National College of Ireland

Research Report

MINIMIZING FALSE POSITIVE IN SOC USING AI/ML

MSc in Cybersecurity

Kiran Sreekumar

Student ID: x23279851

School of Computing

National College of Ireland

Supervisor: Michael Prior

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Kiran Sreekumar		
Student ID:	x23279851		
Programme:	MSc in Cybersecurity	**Year:**	September 2024
Module:	MSc (Research) Practicum		
Supervisor:	Michael Prior		
Submission Due Date:	11th August 2025		
Project Title:	MINIMIZING FALSE POSITIVES IN SOC USING AI/ML		

Word Count: 4742 **Page Count:** 29

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Kiran Sreekumar

Date: 11th August 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

MINIMIZING FALSE POSITIVES IN SOC USING AI/ML

KIRAN SREEKUMAR

x23279851

Abstarct

This dissertation reviews the application of current artificial intelligence and machine learning (AI/ML) models in reducing false positives in Security Operations Centers (SOCs). The paper features a cybersecurity dataset as a real-world problem, and preprocessing was standardized through its implementation on a real-life cybersecurity dataset, utilizing two models: Random Forest and Gradient Boosting. The most commonly used metrics that will be used to test these models include accuracy, precision, recall, and F1-score. It is clear in the findings that both models were good; Gradient Boosting has a bit better true and false positive balance. The study accords with the existing literature, and it has an applied and scholarly direction of understanding how to improve alert sensitivity in the SOC settings. It also provides a list of shortcomings and suggests plans for improvement using powerful models, real-time processing, and explainable AI in order to aid the decision-making of the analysts.

Chapter 1: Introduction

1.1 Background and Context

The best-organized form of organizational cybersecurity in modern organisations is Security Operations Centres (SOCs), which aim at detecting and responding to threats in a real-time manner. Nevertheless, a sheer number of false positive alerts used to be supplied by detection systems implemented with the models formerly faced analysts with a sheer number of decryptions and compromised performance in that way. In reply, technologies such as Artificial Intelligence (AI) and Machine Learning (ML) have been considered as potentially useful in the improvement of the accuracy of alerts (Olateju *et al.* 2024).

1.2 Problem Statement

SOC security analysts have a very big challenge with regard to the large number of false positives that are generated by their automated threat detection mechanisms. These false warnings waste precious time and will result in alert fatigue and may even result in the obscuring of actual threats. Although there have been developments in the field of AI/ML, no one is quite sure how useful the current models can be toward eliminating false positives.

1.3 Research Aim and Objectives

Aim

This dissertation aims to evaluate the effectiveness of existing AI/ML models in minimizing false positives within Security Operations Centres (SOCs), using real-world cybersecurity datasets and standardized preprocessing techniques to identify the most suitable approaches for improving alert accuracy.

Objectives

- To review and analyze existing literature on the use of AI/ML techniques for reducing false positives in Security Operations Centres (SOCs).
- To preprocess and explore real-world cybersecurity datasets for model training and evaluation.
- To implement and evaluate the performance of selected AI/ML models in classifying alerts within SOC environments.
- To compare model performance using appropriate evaluation metrics and determine their effectiveness in minimizing false positives.

1.4 Research Questions

1. What AI/ML techniques have been explored in existing literature to address false positives in Security Operations Centres (SOCs)?

2. How can real-world cybersecurity datasets be preprocessed to effectively support the training and evaluation of AI/ML models?
3. How accurately can selected AI/ML models classify alerts within SOC environments when applied to real-world data?
4. Which evaluation metrics best reflect the effectiveness of AI/ML models in minimizing false positives, and which models perform most effectively based on these metrics?

1.5 Significance of the Study

This paper has practical and academic implications as it answers the urgent question of false positives in SOCs that inhibit rapid and successful response to threats. The conducting of real-world testing of the available AI/ML models using cybersecurity datasets allows the research to conclude their applicability and performance in the real world (Al-Quayed *et al.* 2024).

1.6 Scope and Limitations

This research will pay attention to comparing current AI/ML models relating to SOC environments regarding false positive avoidance based on freely available data in cybersecurity. The field of interest consists of data preparation, training, and testing. Weaknesses are that secondary data was used, live SOCs are not available, and the results are only valid to a limited extent outside of the tested data sets and models (Khayat *et al.* 2025).

1.7 Dissertation Structure

The dissertation has been laid out in six chapters. In chapter 1, the background of the research, the aim and objectives, and the scope of the research have been introduced. Chapter 2 involves a literature review of the concept of false positives and AI / ML in SOCs. Chapter 3 provides the methodology of the research. Chapter 4 explains the process of design and implementation. Chapter 5 is used to assess the performance of the models, and the last chapter is conditioned by the summary of the results. Chapter 6 addresses the findings as well as limitations, and future research directions.

Chapter 2: Literature Review

2.1 Introduction

The literature review examines the current body of literature concerning artificial intelligence and machine learning use in cybersecurity, specifically in Security Operations Centres (SOCs). It will form a baseline discussion on the technologies that are used to identify, characterize, and countermeasure cyber threats. The chapter discusses in a systematic way the key topics, including alert classification models, ongoing learning via analyst feedback, schema-based data integration, and model assessment in a real-world setting. This gives grounds to locate the proposed hybrid alert classification model in the bigger academic and practical picture, where the importance of more adequate and dynamic cybersecurity solutions is justified.

2.2 Machine Learning Approaches for Alert Classification in Security Operations Centres (SOCs)

Machine learning (ML) in Security Operations Centres (SOCs) has become one of the effective ways to improve threat detection and alert classification. Adjusting to the complexity and magnitude of contemporary cyber threats with conventional, rule-based systems has proven to be a challenge; the idea of deploying smart, adaptive methods is a solution. Ismail et al. (2022) introduced the light ML ensemble-based approach called the Weighted Score Selector (WSS) specifically in consideration of Wireless Sensor Networks. Along a similar vein, Yadulla et al. (2023) presented a hybrid machine learning (ML) based Intrusion Detection System (IDS) system that integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models..

