AI-Driven Alert Prioritization & Fatigue Reduction Framework for SOCs

Student Name: Hiteesh Pushpendra Kondepati
Student ID: S3021074
Course: MSc. Computer Science
Submission Date: 13/01/2026
Module Leader: Dr. Zia Ush Shamszaman
Module Supervisor: Hannah Stothards

Table of Contents

Introduction

Overview

- Organizations are today overly dependent on integrated systems, cloud-based environment, and real-time data flow to facilitate important processes in their operations. Issues such as this digital transformation has made it faster in terms of productivity and connectivity but has also made the attack surface of cyber threats grow exponentially.
- Security Operations Centers (SOCs) are the front-line defense, and they continuously observe large volumes of security alerts created by intrusion detection, firewalls, endpoint monitoring systems and threat intelligence systems.
- The alert fatigue happens when SOC analysts receive too many security alerts, which in turn causes fatigue and slowness in responding, and failure to detect true threats consequently posing economic and reputational risks to organizations.

Figure: Main Functions of a SOC

- As thousands of alerts are received by analysts on a daily basis, and resources at their positions are limited, proper and prompt alert categorization has become highly important, particularly with cyberattacks increasingly getting sophisticated.
- Machine Learning is the most effective approach since it automates the classification of the alerts and can adjust to the changing patterns of attacks as opposed to the traditional rule-based systems that use predefined signatures.
- This paper evaluates and compares the performance of several machine learning models, such as Logistic Regression, Random Forest, XGBoost, Support Vector Machine, and Neural Networks, based on benchmark datasets of cybersecurity (UNSW-NB15 and CICIDS2017) to determine how they reduce false positives, enhance threat detection, and can be used in practice as SOC.

Introduction

Background and Context

1. Essentiality

‡ Online security is of vital importance to the existence of an organization.
‡ SOCs scan and react to attack with the help of SIEM, IDS/IPS, and EDR.
‡ Hundreds of alerts are received by analysts every hour and some are repetitive.
‡ Triage by hand results in alert fatigue and the loss of severe threat.
‡ Classification of alerts should be automated so that the number of analysts decreases.

2. Age of Data & Machine Learning

‡ Cybersecurity produces high amounts of network, log, and user data.
‡ Classical rule-based systems do not deal with the changing and unknown attacks.
‡ Machine Learning makes it possible to detect threats dynamically and in a data-driven way.
‡ Such algorithms as Random Forest, XGBoost and Neural Networks deal with complex patterns.
‡ Explainable AI (SHAP, LIME) enhances analysts trust and transparency.

3.Challenges, Opportunities & Relevance

‡ Major issues: concept drift, noises, interpretability, and imbalance in the data.
‡ Requirement of life-long learning and understandable models in risky settings.
‡ ML-based alert classification leads to a quicker reaction and improved resource distribution.
‡ Study is timely owing to the increased volumes of alert and use of AI in SOCs.
‡ Research is at par with the industry and regulatory orientation on Explainable AI and ethical use.

Introduction

Aim and Approach of the Study

- ***Data Selection and Preparation:*** The dataset used to do the experiment will be the UNSW-NB15 and CICIDS2017 databases, which consist of labelled network traffic of normal and malicious activity.
- ***Data Preprocessing:*** Removing and normalizing data, as well as encoding it to be processed by machine learning algorithms.
- ***Feature Engineering:*** The choice and extraction of usable features that user behavioural as well as contextual features of network traffic.
- ***Model Development:*** Model training and model testing of several classification models, including Logistic Regression, Random Forest, XGBoost, SVM, and Neural Networks.
- ***Model Evaluation:*** These metrics can be used to compare models based on accuracy, precision, recall, F1-score, and ROC-AUC and find the most successful classifier.
- ***Performance Analysis:*** Measuring the increase in the reduction of alerts, the rate of detecting alerts, and the interpretability of the model.

