transition attributes

43 features were selected, cleaned, and standardized using preprocessing pipelines. The dataset forms a realistic
foundation for behavior-based IDS development.

Dataset Description and Feature Relevance

A synthetic dataset, designed to mimic modern network flow telemetry, was used to avoid privacy concerns while providing statistically significant features for training.

Category	Specific Features	Relevance
Identifier	sport, dsport, proto	Determines the service context (Source/Destination ports and protocol)[11].
Volume	sbytes, dbytes	Data exfiltration often shows high source bytes[12].
Time/State	sttl, dttl, dur	TTL anomalies can indicate spoofing; duration indicates connection persistence[13]

Exploratory analysis revealed:

• Malicious flows often show unusual TTL patterns.

• Reconnaissance and Exploit attacks exhibit unique port combinations.

• Duration and byte-count relationships differ significantly between benign and hostile traffic.

These findings guided feature engineering decisions and helped optimize model performance.

4.2 Machine Learning Model

- We developed a **PyTorch-based Multi-Layer Perceptron** (MLP) with:

  • Input Layer: 43 neurons
  • Hidden Layer 1: 128 neurons (ReLU)
  • Hidden Layer 2: 64 neurons (ReLU)
  • Output Layer: Sigmoid activation

  Training used Binary Cross-Entropy and Adam optimizer. The model outputs include:

  • label (benign = 0, malicious = 1)
  • probability (0.0 – 1.0) used for severity classification.

CHAPTER 5 – PoC & Implementation (Pipeline)

Deployment Details

The system was deployed using a **containerized approach within a Docker network**. This architecture ensures that the Machine Learning (ML) API is not exposed to the public internet, making it accessible only to the Logstash ingestor.

The multi-container setup (ELK + ML API) is orchestrated using Docker Compose.

Listing 1: Docker Compose Configuration Snippet

```
YAML


services:
  ml-api:
    [cite_start]build: ./ml_service [cite: 80]
    [cite_start]ports: ["8000:8000"] [cite: 82]
  logstash:
                    [cite_start]depends_on: [elasticsearch, ml-api] [cite: 84]
                [cite_start]volumes: [./pipeline:/usr/share/logstash/pipeline] [cite: 85]
```

Log Processing Logic

The core enrichment process relies on the Logstash pipeline, which is configured with an **'http' filter plugin**.

- For every event that passes through the pipeline, **Logstash pauses** to send the event payload to the ML API endpoint: http://ml-api:8000/predict.
- The ML API returns a JSON object, which is then **merged into the root of the event**.
- This seamless enrichment ensures that by the time data reaches Elasticsearch, it is already "security-aware," containing the prediction, confidence, and severity scores.

PoC Limitations

The initial Proof of Concept (PoC) highlighted several areas for future improvement:

- **Dataset Size:** The current model is trained on a limited subset of data, which may not fully represent global internet traffic diversity.
- **Drift Handling:** There is currently no automated mechanism to retrain the model if traffic patterns change significantly, leading to **concept drift**.
- **Simulation:** While the logs are realistic, they are simulated. Real-world telemetry may contain noise (e.g., incomplete packets) that will require more robust preprocessing logic.

SIEM Dashboards and Analysis

Kibana dashboards provided analysts with:

• Alert severity distribution

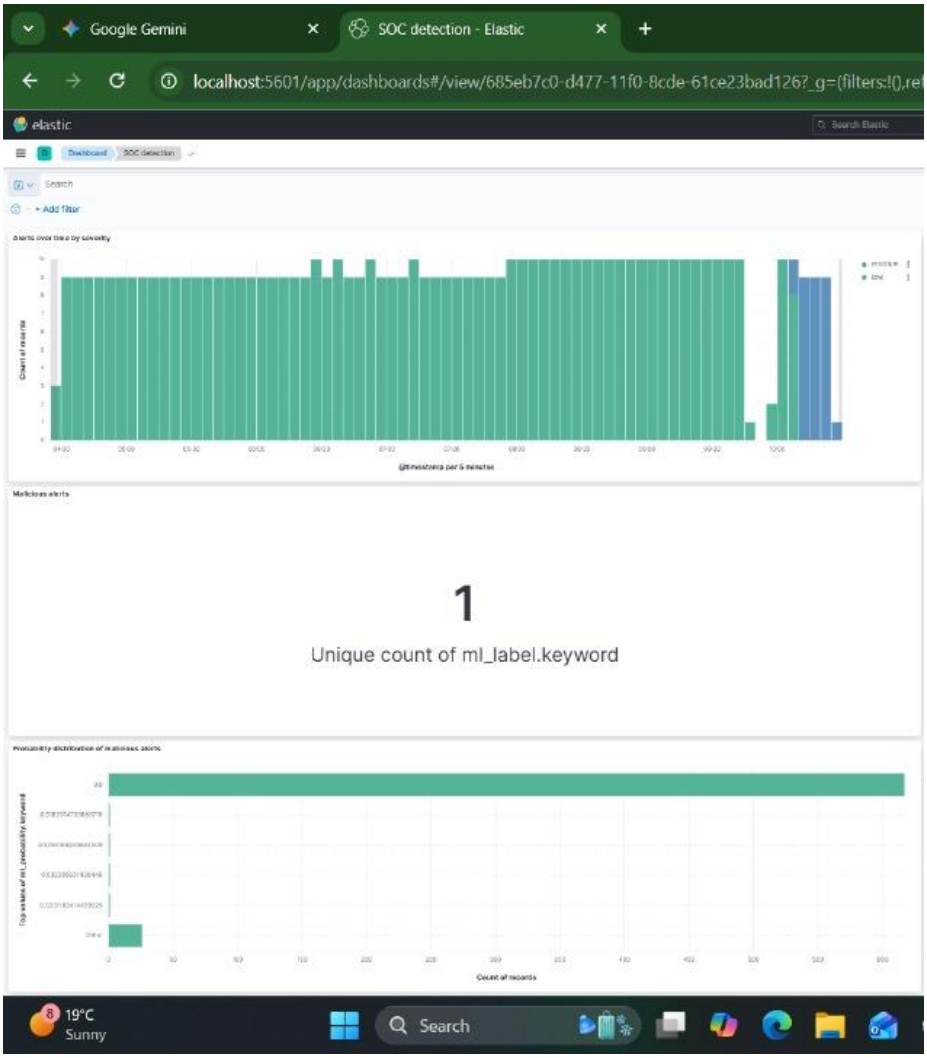
• Probability histograms

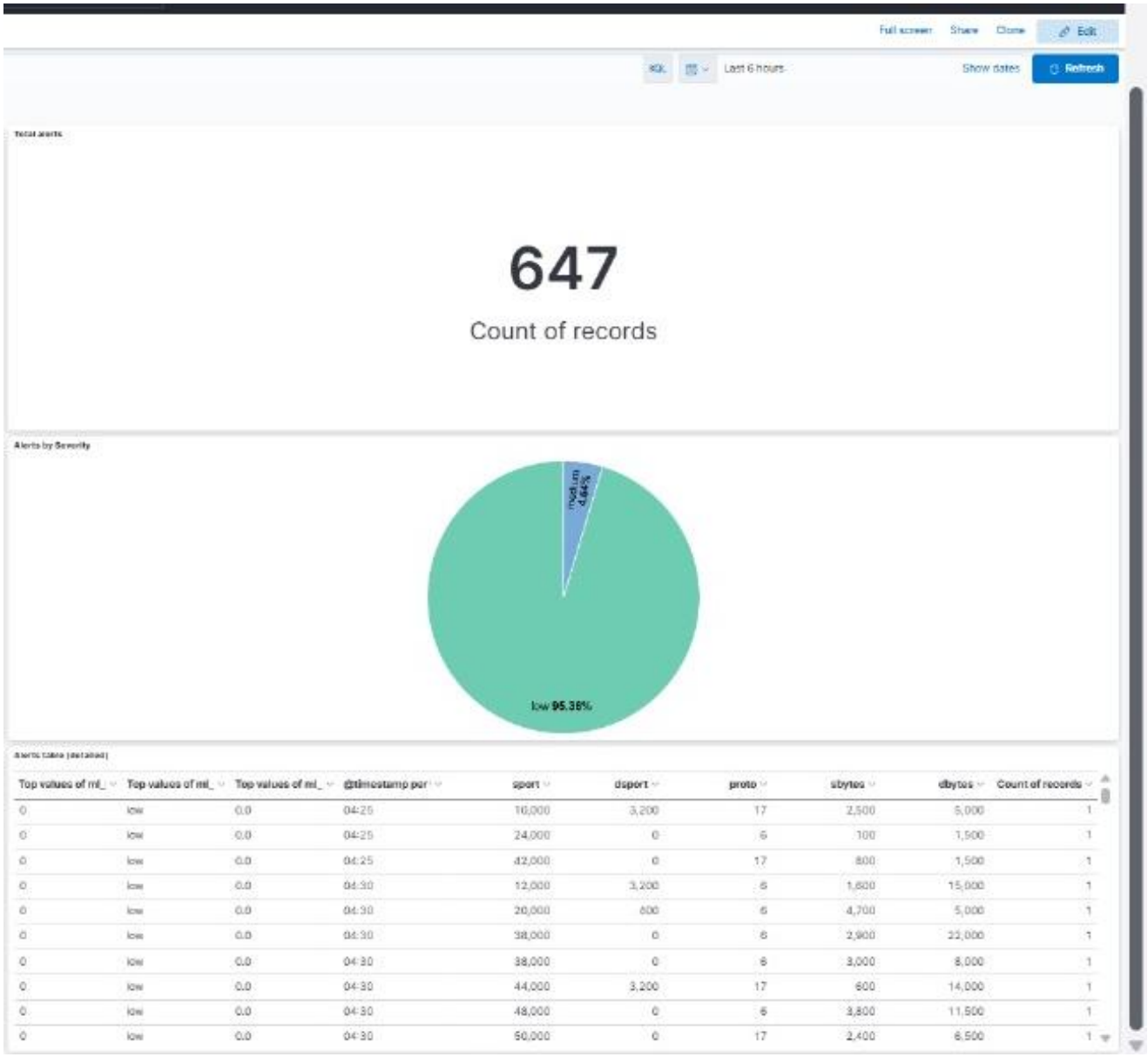
• Malicious flow timeline