2.3 Continuous Learning and Analyst Feedback Integration in Adaptive Cybersecurity Systems

Cybersecurity systems also need to adapt to the changing and increasingly frequent cyber threats via continuous learning and productive incorporation of feedback provided by cybersecurity analysts. In many cases, traditional intrusion detection systems may be hampered by alarm fatigue when there are too many false positives and reduced effectiveness in a real-time environment. To overcome this difficulty, Kim et al. (2024) introduced a human-in-the-loop detection framework, which improves situation awareness with the following three mechanisms: dynamically prioritizing alerts, investigation of alerts by humans, and sequential hypothesis testing with pruned Hidden Markov Models (HMMs). Oyinloye et al. (2025) handled the issue of imbalanced and noisy data in intrusion detection in the same manner, by proposing an improved Artificial Neural Network (ANN) model.

2.4 Standardized Data Models and Schema-Driven Integration for Enhanced Security Analytics

The growing complexity of cybersecurity ecosystems, especially the incorporation of IoT and Industry 5.0 technologies, necessitates the use of standardized data models and more formal schema-driven integration to enhance interoperability and threat detection capabilities. Cyber Threat Intelligence (CTI) plays a key role in enabling this structured data management. Alaeifar et al. (2024) prioritize the significance of CTI as the mechanism of facilitating the proactive mitigation of threats through the process of sharing structured information between different organizations. The lack of a standardized schema, such as the Open Cybersecurity Schema Framework(OCSF), renders the threat data integration procedure across numerous sources prone to errors and inefficient, which nullifies the gains associated with collaborative intelligence. On the other hand, Odeh and Abu Taleb (2023) regard intrusion detection on IoT systems based on deep learning and the relevance of structured analytics models.

2.5 Performance Evaluation and Generalizability of AI/ML Models in Real-World SOC Environments

In order to effectively implement and test AI/ML models in Security Operations Centres (SOCs), it is necessary to establish strict protocols that would establish a balance between model accuracy and adaptation, and generalization in the real world. In order to do so, they propose a new mechanism called Loss with a Decaying Factor (LDF), which increases the generalization of the pre-trained language models (PLMs) using heterogeneous data. The methodology changes the orientation of the model based on the events that take place in real-time, since the scenario is not irrelevant information, but recent.

2.6 Theoretical Framework

The theoretical approach applied by the research integrates the theory of machine learning and cyber threat detection with data-driven decision-making at SOCs. It is based on the concept of hybrid learning paradigms, supervised, unsupervised, and semi-supervised, which can jointly enable the adequate classification of the type of alerts. The model also leverages the continuous learning theory, which focuses on dynamic adaptation via feedback from analysts and threat intelligence.

2.7 Literature Gap

Although the past few years have witnessed considerable progress regarding the use of machine learning and deep learning methods to ensure cybersecurity, there are still some serious gaps in the literature. One is that numerous studies tend to concentrate on either the supervised or

unsupervised model and not much on the hybrid model that involves the use of supervised, unsupervised, and semi-supervised learning to complement each other in alert classification. Second, though there are studies that incorporate human feedback to some degree, there are no studied frameworks that constantly learn based on analyst interactions and conform to new threat intelligence on the fly.

2.8 Summary

The chapter has revisited the major literature that supports the design of smart, dynamic cybersecurity systems to be used in Security Operations Centres (SOCs). It started by reviewing several machine learning methods applicable to classifying alerts, and how hybrid models are effective in improving the accuracy of the detection. False positives were demonstrated to be reduced by the combination of continuous learning and analyst feedback, and this promotes model adaptability. The literature gap helped to conceptualize the opportunity to innovate in the area of hybrid modeling and adaptive feedback integration, whereas the theoretical framework ensured a conceptual basis upon which this research was guided.

(Summary table have been added in Appendix A)

Chapter 3: Research Methodology

3.1 Introduction

This chapter gives an overview of the research methodology that has been used to fulfill the goal of minimizing false positives in Security Operations Centers (SOCs) by applying Artificial Intelligence (AI) and Machine Learning (ML) methods (Alzahrani, 2021). The methodology is based on secondary quantitative data and was gathered from database which is publicly available online sources, namely Kaggle. The implementation and analysis work are all done with the Python programming language on the Jupyter Notebook platform, using the following well-known libraries in machine learning.

3.2 Research Design and Approach

3.2.1 Research Design

This research will use a descriptive research design since this design mainly aims at giving the correct description of characteristics, trends, and relations within a certain phenomenon. Within the framework of this study, the descriptive design allows for systematic analysis of the cybersecurity alerts, which will allow for determining the nature of cybersecurity false positive rates and their average frequencies in Security Operations Centers (SOCs) (Hakizimana and Ledezma Espino, 2025).

3.2.2 Research Approach

The study adopts a deductive research approach that starts with theoretical knowledge based on previous research techniques and interprets it through empirical research. Machine learning theories and models implemented in this research include Random Forest, SVM, and Neural Networks, which work on real-world datasets in order to test the efficiency in the reduction of false positives. Hypothesis-driven analysis, which compares assumptions based on their results that are measurable, is possible with this top-down reasoning (Alharbi *et al.* 2021).

3.3 Dataset Selection and Description

This work will employ two secondary data sources that are based on Kaggle, an established internet platform for data sharing methods and machine learning competitions. The datasets were selected so that they enable the assessment of a kind of situation with the threat of cybersecurity that exists nowadays and how it has evolved with time in such a way that it will be possible to rely on more competent models, which will help to decrease the number of false alarms in the Security Operations Center (SOC). It contains the characteristics of real-life SOC alert conditions, where it can be utilized in developing and testing the practical models (Azad *et al.* 2021).

The second one, the Cyber Security Attacks, presents Cyrillic data on several types of attacks, methods, techniques, and logic networks. These two different sets of data can be used to obtain a complete process of training and validation. An unseen testing dataset can also be included in the research at a later date, in order to determine how generalizable the model can be used in a new environment (Hiremath *et al.* 2023).

3.4 Data Preprocessing and Analysis Plan

Data preprocessing is very instrumental when it comes to reliable and accurate machine learning processes. In the topic of this research, two chosen datasets are initially pre-processed using a set of preprocessing pipelines so that the data can be of high quality and be compatible with ML algorithms. The numerical features are normalized or scaled, especially when models sensitive to feature magnitudes, such as Support Vector Machines, Neural Networks, are used (Subbiah *et al.* 2022).

