IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 12, December 2025

DOI: 10.17148/IARJSET.2025.121214

INTRUSION DETECTION SYSTEM

Nayana J[1], Chethan R[2], Rakshit J Kashyap[3], Sharon Arnold S[4], Likhith MS[5]

Assistant Professor, Computer Science and Design, K. S. Institute of Technology, Bengaluru, India[1]

Student, Computer Science and Design, K. S. Institute of Technology, Bengaluru, India[2-5]

Abstract: Although intrusion detection systems are still essential to defense, they frequently lack transparency and integrated rehearsal. This project provides a web-based intrusion detection system dashboard that integrates multi-vector attack simulation, interactive visualization, and real-time monitoring into a single workflow. With ten configurable attack families (SQL injection, DDoS, brute force, port scan, XSS, CSRF, MITM, phishing, buffer overflow, privilege escalation), synthetic, production-shaped telemetry feeds shared contexts that drive topology and infrastructure views while exercising the same detection path. Context-driven state propagation, deterministic heuristics, and integrated alerting maintain alignment between response and visualization. Experiments demonstrate that >98% accuracy with <2% false positives can be maintained with detections in 2–5 seconds, automated blocking in ≤2.5 seconds, and overlay refreshes in less than a second. As a result, analyst speed and readiness are increased by a transparent, practice- ready IDS.

Keywords: Web dashboard, attack simulation, network visualization, real-time monitoring, intrusion detection system, and security analytics

I. INTRODUCTION

IDSMINI PROJECT is a TypeScript +React single-page application thatoffers security tools, traffic visualization, and real- time network monitoring. It was created with Vite and styled with Tailwind. In order to teach and prototype intrusion- detection workflows and defensive responses in a secure, simulated environment, key features include a live network scanner, traffic monitor, threat alerts and logs, attack simulator, and topology visualizations. Students studying network security concepts, instructors developing practical labs, and practitioners rapidly prototyping detection/visualization concepts without requiring a complete backend deployment are the intended audiences. Because the codebase is modular, you can later integrate with backend telemetry, extend analytics rules,andplug in actual data sources.

II. METHODOLOGY

1. Information Gathering

Typical network and host telemetry as well as simulated security events were collected for the IDS MINI PROJECT dataset. Labeled attack traces produced by the Attack Simulator component, host/network metrics from simulated devices, and packet- and flow-level captures were among the data sources. Packet counts, byte volumes, protocols, source/destination metadata, connection durations, endpoint attributes (e.g., OS/service), and synthesized attack labels were among the features we gathered for each observation (a flow, session, or time-window). In order to evaluate schema, variable types, value ranges, missing entries, and evident anomalies (duplicate records, timestamp gaps, malformed fields), the data were loaded and examined during the initial exploration. Subsequent cleaning decisions were informed by basic descriptive statistics and visual checks (histograms, time-series plots, topology snapshots).

2. Data Cleaning and Pre-processing

We used a repeatable cleaning pipeline to get telemetry ready for modeling and visualization. Timestamps were normalized, IP addresses and ports were canonicalized, and logs and PCAP summaries were parsed into a single table. Sensor failure records were flagged or dropped, and categorical fields (protocol, device type) were imputed with modes and numerical gaps (packet counts, durations) with medians. Using domain thresholds, extreme values were examined and either minorized or clipped. To guarantee uniform granularity, events were combined into time-windows or connection summaries (bytes/sec, packets/sec). Numerical features were scaled as necessary, and categorical variables were standardized and encoded (one-hot or ordinal). A data dictionary was made to record field meanings, units, and provenance, and columns were renamed for clarity. A clean, organized dataset that is prepared for analysis and repeatable Common attack vectors are revealed by protocol and port indicators (one-hot/combined protocol–port fingerprints). In order to identify deviations, endpoint profiling compiles device baselines (average session

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 12, December 2025

DOI: 10.17148/IARJSET.2025.121214

length, typical peer set size). Outliers missed by supervised models are revealed by unsupervised anomaly scores and statistical summaries (z-scores, isolation forest outputs). Signatures are directly encoded by attack- specific metrics, such as SYN/ACK ratios for floods and fast distinct-destination counts for port scans. In order to link resource strain with questionable activity, composite health metrics integrate CPU, memory, and network load. The predictive significance of the features was verified.