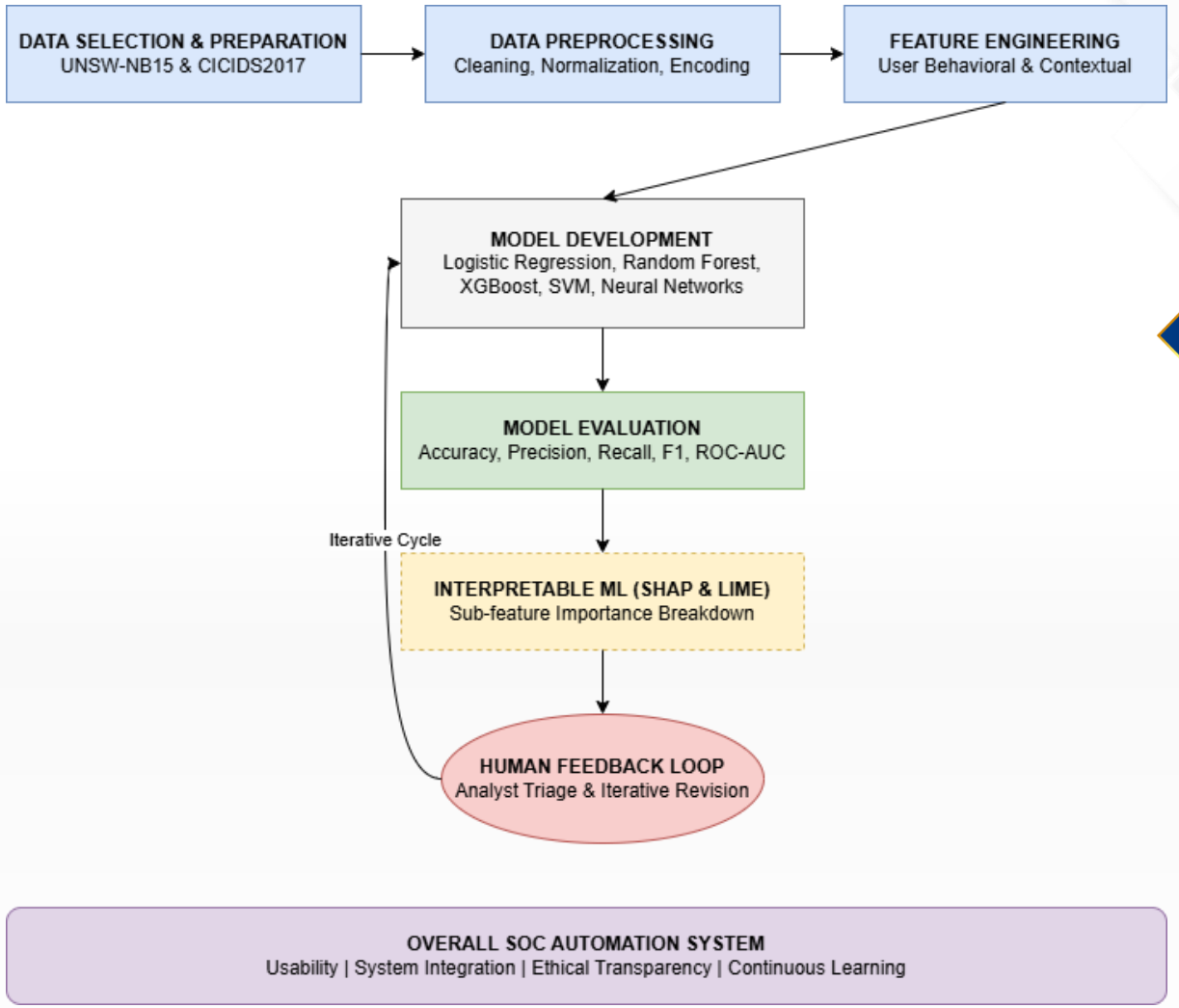


Figure: Aim and Approach of the Study

Introduction

Problem Statement & Research Question

‡ The research question that will be used in this dissertation is:

Can we develop a unified, machine-learning-driven SOC framework that automates alert triage, integrates analyst learning, and predicts escalation likelihoods—while maintaining transparency and interpretability through explainable AI?

‡ The question is an indication of both the technical and operations facets of the issue. It seeks to determine whether machine learning can differentiate benign and malicious alerts with an acceptable degree of accuracy, interpretability, and efficiency to make a more effective contribution to the actual SOC activities.

‡ The secondary sub-questions comprise:

1. *What machine learning classifiers (e.g., Logistic Regression, Random Forest, XGBoost, SVM, Neural Network) are the best with regards to their performance in precision, recall, F1-score, and ROC-AUC on cybersecurity alert data?*
2. *What is the effect of preprocessing and feature engineering process on the model performance and generalization ability?*
3. *Is it possible to enhance the interpretability of model predictions to SOC analysts by using explainability tools like SHAP or LIME?*
4. *What are the trade-offs between the accuracy of models, computation and interpretability of using ML-based alert classification systems?*

‡ All of these questions shape up the research design, experimentation and evaluation process all through the dissertation.

Introduction

Research Objectives

- Objective 1: To perform an in-depth analysis of the literature and studies regarding AI-based systems to manage alerts.
- Objective 2: To prepare and preprocess cyberspace security data (UNSW-NB15 and CICIDS2017) to train and evaluate the model successfully.
- Objective 3: To deploy and train various machine learning classification models in alert classification.
- Objective 4: To compare and contrast the performance of a model on a technical and practical basis.
- Objective 5: To improve the understanding of interpretability by explainable AI methods.
- Objective 6: To introduce a comparative analysis and suggest the most effective machine learning model to classify alerts of SOC.

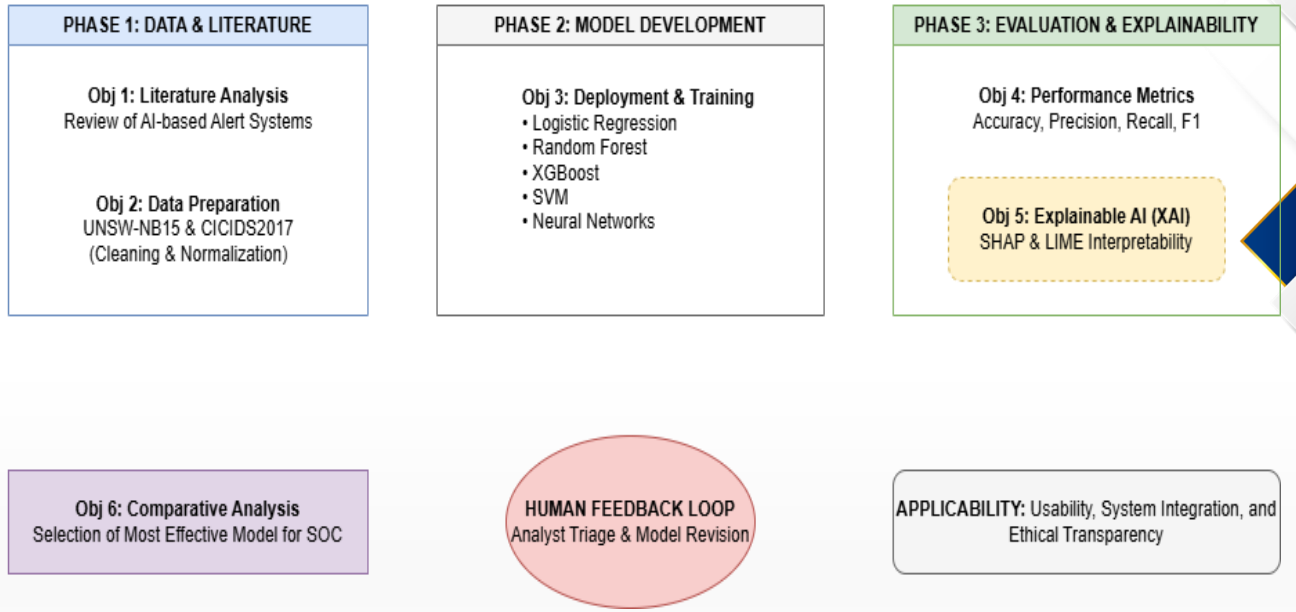


Figure: Research Objectives

Introduction

Analytical Framework

‡ This research adopts an applied experimental design to develop and evaluate a comprehensive machine learning pipeline for automating high-volume security alert triage in Security Operations Centers (SOCs). The pipeline transforms raw network flow records into a prioritized alert queue, assigning each alert a unified priority score (0–10) to guide analyst investigations and reduce overload from false positives.