To correct this, overrepresentation (balancing) of the dataset can be done using methods like Synthetic Minority Oversampling Technique (SMOTE) or random undersampling. The confusion matrices and class distribution help to understand label distribution, which would help in selecting performance metrics (Khayat *et al.* 2025).

3.5 Model Selection and Justification

The first way is to find the right machine learning models so that false-positive reduction can be effective in Security Operations Centers (SOCs). The choice of these models was made possible by the fact that they have been tested and proven to perform well in the laws of classification, particularly in the field of cybersecurity.

Random Forest is an algorithm of ensemble learning that creates many different decision trees and produces a more perceptive result by averaging the outputs.

Gradient Boosting Machines, XGBoost, and LightGBM are selected because of their high efficacy in prediction and efficiency.

3.6 Model Implementation Strategy

Python will be used on the basis of a high level of machine learning libraries support, simplicity of use, and a high level of community support. Python libraries like scikit-learn, XGBoost, LightGBM, TensorFlow, and Keras are used to support the development, training, and testing of models.

The training data is used to train each of the models, and the performance is tested on the validation data. Measurement metrics will involve the accuracy, precision, recall, F1-score, and the ROC-AUC, and of most interest will be the precision and recall since they are applicable in determining the false positives.

Hyperparameter tuning is employed to enhance the performance of the model, and this is done through methods like grid search or randomized search, depending on the cost of computation and the complexity of the model.

3.7 Design Specification

The system architecture of the proposed research will allow a modular, scalable, and efficient workflow of methods to reduce the number of false positives in Security Operations Centers (SOCs) through AI and machine learning methods. There are three core layers, namely, the layers of data acquisition, data processing and analysis, and model development and evaluation in architecture.

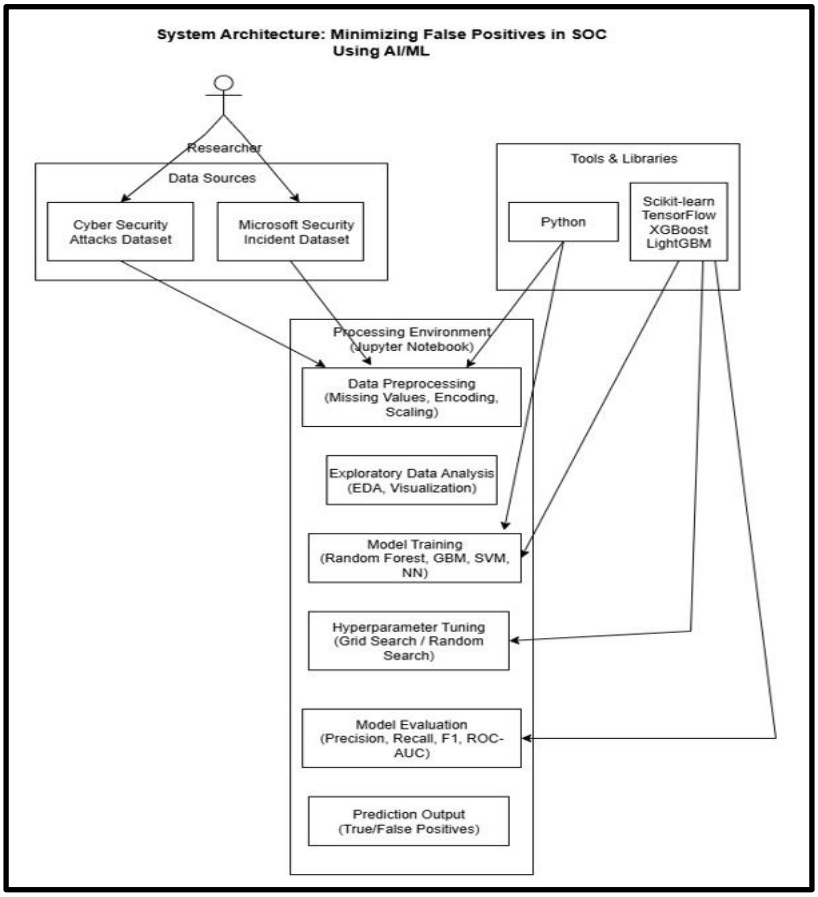


Figure 3.7.1: Diagram of System Architecture

(Source: Self-created)

The data is taken from two publicly available cybersecurity datasets available online on Kaggle, and they are loaded into the Jupyter Notebook environment, pre-processed, and explored there. Its main implementation is based on the applicational use of machine learning models, including Random Forest and Gradient Boosting (Aziz, 2023).

3.8 Limitations and Ethical Considerations

Although the research methodology that will be used in this research is structured and effective enough, there are certain limitations to it. Upending secondary datasets limits the control of the quality of the data, availability of features, and labeling accuracy. There is also the issue of the controlled environment that the model is put into, and it might not be a full-scale repeat of live Security Operations Centers (SOCs), so its generalizability might be compromised in the real world (Okoli *et al.* 2024).

3.9 Summary

In this chapter, the researchers have provided an in-depth description of the methodology they applied regarding the research conducted to eliminate false positives in Security Operations Centers (SOCs) by employing the AI/ML techniques. The deliberate research method consisted

of a descriptive design and a deductive approach; it was based on secondary quantitative data that was extracted by Kaggle. Preprocessing of data, exploratory analysis, selection of models, implementation, and assessment were all important in the Python-based Jupyter Notebook framework.

Chapter 4: Design and Implementation Specifications

4.1 Introduction

In this chapter, the design of AI/ML models to solve this problem in Security Operations Centres (SOCs) is explained. It contains such things as the architecture choices, preprocessed in what way, tools, and the products produced. The implementation will be consistent with the dissertation scope, which entails an assessment of the usefulness of the model based on actual cybersecurity data.

4.2 Design Specification

4.2.1 Framework Overview

The architecture used in this paper is a rigid data processing approach that involves data preparation, the derivation of features, model training, and evaluation.. The first steps were cleaning missing values and drawing high-cardinality fields, and the absence of data quality. Label encoding was then done on categorical features (Otoum *et al.* 2022). Random Forest and Gradient Boosting are the 2 AI/ML models that were chosen to be compared with each other since they have good results in terms of the ability to classify things. The model performance was utilized in order to optimize and tune each model to determine its ability to be used in reducing false positives in SOC alert classification tasks.