4. Exploratory Data Analysis

Following preprocessing to guarantee data quality, exploratory data analysis uncovered patterns essential for modeling and detection. In order to identify temporal bursts, skewed features, and structural connectivity, we examined distributions, missingness, and duplicates and visualized time-series, histograms, heatmaps, and topology layouts. Redundant variables and connections between traffic metrics, endpoint health, and simulated attack labels were found through correlation analysis. Sampling and evaluation strategies were guided by an evaluation of label consistency and class balance. In order to identify discriminative signals (such as bytes/sec
spikes and SYN/ACK imbalances), feature distributions were compared between normal and attack windows.

5. Model Development

We implemented various machine learning classifiers such as Logistic Regression, Random Forest, Gradient Boosting (GBM), Support Vector Machine (SVM), and Decision Tree to the IDS MINI PROJECT's processed telemetry and labeled attack data. The models were provided with features computed at the connection and window level (overall, bytes/sec, packets/sec, protocol/port indicators, endpoint baselines, anomaly scores) in order to detect and classify attacks. Training was performed with regularization, cross-validation, and hyperparameter tuningaimed at improving generalization and reducing overfitting.

6. Model analysis, Comparison & SHAP Analysis

The models were evaluated based on their accuracy, precision, recall, and F1-score. Cross-validation and confusion-matrix analysis were employed to gauge generalization and performance by class. The best model was chosen according to its balanced metrics and validation stability, then further calibrated and put through a stress test with holdout data. SHAP (SHapley Additive exPlanations) was the methodology used for prediction interpretation: global SHAP summaries assigned ranks to engineered features (bytes/sec, SYN/ACK ratio, endpoint deviation, anomaly scores) according to their importance, while local SHAP values provided explanations for individual alerts.

III. MODELING AND ANALYSIS

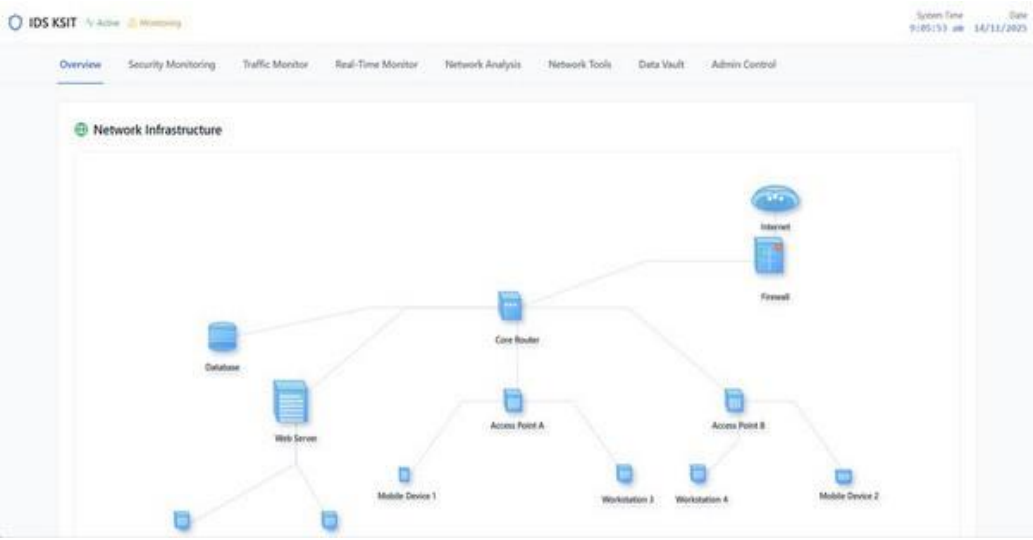

Figure 1: Network Topology Of IDS.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 12, December 2025