‡ **Core stages include:**

1. Data acquisition using benchmark datasets UNSW-NB15 and CICIDS2017 for reproducibility and ethical considerations.
2. Rigorous preprocessing: data cleaning, handling missing/infinite values, categorical encoding (one-hot/label), normalization/standardization, and stratified train-test splits to prevent information leakage.
3. Feature engineering to extract meaningful indicators like packet rates, connection durations, and protocol patterns.
4. Model development: XGBoost for tabular classification, LSTM networks for capturing short-term temporal dependencies in sequences (10-step windows), and DBSCAN density-based clustering for alert consolidation and outlier detection.
5. Unified scoring integrating probabilistic outputs with clustering signals; human-in-the-loop feedback mechanism dynamically adjusts weights for adaptation.
6. Evaluation focuses on both technical metrics (precision, recall, F1, ROC-AUC) and operational utility (alert volume reduction, threat retention rates).

‡ This hybrid approach ensures robustness, interpretability via SHAP, and practical alignment with real-world SOC workflows. Include diagram of pipeline stages.

Introduction

Scope and Limitations

- This dissertation evaluates the effectiveness of machine learning models for security alert classification in a controlled, offline setting.
- Due to limited SOC feedback data and computational constraints, the study focuses on alert classification and explainability only.
- Five ML models are compared using public datasets, emphasizing performance, interpretability, scalability, and practical relevance for future SOC adoption.

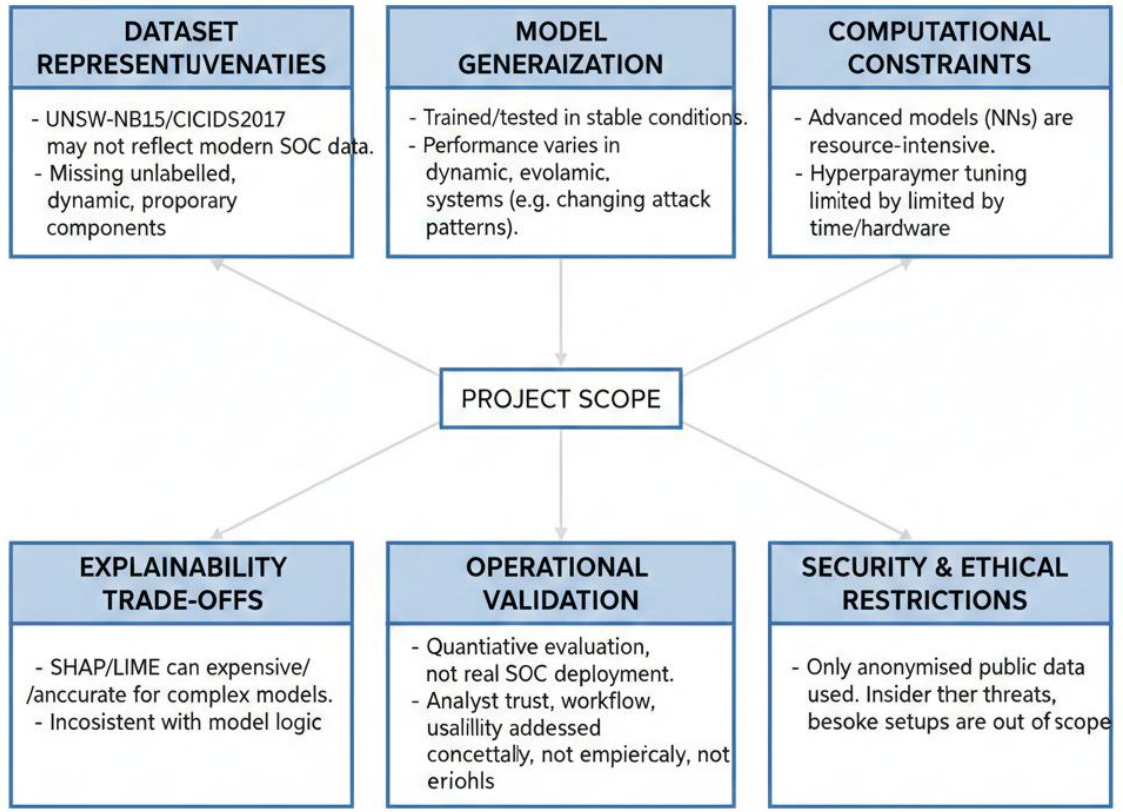


Figure: Limitations

Literature Review

- In this chapter, the review will examine literature on the areas of machine learning in intrusion detection, human-AI collaboration, explainable AI in cybersecurity, and integration barriers into operations.
- Past comparative research (Kaushik & Goel, 2021; Ma'aji et al., 2023; Malik & Singh, 2022; Naseer et al., 2018; Sharma, 2025) was used to establish benchmarks on the performance used to track the performance of ML-based intrusion detection and alert filtering. These papers demonstrate ensemble and deep-learning models to be highly accurate, yet opaque or, in other words, non-integrated to operations.
- There is a recent shift to human centered SOC automation (Stiborek et al., 2023; Zhou et al., 2024). In these studies, human-in-the-loop frameworks are adopted to have analyst expertise in retraining models and active learning. Besides, Lin et al. (2023) and Alhaidari et al. (2024) consider the explanatory AI (XAI) tool usage among SOCs and mention possible advantages of transparency and enhanced trust in the analyst experience and system acceptance.
- The challenges identified by Ogino, 2015 and Tang et al., 2016 regarding deployment include concept drift, latency, and continuous retraining (deployment challenges) and imply the necessity of flexible design systems. On the whole, the reviewed studies suggest SOC frameworks and a balance between accuracy, explainability, adaptability, and human-AI collaboration.
- We provide a replicable comparative baseline and conceptual framework in this project that connects automated classification, explainability, and analyst feedback loops to provide the initial foundation of future escalation prediction research and real-time implementations.