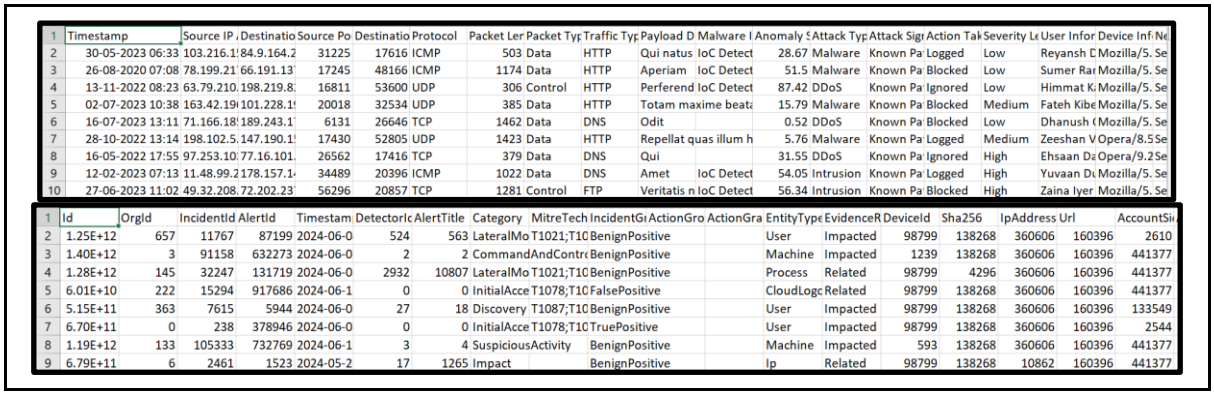
4.2.2 Requirements and Tools

The implementation of the design used a number of hardware and software tools that could assist with data processing, machine learning, and visualization. The local execution was performed on a standard personal computer with a minimum of 8 GB RAM and an Intel i5 processor. The reason Python was selected is due to the large library support it has to offer in data science and machine learning. The following core libraries were used: Pandas, Scikit-learn, Matplotlib, and Seaborn, to handle the data, preprocess, train, and validate the mathematical models, and visualize the data, respectively (Latif *et al.* 2025). Hyperparameter tuning was completed with the help of the RandomizedSearchCV utility of scikit-learn.

4.3 Dataset Preparation and Preprocessing

4.3.1 Dataset Overview

The data that is utilized in this investigation, which is cybersecurity_attacks.csv, is the actual use of alert records that mimic the traffic in the Security Operations Centre (SOC).



Timestamp	Source IP	Destinatio	Source Po	Destinatio	Protocol	Packet Ler	Packet Ty	Traffic Ty	Payload D	Malware I	Anomaly S	Attack Typ	Attack Sig	Action Tak	Severity Le	User Infor	Device Inf	Ne
30-05-2023 06:33	103.216.1	84.9.164.2	31225	17616	ICMP	503	Data	HTTP	Qui natus	IoC Detect	28.67	Malware	Known Pa	Logged	Low	Reyansh D	Mozilla/5.	Se
26-08-2020 07:08	78.199.21	66.191.13	17245	48166	ICMP	1174	Data	HTTP	Aperiam	IoC Detect	51.5	Malware	Known Pa	Blocked	Low	Sumer Rat	Mozilla/5.	Se
13-11-2022 08:23	63.79.210.	198.219.8	16811	53600	UDP	306	Control	HTTP	Perferend	IoC Detect	87.42	DDoS	Known Pa	Ignored	Low	Himmat Ki	Mozilla/5.	Se
02-07-2023 10:38	163.42.19	101.228.1	20018	32534	UDP	385	Data	HTTP	Totam maxime beata		15.79	Malware	Known Pa	Blocked	Medium	Fateh Kibe	Mozilla/5.	Se
16-07-2023 13:11	71.166.18	189.243.1	6131	26646	TCP	1462	Data	DNS	Odit		0.52	DDoS	Known Pa	Blocked	Low	Dhanush (	Mozilla/5.	Se
28-10-2022 13:14	198.102.5.	147.190.1	17430	52805	UDP	1423	Data	HTTP	Repellat quas illum h		5.76	Malware	Known Pa	Logged	Medium	Zeeshan V	Opera/8.5	Se
16-05-2022 17:55	97.253.10	77.16.101.	26562	17416	TCP	379	Data	DNS	Qui		31.55	DDoS	Known Pa	Ignored	High	Ehsaan Da	Opera/9.2	Se
12-02-2023 07:13	11.48.99.2	178.157.1	34489	20396	ICMP	1022	Data	DNS	Amet	IoC Detect	54.05	Intrusion	Known Pa	Logged	High	Yuvaan Di	Mozilla/5.	Se
27-06-2023 11:02	49.32.208.	72.202.23	56296	20857	TCP	1281	Control	FTP	Veritatis n	IoC Detect	56.34	Intrusion	Known Pa	Blocked	High	Zaina Iyer	Mozilla/5.	Se

Id	OrgId	IncidentId	AlertId	Timestam	DetectorIc	AlertTitle	Category	MitreTech	IncidentGr	ActionGro	ActionGra	EntityType	EvidenceR	DeviceId	Sha256	IpAddress	Url	AccountSi
1.25E+12	657	11767	87199	2024-06-0	524	563	LateralMo	T1021;T10	BenignPositive			User	Impacted	98799	138268	360606	160396	2610
1.40E+12	3	91158	632273	2024-06-0	2	2	CommandAndContr	BenignPositive				Machine	Impacted	1239	138268	360606	160396	441377
1.28E+12	145	32247	131719	2024-06-0	2932	10807	LateralMo	T1021;T10	BenignPositive			Process	Related	98799	4296	360606	160396	441377
6.01E+10	222	15294	917686	2024-06-1	0	0	InitialAcce	T1078;T10	FalsePositive			CloudLogo	Related	98799	138268	360606	160396	441377
5.15E+11	363	7615	5944	2024-06-0	27	18	Discovery	T1087;T10	BenignPositive			User	Impacted	98799	138268	360606	160396	133549
6.70E+11	0	238	378946	2024-06-0	0	0	InitialAcce	T1078;T10	TruePositive			User	Impacted	98799	138268	360606	160396	2544
1.19E+12	133	105333	732769	2024-06-1	3	4	SuspiciousActivity	BenignPositive				Machine	Impacted	593	138268	360606	160396	441377
6.79E+11	6	2461	1523	2024-05-2	17	1265	Impact	BenignPositive				Ip	Related	98799	138268	10862	160396	441377

Figure 4.3.1.1: Overview of Datasets

(Source: Extracted from Excel)

It has 22 attributes, some of which include protocol information, IP addresses, ports, characteristics of packets, and the details of the attack. These characteristics give a full-fledged training ground to AI/ML models in order to make such classifications and decrease false positives.