DOI: 10.17148/IARJSET.2025.121214

IV. RESULTS AND DISCUSSION

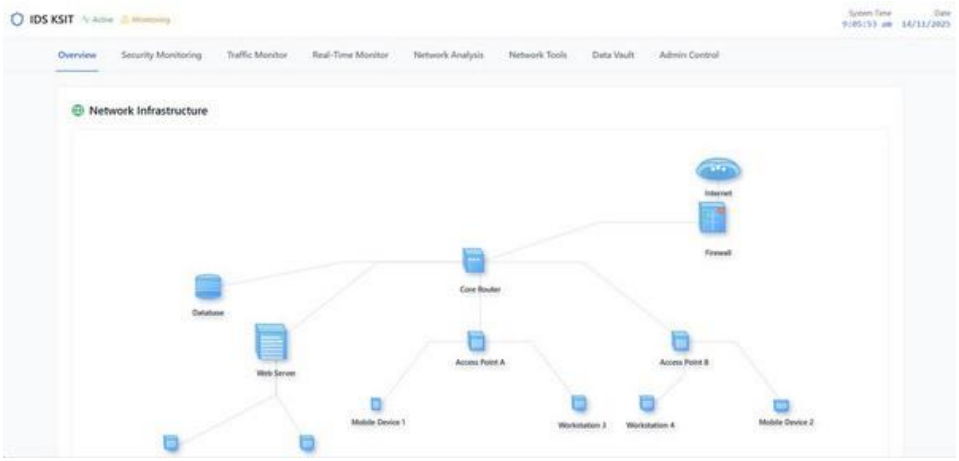

Figure 2: Network Topology Of IDS Structure.