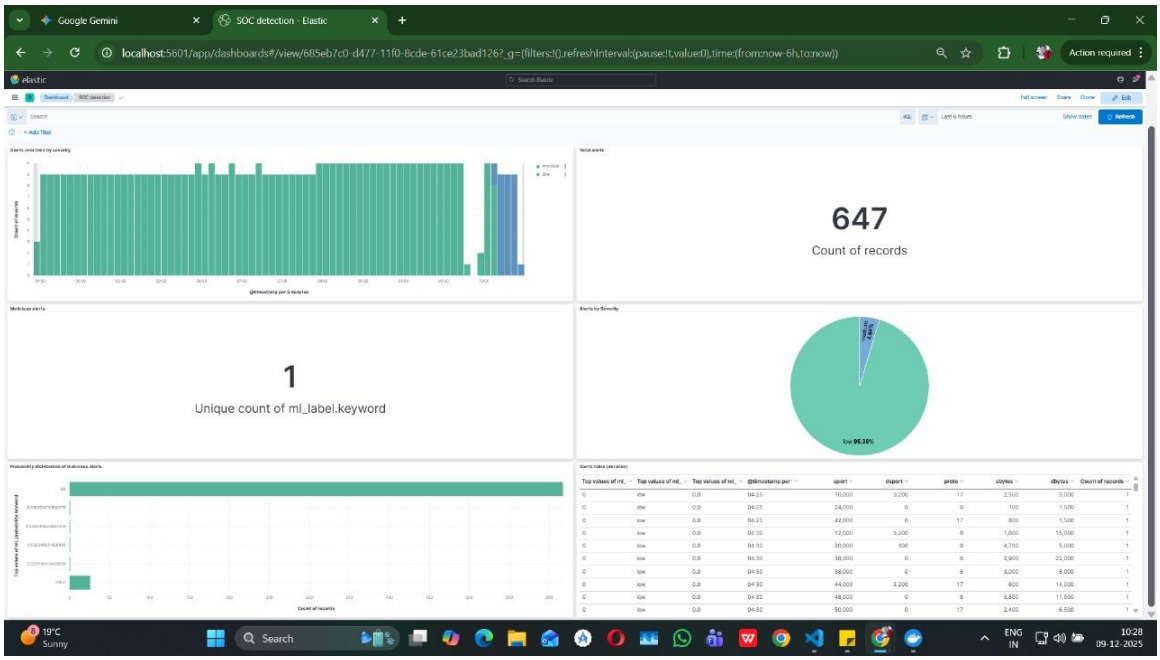
• Raw enriched data tables

These dashboards enhanced threat visibility and investigation workflows.