4.3.2 Handling Missing and High Cardinality Data

The early exploration showed that there are considerable missing values in the data and that some columns (damaged Payload Data, information about the user, and geo-location data) have a high cardinality. To enhance the integrity of the data and computational efficiency, these columns were dropped as they were not important to generate models, or they were too sparse (Saied *et al.* 2023).

4.3.3 Encoding and Splitting

Label encoding was applied to the categorical variables to convert them into a numeric format, which made them compatible with machine learning algorithms. The coding of the target variable (various types of attack) was done as well.

```python
# Identify categorical columns
categorical_cols = cyber_df.select_dtypes(include='object').columns.tolist()

# Define target column (change this if needed)
target_col = 'Attack Type'

# Remove target from features
categorical_cols.remove(target_col)

# Encode categorical features
label_encoders = {}
for col in categorical_cols:
    le = LabelEncoder()
    cyber_df[col] = le.fit_transform(cyber_df[col])
    label_encoders[col] = le

# Encode target
target_encoder = LabelEncoder()
cyber_df[target_col] = target_encoder.fit_transform(cyber_df[target_col])
```

```python
# Separate features and target
X = cyber_df.drop(columns=[target_col])
y = cyber_df[target_col]

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

Figure 4.3.3.1: Encode Categorical Variable and Split the Dataset

(Source: Extracted from Jupyter Notebook)

After the encoding process, the data were divided into features (X) and target labels (y). To test the generalizability of the models, a train-test split of 80/20 was used. Such an organized preprocessing flow made the data clean, consistent, and good for developing and evaluating supervised learning models.

4.4 Model Development and Tuning

4.4.1 Random Forest Classifier

One of the models adopted was the Random Forest Classifier, which is used to classify cybersecurity attacks, as one of the aims is to determine the effectiveness of the model in reducing false positives in SOCs. It was selected because it is robust and can work with high-dimensional data (Kumar *et al.* 2025).

```
Fitting 3 folds for each of 20 candidates, totalling 60 fits

RandomizedSearchCV(cv=3, estimator=RandomForestClassifier(random_state=42),
                   n_iter=20, n_jobs=-1,
                   param_distributions={'bootstrap': [True, False],
                                        'max_depth': [10, 20, 30, None],
                                        'min_samples_leaf': [1, 2, 4],
                                        'min_samples_split': [2, 5, 10],
                                        'n_estimators': [100, 200, 300]},
                   random_state=42, verbose=1)
```

Figure 4.4.1.1: Hyperparameter Tuning for Random Forest Model

(Source: Extracted from Jupyter Notebook)

Hyperparameter tuning was carried out, and parameters optimized included the number of estimators, maximum depth, and minimum sample split using RandomizedSearchCV. The model trained correctly with an accuracy of about 83% and good precision and recall of the dominating class.

```
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.34      0.32      0.33       684
           1       0.34      0.36      0.35       667
           2       0.93      0.93      0.93      6649

    accuracy                           0.83      8000
   macro avg       0.54      0.54      0.54      8000
weighted avg       0.83      0.83      0.83      8000

Confusion Matrix:
[[ 219  233  232]
 [ 194  238  235]
 [ 230  237 6182]]
Accuracy Score:
0.829875
```

Figure 4.4.1.2: Classification Report and Confusion Matrix for Random Forest Model

(Source: Extracted from Jupyter Notebook)

It performed fairly on minority classes that generally reflect false positive cases. This result shows how difficult it is to find a balanced classification in real-world datasets and the necessity to focus on specific model tuning in cybersecurity applications.

4.4.2 Gradient Boosting Classifier

The Gradient Boosting Classifier choice was guided by its success in bringing out robust predictive results using sequential-based ensemble learning.

```
Fitting 3 folds for each of 20 candidates, totalling 60 fits

RandomizedSearchCV(cv=3, estimator=GradientBoostingClassifier(random_state=42),
                   n_iter=20, n_jobs=-1,
                   param_distributions={'learning_rate': [0.01, 0.05, 0.1],
                                        'max_depth': [3, 5, 7],
                                        'min_samples_leaf': [1, 2, 4],
                                        'min_samples_split': [2, 5, 10],
                                        'n_estimators': [100, 200, 300],
                                        'subsample': [0.8, 1.0]},
                   random_state=42, verbose=1)
```

Figure 4.4.2.1: Hyperparameter Tuning for Gradient Boosting Model

(Source: Extracted from Jupyter Notebook)

Optimization of hyperparameters, including learning rate, number of estimators, maximum depth, and subsampling rate, was achieved through the use of the RandomizedSearchCV algorithm. The tuned model had a similar level of accuracy of about 83%, with higher recall and F1-score on the majority class than the Random Forest.

```
Gradient Boosting Classification Report:
              precision    recall  f1-score   support

           0       0.34      0.32      0.33       684
           1       0.34      0.32      0.33       667
           2       0.92      0.93      0.93      6649

    accuracy                           0.83      8000
   macro avg       0.54      0.53      0.53      8000
weighted avg       0.83      0.83      0.83      8000