Literature Review

S. No.	Citation	Main Concept	Remarks	Limitations
1	Kaushik & Goel (2021)	Comparative evaluation of ML classifiers for intrusion detection using NSL-KDD dataset	Ensemble models, particularly Random Forest, achieved higher detection rates and reduced false positives; feature selection and preprocessing are critical	Dataset imbalance and feature redundancy remain unresolved; based on benchmark dataset
2	Ma'aji et al. (2023)	Comparison of ensemble and simple ML models for intrusion detection	Ensemble-based models showed higher accuracy and robustness; emphasizes generalisation to real-world traffic	Uses mostly clean academic datasets, which may not reflect real SOC alert noise
3	Malik & Singh (2022)	Supervised ML methods for intrusion detection	Even simpler ML models outperform rule-based systems when data preparation is effective; highlights importance of feature engineering	Model performance heavily dependent on dataset quality rather than model complexity
4	Naseer et al. (2018)	Deep learning approaches for network anomaly detection	Deep models (CNNs, LSTMs, autoencoders) improve AUC and precision-recall metrics	High computational cost, large training requirements, and reduced interpretability
5	Ogino (2015)	Real-time intrusion detection using ML in SOC environments	Highlights deployment challenges such as concept drift, latency, and streaming data handling	Focuses on deployment issues rather than optimizing model accuracy
6	Tang et al. (2016)	LSTM-based intrusion detection in software-defined networks	Temporal and sequential patterns improve detection performance	Focus limited to sequence-based models; higher computational complexity
7	Sharma (2025)	Comparative study of ML models for intrusion detection	XGBoost balances accuracy, interpretability, and computational efficiency; stresses analyst trust	Does not fully address long-term deployment or evolving threat adaptation

Methodology

Datasets and Data Governance

- Two publicly available intrusion detection datasets are employed so that the methodological reproducibility can be guaranteed and the models can be subjected to the heterogeneous feature spaces and traffic distributions.
- UNSW NB15 contains labelled network flow data that is modern in attack categories and has mixed numeric and categorical attributes, and was designed as a successor-style dataset to resolve the shortcomings of previous benchmarks (Moustafa and Slay, 2015; UNSW Canberra Cyber).
- CICIDS2017 offers benign and attack traffic based on realistic user profiles, and modern attack conditions with large flow capabilities typically employed in intrusion detection studies and testing (Sharafaldin et al., 2018; Canadian Institute for Cybersecurity).
- The methodological protocol assumes that each dataset is a separate experimental domain.
- Training and evaluation of models is done in each dataset to prevent cross domain label drift and schema misalignment in features.
- This choice agrees with the reality that UNSW NB15 and CICIDS2017 vary significantly in definitions of features, capture setting, and classes priors (Moustafa and Slay, 2015; Sharafaldin et al., 2018).
- All the preprocessing statistics, encoders, and scaling parameters are trained on just the training partition of each dataset, and applied to the held-out test partition, to maintain the causal direction as intended at deployment time.
- This is an important governance action since leakage may swell offline performance and fail during deployment when there will be no future information (National Institute of Standards and Technology, 2025).