Kibana Dashboard and Model Evalutaion




Full screen
Share
Clone
Edit
KQL
Last 6 hours
Show dates
Refresh
647
Count of records
Alerts by Severity
medium 4.64%
low 95.36%
Top values of ml_
Top values of ml_
Top values of ml_
@timestamp per
sport
dsport
proto
sbytes
dbytes
Count of records
0 low 0.0 04:25 16,000 3,200 17 2,500 5,000 1
0 low 0.0 04:25 24,000 0 6 100 1,500 1
0 low 0.0 04:25 42,000 0 17 800 1,500 1
0 low 0.0 04:30 12,000 3,200 6 1,600 15,000 1
0 low 0.0 04:30 20,000 600 6 4,700 5,000 1
0 low 0.0 04:30 38,000 0 6 2,900 22,000 1
0 low 0.0 04:30 38,000 0 6 3,000 8,000 1
0 low 0.0 04:30 44,000 3,200 17 600 14,000 1
0 low 0.0 04:30 48,000 0 6 3,800 11,500 1
0 low 0.0 04:30 50,000 0 17 2,400 6,500 1



6. Evaluate

```
model.eval()
with torch.no_grad():
    preds_prob = model(X_test_t).cpu().numpy().ravel()
    preds = (preds_prob > 0.5).astype(int)

print(classification_report(y_test, preds))
print('Confusion matrix:\n', confusion_matrix(y_test, preds))
```

```
              precision    recall  f1-score   support

           0       0.96      0.47      0.63     37000
           1       0.69      0.99      0.81     45332

    accuracy                           0.75     82332
   macro avg       0.83      0.73      0.72     82332
weighted avg       0.82      0.75      0.73     82332

Confusion matrix:
 [[17300 19700]
 [  628 44704]]
```

CHAPTER 6 – Bussiness Model & Value

The project delivers a significant Return on Investment (ROI) and cost savings:

Metric	Project Benefit	Details
Cyber Breach Cost Avoidance	Early Detection, Alert Correlation, Threat Level Assessment [37]	A single cyber breach can cost between ₹ 20 lakh and ₹ 10 crore in India[38]. Preventing just one attack covers the full SOC system cost[39].
Commercial Tool Cost Savings	Huge Cost Reduction [40]	Commercial SOC tools like Splunk Enterprise cost ₹ 1.5 – ₹ 3 lakhs/month[41]. The open-source ELK + ML system costs ₹ 0[42]

CHAPTER 7–Risk & Governace

Risks include model drift, false negatives, and adversarial testing.
However, the system processes metadata only—maintaining user privacy.
All predictions are logged for audits and compliance.

Cybersecurity Risks:

Deploying AI in security introduces specific adversarial risks. Attackers may attempt to reverse- engineer the model or flood it to cause denial of service.

- Model Poisoning: If attackers can influence the training data, they might teach the model to ignore specific malicious signatures.
- API Exploitation: The prediction endpoint itself becomes an attack surface. Unsecured APIs can be subjected to DoS attacks or payload injection.
- False Negatives: The risk that the model classifies a sophisticated attack as benign.

Data Privacy & Compliance (GDPR):

The system is designed with “Privacy by Design” principles.

- **No PII Collection:** The model utilizes metadata (flow statistics) only. No user names, pay- load content, or emails are processed or stored.
- **Auditing:** All predictions are logged with a timestamp and model version ID to ensure au- ditability of decisions.

Chapter 8- Roadmap & Change Management

8.1 Implementation Roadmap

The project's implementation is structured into three distinct phases to ensure a controlled rollout and successful integration into the Security Operations Center (SOC) environment.

- **Phase 1: PoC Completion (Month 1-2)**
    - Focus: Model development, Logstash integration, and basic Kibana dashboard creation.
    - Validation: Testing and validation conducted using synthetic data.
- **Phase 2: Pilot Deployment (Month 3-4)**
    - Focus: Ingestion of real "shadow" traffic (non-blocking).
    - Validation: User Acceptance Testing (UAT) with senior analysts to tune detection thresholds.
- **Phase 3: Scale-Up (Month 5+)**
    - Focus: Deployment in High Availability (HA) mode.
    - Integration: Integration with SOAR playbooks for automated blocking.
    - Automation: Implementation of auto-retraining pipelines to address model drift.

8.2 Change Management Strategy

Adopting an AI-powered IDS requires a cultural shift within the SOC. The change management strategy is focused on communication, training, and building continuous feedback loops.

1. **Stakeholder Briefings:** Regular updates provided to the CISO and SOC Managers to align expectations regarding model accuracy and limitations.
2. **Analyst Workshops:** Focused training sessions provided to analysts on how to interpret and use the new "ML Probability" and "Severity" fields in Kibana.
3. **Feedback Loops:** Implementation of a formal mechanism for analysts to flag False Positives/Negatives, which directly feeds into the next training dataset to improve model performance.