Confusion Matrix:
[[ 221  205  258]
 [ 204  215  248]
 [ 228  205 6216]]
Accuracy Score:
0.8315
```

Figure 4.4.2.1: Classification Report for Gradient Boosting Model

(Source: Extracted from Jupyter Notebook)

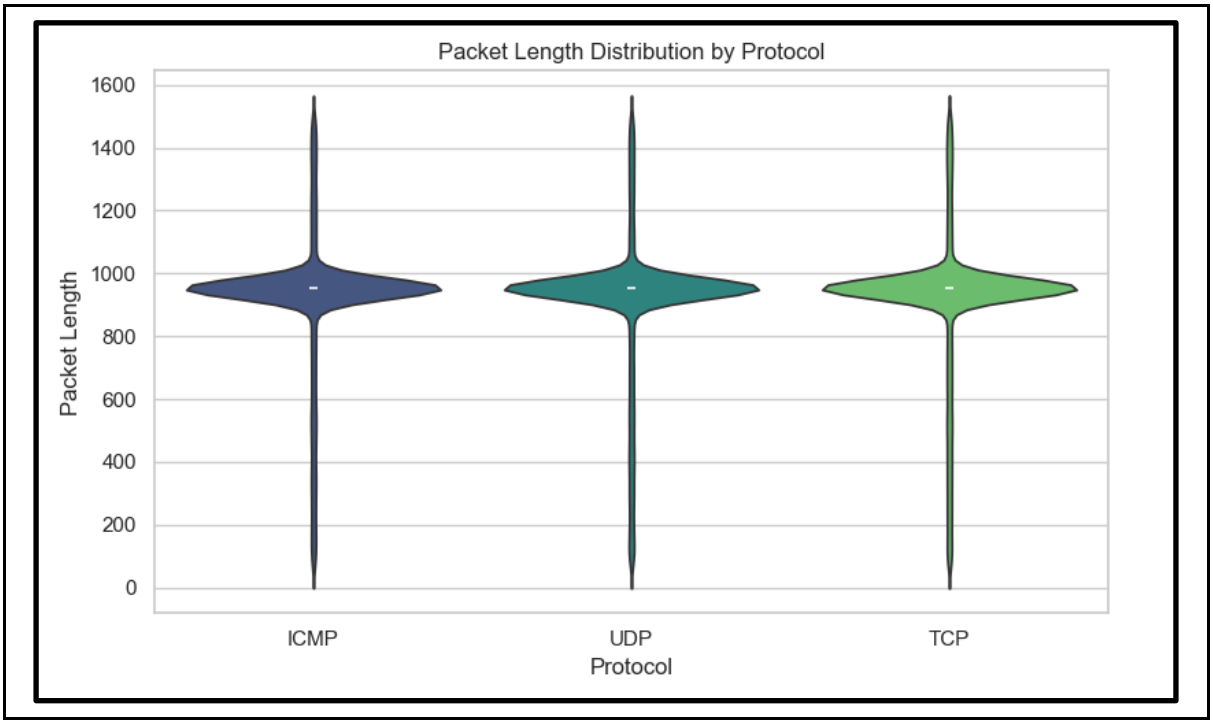
It showed similar performance on minority classes, which means that, along with boosting, there is always a problem of class imbalance. The incremental training method used by the model also made it noise-sensitive, where the selection of features and preprocessing is paramount. On the whole, Gradient Boosting proved to be a competitive model in the domain of SOC alert classification.

4.4.3 Evaluation Metrics Used

The evaluation of Model performance was evaluated on the basis of accuracy, precision, recall, and F1-score. These measures provided the overall outlook on the success of these models to establish the genuine threats, rather than false positives. The study used weighted averages to accommodate the presence of imbalance in classes (Prinzi *et al.* 2024).

4.5 Output Summary and Visualizations

The implementation yielded an overall assessment of Random Forest and Gradient Boosting models with the help of classification reports, confusion matrices, and visuals. Among the outputs, it was also possible to find the comparative bar plots of the accuracy, precision, recall, and F1-score that showed quite good results for both models demonstrated on the dominant class, but poor results on the minority classes, which identified the area that was more likely to have false positives.



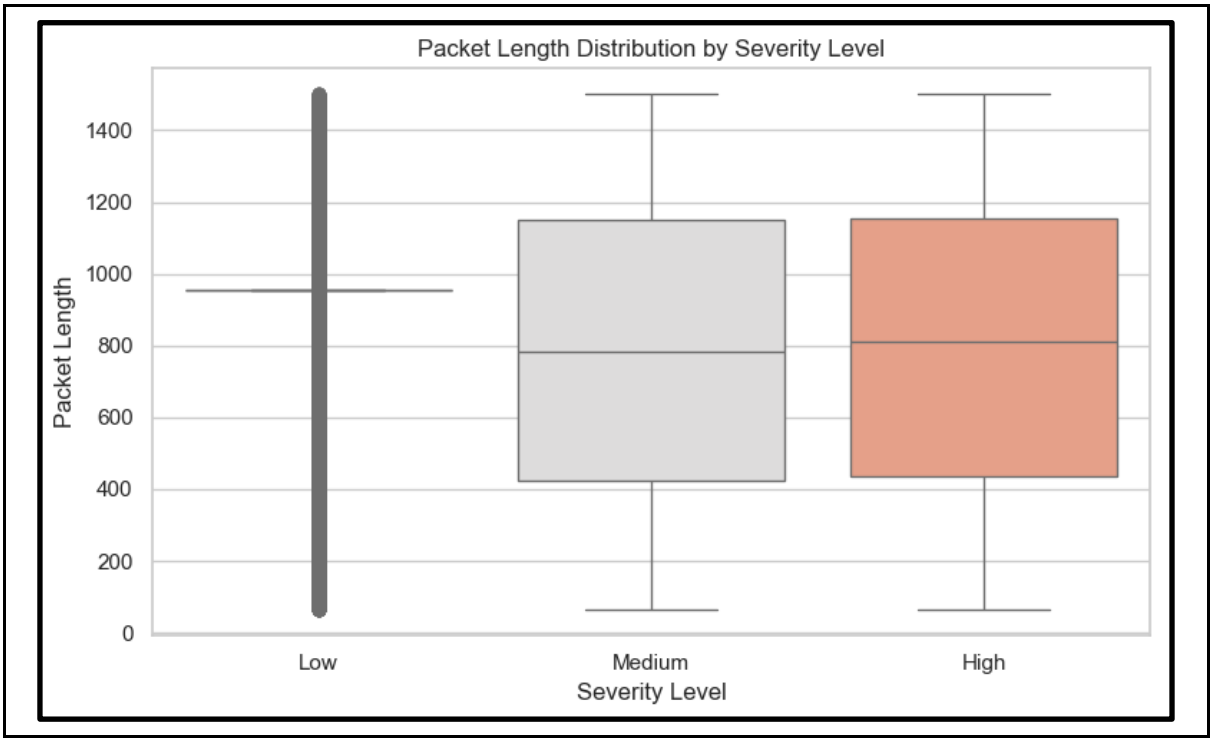


Figure 4.5.1: Violin and Box Plot

(Source: Extracted from Jupyter Notebook)

Violin and box plots gave an insight into the packet length distributions between protocols, as well as what level of severity is represented, thus helping in the interpretation of the features.

4.6 Summary

This chapter has elaborated on the development and deployment of AI /ML models to minimize false attributions in SOCs. It also described the structure, resources, preprocessing, model training, and testing. The results obtained provided quantitative evidence of the model's success in directly addressing the dissertation's aims, objectives, and research questions through planned experimentation.

Chapter 5: Evaluation

5.1 Introduction

The chapter provides a well-developed assessment of the AI/ML models created to keep the false positives in Security Operations Centres (SOCs) at a minimum level. It evaluates the performance of the models critically based on the standardised evaluations and visualisations, and attributes them to the research questions and objectives to infer their practical as well as scholarly implications in categorizing cybersecurity alerts.

5.2 Performance Comparison of Models

This section comments and compares the outcomes of two classifiers, Random Forest and Gradient Boosting, used to predict the cybersecurity alert dataset. To establish the efficacy of the two models in terms of averting false positives in SOCs, the models were evaluated on their performance using relevant measures of effectiveness such as accuracy, precision, recall, and F1-score.