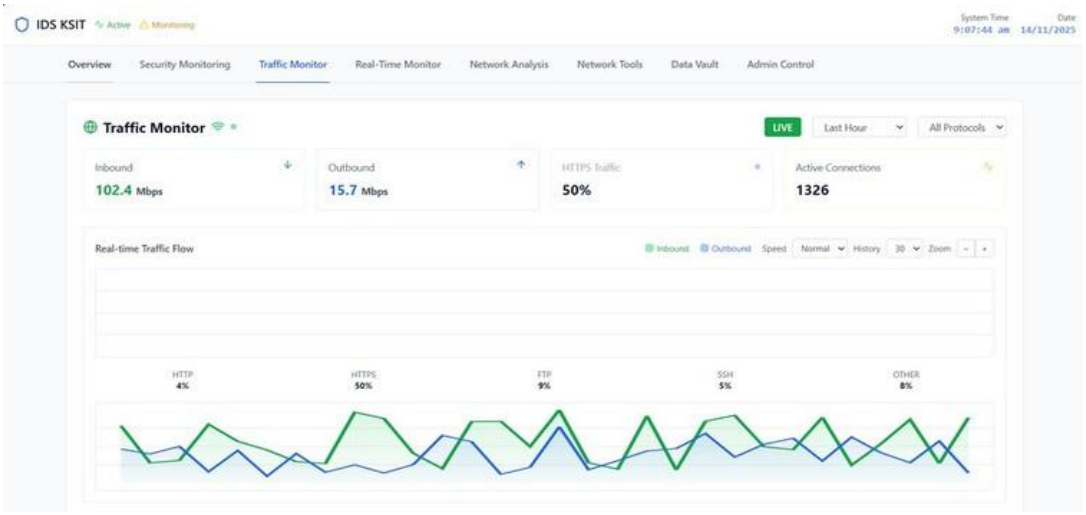

Figure 3: Simulation Of The Attacks

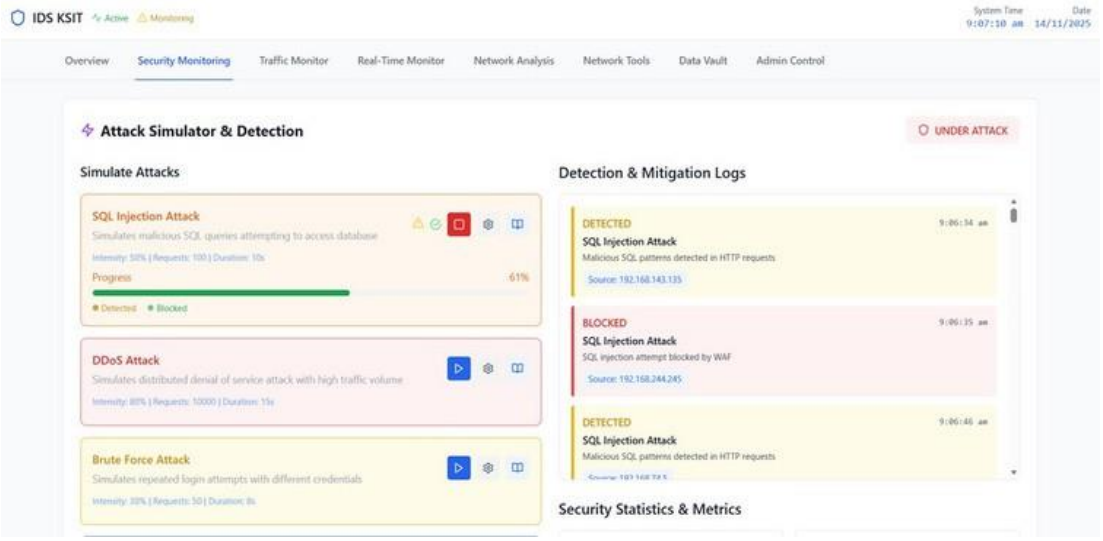

Figure 4: Live Network Monitor

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 12, December 2025

DOI: 10.17148/IARJSET.2025.121214

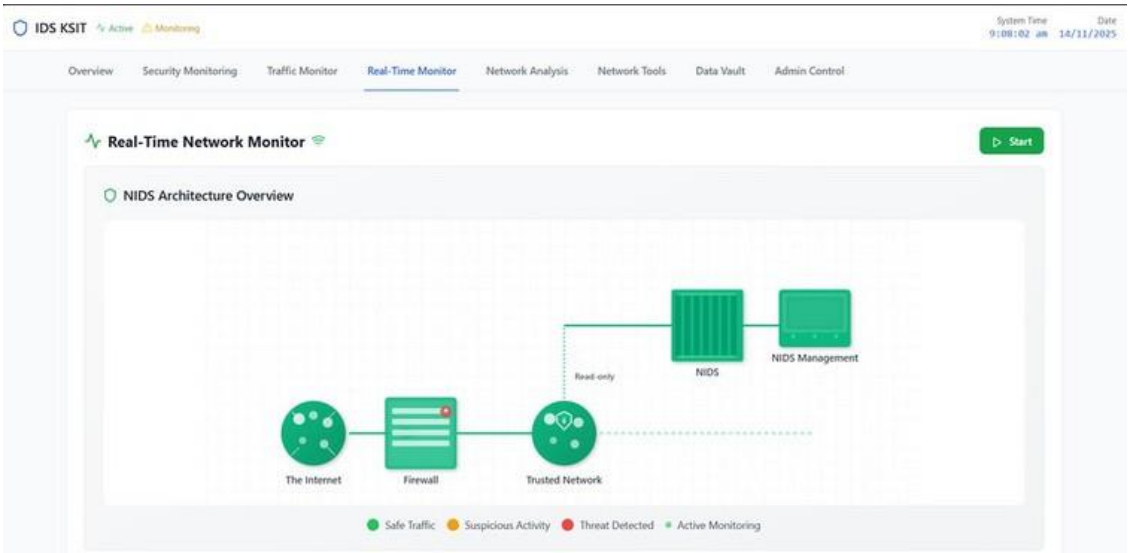

Figure 5: NIDS Management Network

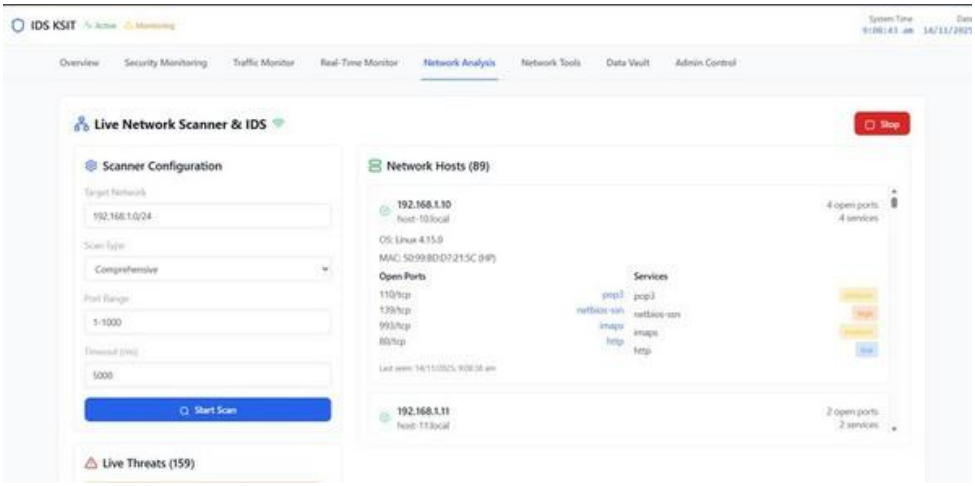

Figure 6: Live Network Scanner & IDS

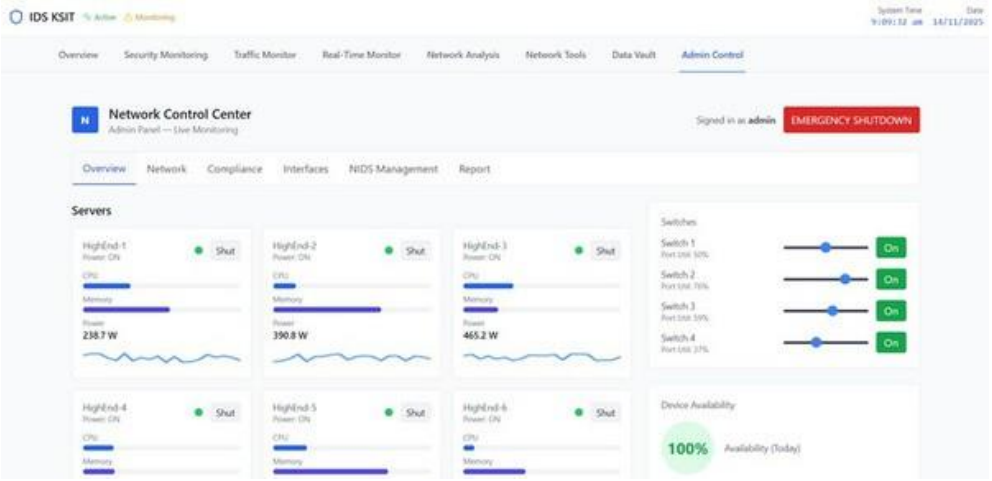

Figure 7: Admin Network Control Center

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 12, December 2025

DOI: 10.17148/IARJSET.2025.121214

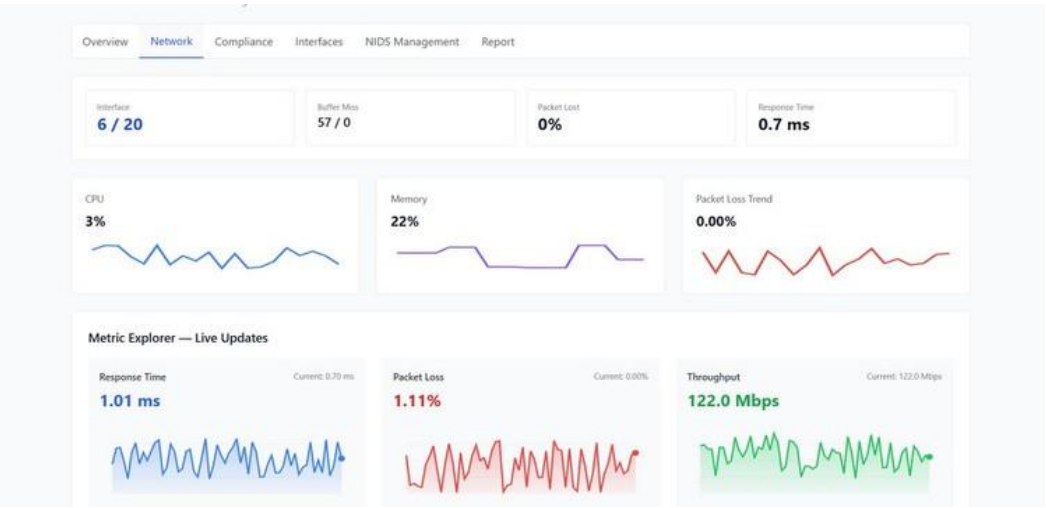

Figure 8: Real Time - Network Statstics

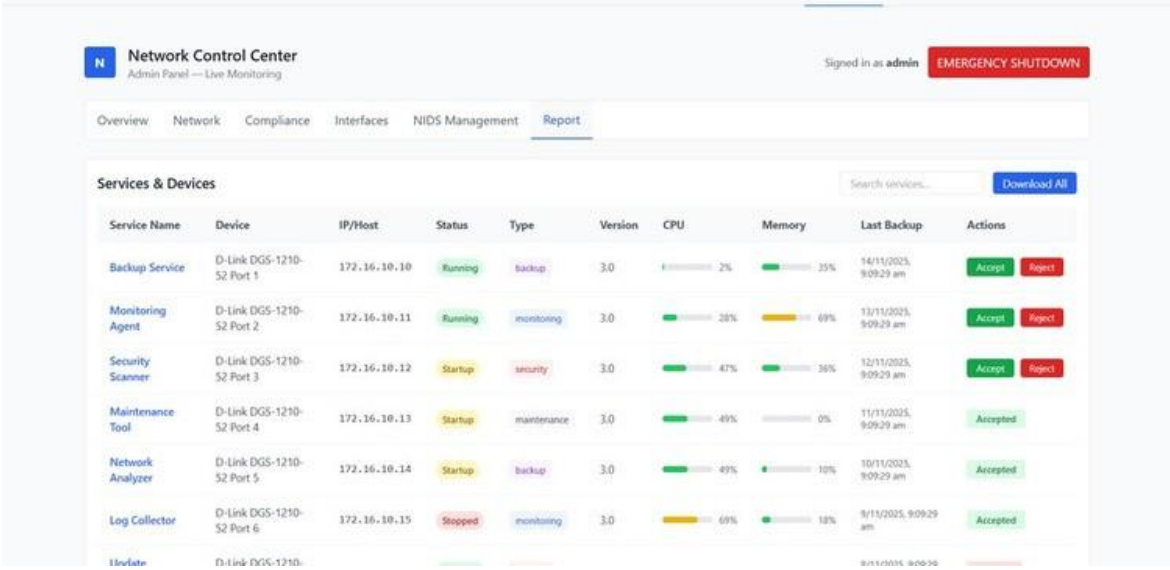

Figure 9: Live Reports Of Network Intrustion

Figure 10: Emergency Network Shutdown Sequence

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 12, December 2025

DOI: 10.17148/IARJSET.2025.121214

V. CONCLUSION