Methodology

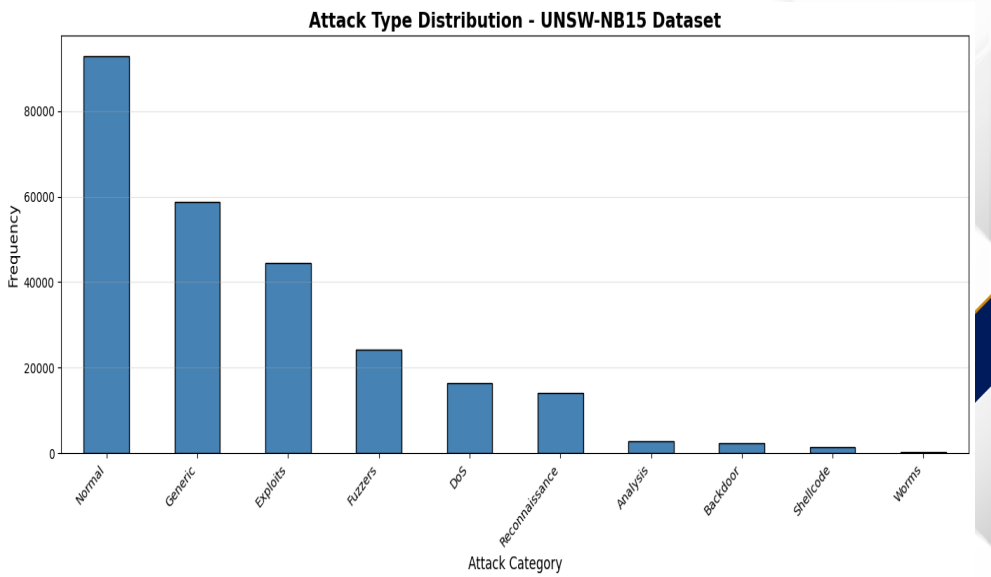
Preprocessing and Feature Engineering

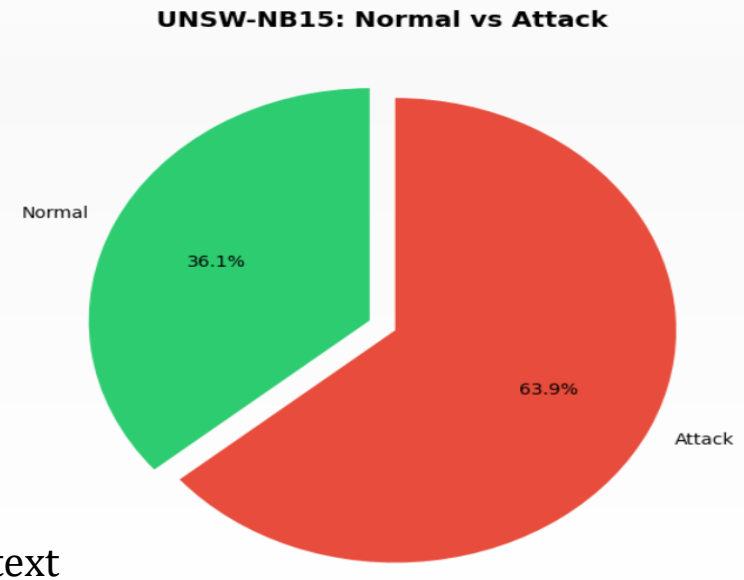
- ***Realistic deployment focus:*** preprocessing designed to avoid data leakage and reflect real-world conditions
- ***UNSW-NB15:***
  - Binary labels: Normal (0), Attack (1)
  - Removed label, attack-type, and identifier columns
  - Stratified train–test split before encoding
  - Missing values imputed using training data only (median for numerical, mode for categorical)
  - Categorical features label-encoded using training-fitted encoders
- ***CICIDS2017:***
  - Similar governance approach with larger scale and many numerical features
  - Handled missing and infinite values
  - Feature normalization based on training data parameters
  - Stratified sampling applied post split to manage memory constraints
- ***Class imbalance handling:***
  - Treated as a key methodological issue
  - Used class-weighted learning instead of accuracy-based evaluation
  - Oversampling (e.g., SMOTE) acknowledged, but weighting preferred for reliable network telemetry

Result and Discussion

Results Overview and Evaluation Design

- ***Unified SOC workflow:***
  - Ranked alert queue generated using three signals:
  - Tabular classifier output (XGBoost)
  - Temporal sequence model output (LSTM)
  - Incident grouping via density-based clustering
- ***Evaluation objectives:***
  - Model discrimination on held-out test data
  - Structure discovery in alert space
  - Operational impact via alert reduction while retaining true threats
- ***Data splits:***
  - UNSW-NB15: 80k training / 20k testing
  - CICIDS2017: sampled to 100k → 80k training / 20k testing
- ***Priority scoring mechanism:***
  - Priority score range: 0–10
  - Dynamic weighted fusion of signals:
    - XGBoost: 0.50
    - LSTM: 0.30
    - Clustering: 0.20
- Emphasis on classifier confidence, enhanced by temporal and incident density context





Result and Discussion

Dataset Characterization and Preprocessing Outcomes

‡ ***UNSW-NB15:***

- 257,673 total records (63.9% attacks, 36.1% normal)
- 10 attack categories, dominated by Generic and Exploits
- Represents a high-attack environment, suitable as a SOC stress test
- Encourages prioritization to suppress high-volume, low-risk alerts while retaining high-impact threats

‡ ***CICIDS2017:***

- Original size: ~2.83 million records; 100,000 sampled for training
- Strong class imbalance: 80.45% normal, 19.55% attack
- Reflects a benign-dominated environment, prone to alert fatigue from false positives

‡ ***Comparative insights:***

- UNSW-NB15: frequent malicious activity → higher base rate of actionable alerts
- CICIDS2017: diverse traffic → greater need for careful prioritization

‡ ***Protocol & port diversity:***

- UNSW-NB15: 133 protocols; limited port ranges
- CICIDS2017: wide protocol and port diversity, ports spanning near full 16-bit range
- Highlights differing network regimes and the importance of threshold tuning

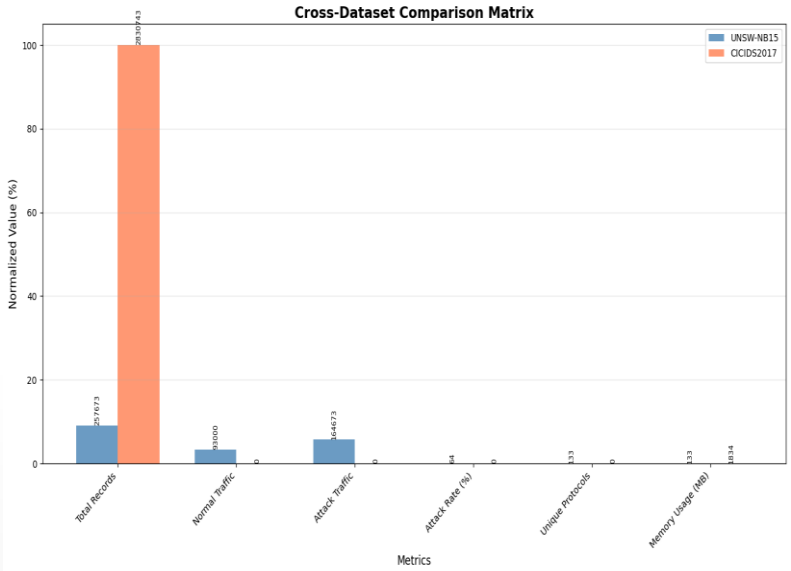


Figure: Cross dataset comparison matrix

Result and Discussion

XGBoost alert filtering results

- ***Imbalance-aware XGBoost model*** demonstrated strong discrimination on UNSW-NB15 test data
- ***Performance metrics:***
  - Accuracy: 0.94
  - F1-score: 0.95
  - ROC-AUC: 0.99
  - Indicates excellent attack–normal separation and well-ordered probability scores for prioritization
- ***Key influential features:***
  - sttl, sload, dload, sbytes, dbytes
  - Capture traffic intensity and packet-level behaviour
  - Align with known intrusion detection heuristics
- ***Operational relevance:***
  - High-priority alerts show near-unit attack probabilities
  - Confirms strong confidence at the top of the alert queue
  - Provides interpretable, SOC-aligned evidence for alert prioritization