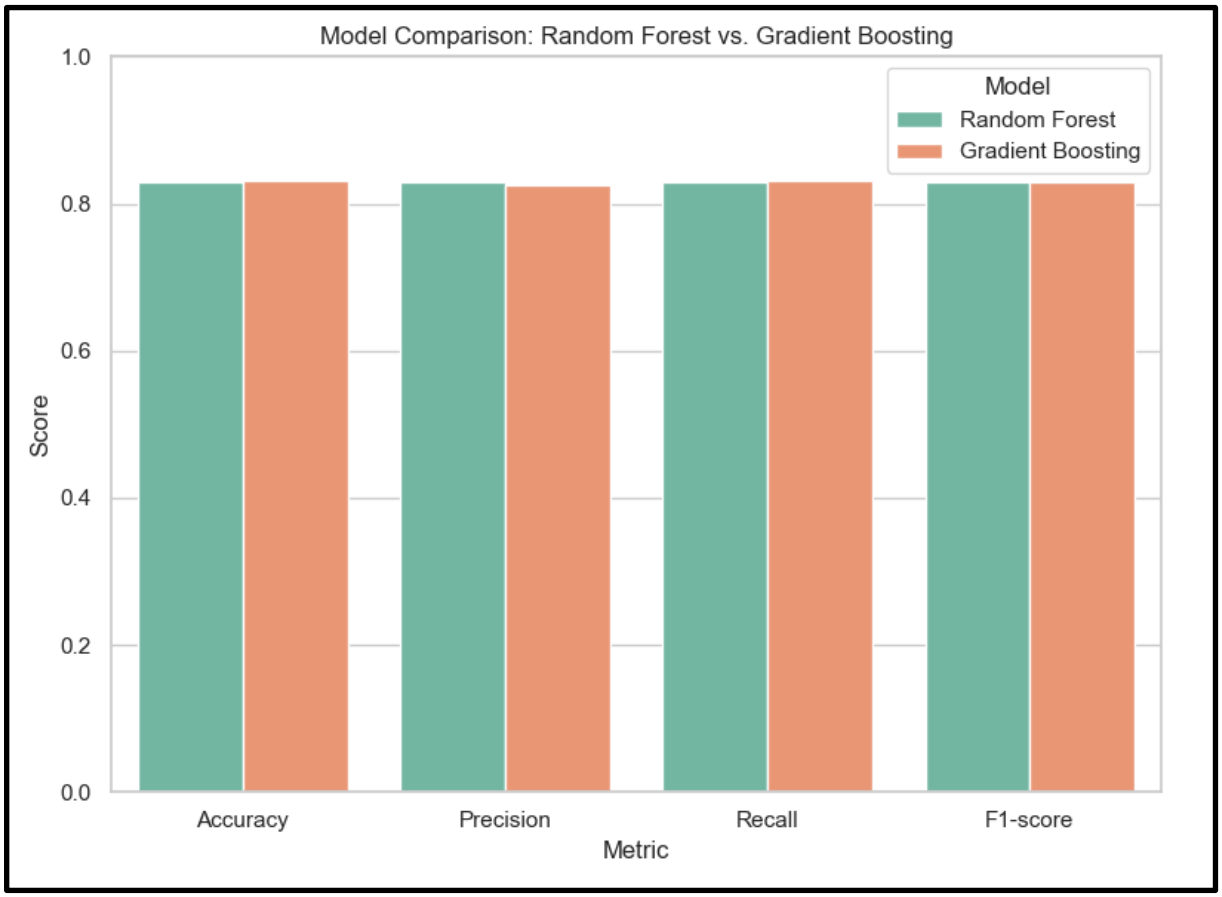


Figure 5.2.1: Model Comparison Chart

(Source: Extracted from Jupyter Notebook)

The Gradient Boosting model has performed really well relative to the Random Forest, not only in accuracy but also in the analysis of the majority alert variable. The accuracy of the two models in minority classes was moderate, and this is why it can be comprehended that there is a likelihood that the least frequent or unclear categories of alerts are confused.

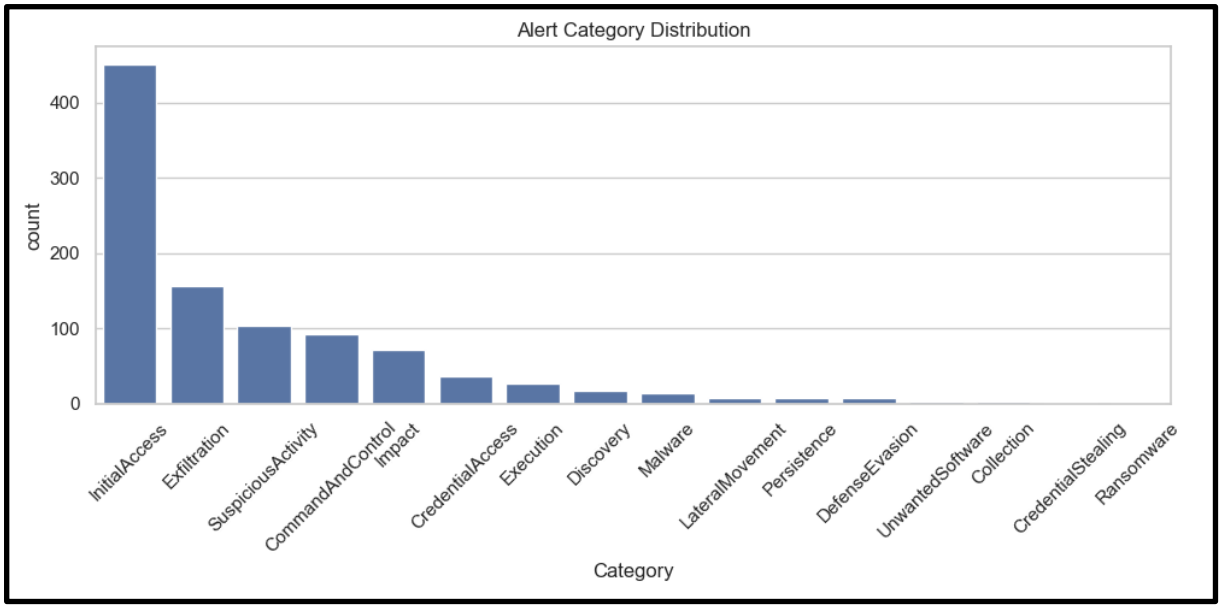


Figure 5.3.1: Alert Category Distribution

(Source: Extracted from Jupyter Notebook)

The findings increase the value of taking into consideration the overall accuracy and the performance of each class, particularly in the context of SOCs, where the multiplication of errors might result in occupational inefficiencies.

5.3 Discussion on False Positives Reduction

The outcomes of the assessments prove that it is possible to significantly reduce the occurrence of false positives in SOC settings with the application of AI/ML. In particular, the Gradient Boosting model presented high discriminative power, especially in legitimate alerts, and, therefore, had fewer chances of a false classification. Although both models encountered issues with minority classes, their overall results of prediction suggest their effectiveness in real-world applications (Bold *et al.* 2022).

5.4 Comparison with Literature

This dissertation has similar findings to the other literature that focuses on the functionality of ensemble approaches in lowering the false positives in SOCs. Research articles considered in the second chapter emphasized the shortcomings of the classic rule-based models and the benefits of the AI/ML-based ones in terms of enhancing alert accuracy. Gradient Boosting scored highly in this study, and this reinstates the same findings in former literature about the performance of Gradient Boosting in imbalanced data sets (Azam *et al.* 2023).

5.5 Practical and Academic Implications

The work has a practical implication since it is able to find effective AI/ML models that could minimize false positives, which would improve the efficiency of SOC and the productivity of

the analysts. It adds academic value in the form of empirical support to the idea of using ensemble learning methods (Sowmya and Anita, 2023).

5.6 Summary

This chapter compares and contrasts AI/ML models on the basis of standardized measures to assess the usefulness of AI/ML models in limiting false positives within SOCs. All of this was relevant to various research goals and provided both academic and practical contributions and answers to research questions, and also pointed the researchers to further study the problem under consideration, as well as assisted in maintaining optimal cybersecurity standards.

Chapter 6: Conclusions and Discussion

6.1 Summary of Findings