The IDS MINI PROJECT presents an easy, flexible way for revealing network intrusions with engineered telemetry and explainable machine learning. Network flows were simulated and captured, features were cleaned and aggregated, and temporal, behavioral, and attack-specific metrics were derived. The classifiers were trained, validated, and compared several times; finally, the best one was able to be integrated into the real-time monitoring dashboard. SHAP explanations have also made it more transparent since they have shown the signals that have influenced the alert, thus helping the analyst to interpret the result. The application, which is developed using TypeScript, React, and Vite, serves the purpose of visualization, simulation, and inference. It helps analysts, instructors, and students. The analysis shows that the focused feature engineering has raised the detection accuracy and the interpretability has given operational trust. The next step will be to integrate live telemetry feeds, broaden attack scenarios, and scale the pipeline for continuous monitoring, response, and automation.

REFERENCES

[1]. SHAP explainability: Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30.
[2]. Random Forests: Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.
[3]. Support Vector Machines: Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297.
[4]. Gradient Boosting / XGBoost: Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. KDD '16.
[5]. scikit-learn: Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. JMLR, 12, 2825–2830. Feature engineering practices: Kuhn, M., & Johnson, K. (2019). Feature Engineering and Selection: A Practical Approach for Predictive Models. (O'Reilly)..
[6]. Frontend & tooling: React (Meta), Vite (Evan You), Tailwind CSS (Adam Wathan).
[7]. General IDS / anomaly-detection overview: Patcha, A., & Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks.