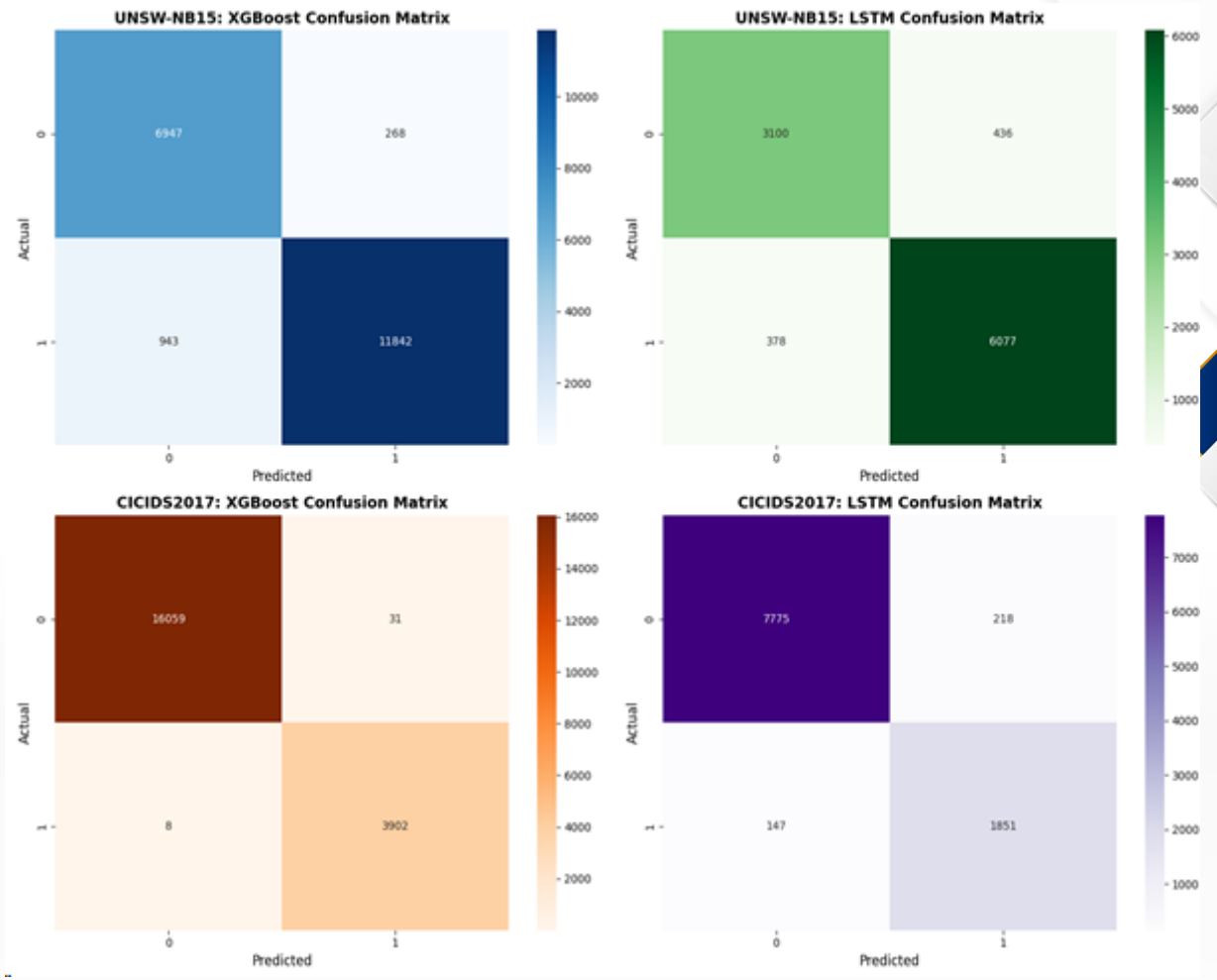


Figure: XG Boost Confusion Matrix

Result and Discussion

LSTM temporal modelling results

‡ ***Model design:***

- LSTM with 10-time-step sequences to capture short-term flow dependencies
- UNSW-NB15: ~50k training / 10k testing sequences (42 features)
- CICIDS2017: ~80k training / 20k testing sequences (78 features)

‡ ***Performance (UNSW-NB15):***

- Accuracy: 0.92
- F1-score: 0.94
- ROC-AUC: 0.98
- Slightly lower than XGBoost → static flow features dominate discrimination

‡ ***Operational insight:***

- Top-ranked alerts show near-unit LSTM probabilities
- Strong agreement with tabular model at highest priorities
- Reduces risk of single-model bias in alert ranking

‡ ***Role in CICIDS2017:***

- Adds separation for bursty and sequential behaviors
- Serves as a secondary, stabilizing signal in the prioritization framework