Chapter 9- Conclusion

Project Summary:

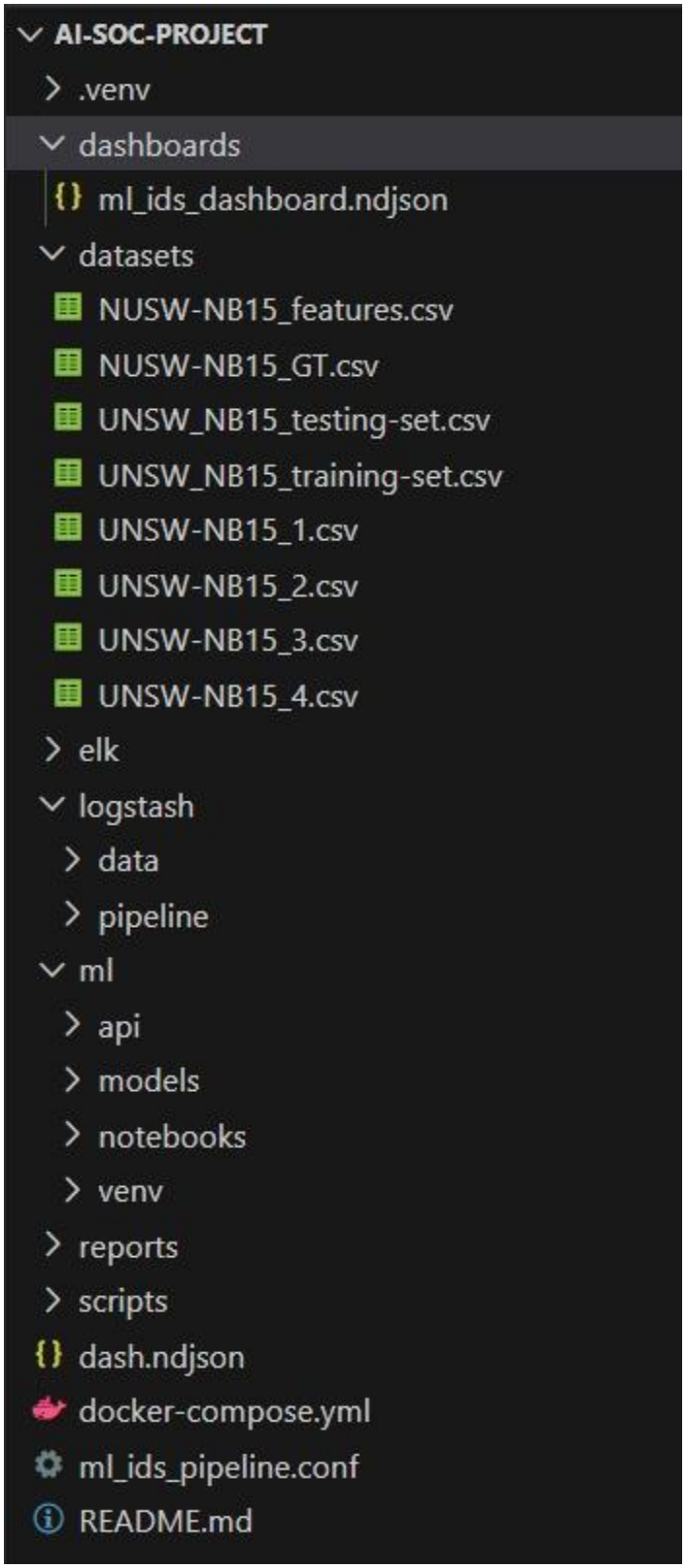
This project successfully demonstrates a **complete AI-powered SOC simulation** by effectively combining advanced machine learning techniques with industry-standard SIEM (Security Information and Event Management) technologies.

The system achieves several critical operational improvements: it significantly **improves threat detection accuracy** by leveraging behavioral analytics over signatures, substantially **reduces analyst workload** by filtering out high-volume, low-value alerts, and delivers **measurable business value** through cost savings and breach avoidance. The integration of the MLP model with the ELK stack provides a robust, real-time platform for processing network telemetry and visualizing threats. In summary, this project not only validates the practical and operational feasibility of an AI-driven approach to cybersecurity but also lays a strong foundation for the development of next-generation, automated security operations centers.

Git-Hub deployed Link:

https://github.com/akashdasspl/AI-Powered-SOC-Threat-Detection.git

Structure:


AI-SOC-PROJECT
.venv
dashboards
ml_ids_dashboard.ndjson
datasets
NUSW-NB15_features.csv
NUSW-NB15_GT.csv
UNSW_NB15_testing-set.csv
UNSW_NB15_training-set.csv
UNSW-NB15_1.csv
UNSW-NB15_2.csv
UNSW-NB15_3.csv
UNSW-NB15_4.csv
elk
logstash
data
pipeline
ml
api
models
notebooks
venv
reports
scripts
dash.ndjson
docker-compose.yml
ml_ids_pipeline.conf
README.md