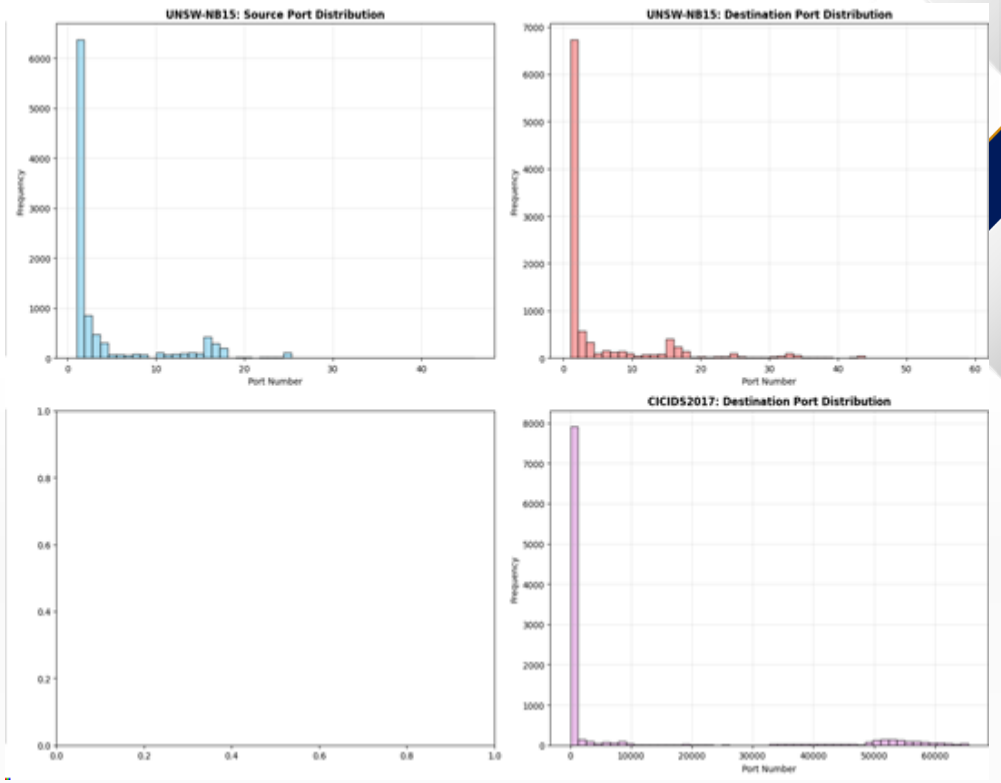


Figure: Sort and Port Distribution

Result and Discussion

Analyst Workflow Simulation

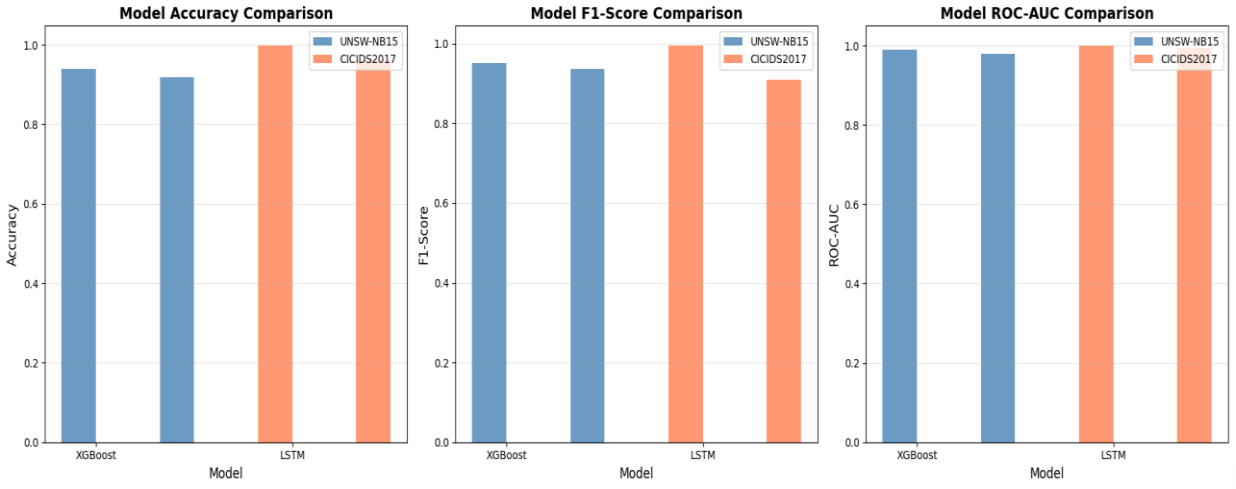


Figure: Model Accuracy, Score, and AUC Comparison

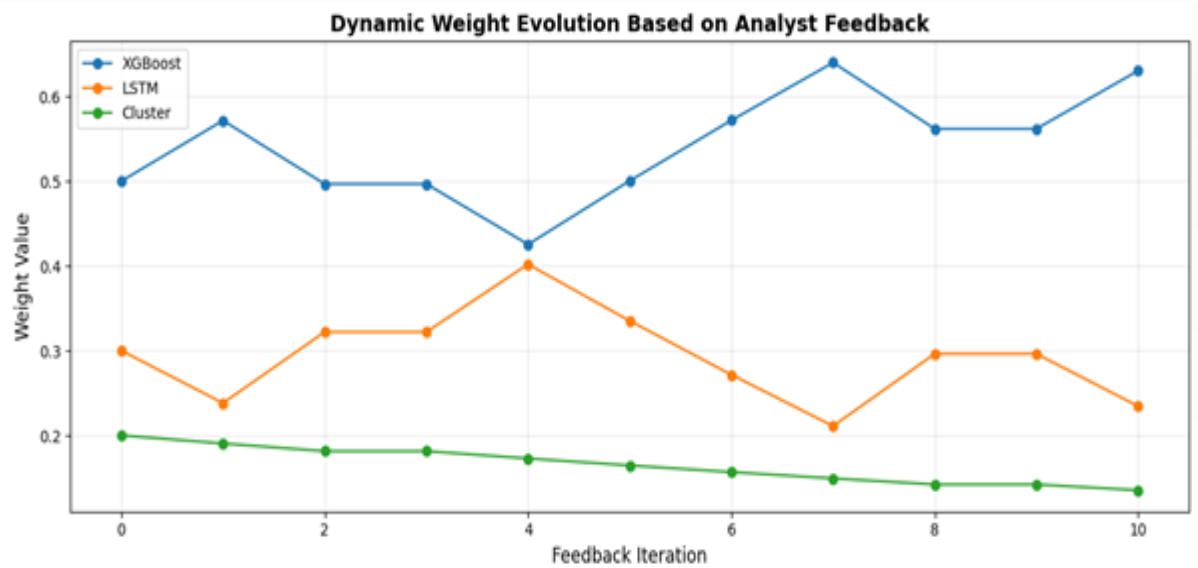


Figure: Dynamic weight evolution based on Analyst Feedback

Conclusion

- ***Core problem addressed:***
  - SOC alert fatigue due to limited analyst capacity, not lack of detections
- ***Proposed solution:***
  - Unified prioritisation architecture combining:
    - Tabular classifier
    - Temporal sequence model
    - Incident-level clustering
  - Produces a single, interpretable priority score for alert queues
- ***Key findings:***
  - Tabular model provides strongest discrimination and reliable ranking
  - Temporal model adds robustness via sequence context
  - Clustering introduces incident structure, reducing repeated low-value alerts
- ***Operational impact:***
  - Priority thresholding significantly reduces alert volume while retaining true threats
  - Thresholds are policy-dependent and dataset-specific
  - High-priority alerts are consistently supported by multiple signals
- ***Main takeaway:***
  - Effective SOC automation must prioritise ranking, aggregation, and interpretability, not accuracy alone

Appendix-I

References

- Moustafa, N. and Slay, J., 2015, November. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 military communications and information systems conference (MilCIS) (pp. 1-6). IEEE.
- UNSW Canberra Cyber. UNSW NB15 dataset technical description.
- Sharafaldin, I., Lashkari, A.H. and Ghorbani, A.A., 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp, 1(2018), pp.108-116.
- Canadian Institute for Cybersecurity. CICIDS2017 dataset documentation.
- McGill T et al. Alert fatigue in security operations centres. ACM. 2025.
- Kearney, P., Abdelsamea, M., Schmoor, X., Shah, F. and Vickers, I., 2023. Combating alert fatigue in the security operations centre. Available at SSRN 4633965.
- National Institute of Standards and Technology. Computer Security Incident Handling Guide SP 800-61 Revision 3. 2025.
- Chen, T., 2016. XGBoost: A Scalable Tree Boosting System. Cornell University.
- Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.
- Ester, M., Kriegel, H.P., Sander, J. and Xu, X., 1996, August. A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd (Vol. 96, No. 34, pp. 226-231).
- Lundberg, S.M. and Lee, S.I., 2017. A unified approach to interpreting model predictions. Advances in neural information processing systems, 30.
- Saito, T. and Rehmsmeier, M., 2015. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PloS one, 10(3), p.e0118432.
- He, H. and Garcia, E.A., 2009. Learning from imbalanced data. IEEE Transactions on knowledge and data engineering, 21(9), pp.1263-1284.
- Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, pp.321-357.
- Shittu, R., Healing, A., Ghanea-Hercock, R., Bloomfield, R. and Rajarajan, M., 2015. Intrusion alert prioritisation and attack detection using post-correlation analysis. Computers & security, 50, pp.1-15.
- Haas, S., 2020. Security monitoring and alert correlation for network intrusion detection (Doctoral dissertation, Staats-und Universitätsbibliothek Hamburg Carl von Ossietzky).
- McElwee, S. and Cannady, J., 2019, April. Cyber situation awareness with active learning for intrusion detection. In 2019 SoutheastCon (pp. 1-7). IEEE.
- Kim, Y., Dán, G. and Zhu, Q., 2024. Human-in-the-loop cyber intrusion detection using active learning. IEEE Transactions on Information Forensics and Security.
- Charmet, F., Tanuwidjaja, H.C., Ayoubi, S., Gimenez, P.F., Han, Y., Jmila, H., Blanc, G., Takahashi, T. and Zhang, Z., 2022. Explainable artificial intelligence for cybersecurity: a literature survey. Annals of Telecommunications, 77(11), pp.789-812.
- Leevy, J.L. and Khoshgoftaar, T.M., 2020. A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data. Journal of Big Data, 7(1), p.104.
- Kaushik, A., & Goel, V. (2021). Intrusion Detection using Machine Learning Techniques. International Journal of Science and Research (IJSR), 10(12), 695–698. https://doi.org/10.21275/sr211211230813
- Ma'aji, M.Y. and Aliyu, M.S., 2023. Models comparison based on intrusion detection using machine learning. SLU Journal of Science and Technology, pp.74-86.
- Malik, B., & Singh, N. (2022). Intrusion Detection using Machine Learning. International Journal of Science and Research (IJSR), 11(5), 283–286. https://doi.org/10.21275/mr22310165547
- Naseer, S., Saleem, Y., Khalid, S., Bashir, M. K., Han, J., Iqbal, M. M., & Han, K. (2018). Enhanced Network Anomaly Detection Based on Deep Neural Networks. IEEE Access, 6, 48231–48246. https://doi.org/10.1109/access.2018.2863036
- Ogino, T. (2015). Evaluation of Machine Learning Method for Intrusion Detection System on Jubatus. International Journal of Machine Learning and Computing, 5(2), 137–141. https://doi.org/10.7763/ijmlc.2015.v5.497
- Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., & Ghogho, M. (2016). Deep learning approach for Network Intrusion Detection in Software Defined Networking. 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM). https://doi.org/10.1109/wincom.2016.7777224
- Vikrant Sharma. (2025). Comparative Analysis of Machine Learning Models for Intrusion Detection Systems. Panamerican Mathematical Journal, 35(3s), 273–285. https://doi.org/10.52783/pmj.v35.i3s.3891

THANK YOU