The research topic of this dissertation was the exploration of the success of AI/ML models to reduce false positives in Security Operations Centres (SOCs). By applying and testing the Random Forest and Gradient Boosting charts on real-life cybersecurity data, it was discovered that the specified classes demonstrate very high overall accuracy, with a slightly better performance of the Gradient Boosting model in the F1-score and precision. But, in both models, lower-performance levels were observed in correctly indicating minority alert classes, as the classes where false positives are most probable. Other data preprocessing procedures were essential in boosting the model, such as dealing with high cardinality and data encoding (Ozkan-Okay *et al.* 2024).

6.2 Validity and Generalisability of Results

The findings of the present study can be regarded as accurate because of using actual cybersecurity data, following preprocessing procedures, and testing them on the predefined performance indicators with cross-validation. The findings could be limited by dataset-specific features, which include class imbalance and domain-specific patterns. Although the models showed great performance on the test data, the results might be different in other SOC environments in the context of different threat environments.

6.3 Implications of the Research

This study gives an insight into the potential practical applications of AI/ML models to improve the efficiency of SOCs due to the false positives that frequently affect the process of dealing with incidents(Voutsas *et al.* 2024). The results advocate the incorporation of well-fitted classifiers, e.g., Gradient Boosting and Random Forest, into workflows of cybersecurity to enhance the accuracy of alerts. To practitioners, this means less analyst burnout and a proper ranking of threats.

6.4 Recommendations for Future Work

Research on uniting deep learning and ensemble learning is a topic of the future, as it promises to drive better classification accuracy and minimize false positives within SOCs. This can be enhanced through the incorporation of real-time data streams and more diverse and wider datasets that might facilitate better generalisability.

References

Alaeifar, P., Pal, S., Jadidi, Z., Hussain, M. and Foo, E., 2024. Current approaches and future directions for cyber threat intelligence sharing: A survey. Journal of Information Security and Applications, 83, p.103786.

Alharbi, A., Seh, A.H., Alosaimi, W., Alyami, H., Agrawal, A., Kumar, R., and Khan, R.A., 2021. Analyzing the impact of cybersecurity-related attributes for intrusion detection systems. Sustainability, 13(22), p.12337.

Al-Quayed, F., Ahmad, Z., and Humayun, M., 2024. A situation-based predictive approach for cybersecurity intrusion detection and prevention using machine learning and deep learning algorithms in wireless sensor networks of Industry 4.0. IEEE Access, 12, pp.34800-34819.

Alzahrani, L., 2021. Statistical analysis of cybersecurity awareness issues in higher education institutes. International Journal of Advanced Computer Science and Applications, 12(11), pp.630-637.

Azad, S., Naqvi, S.S., Sabrina, F., Sohail, S. and Thakur, S., 2021, December. IoT cybersecurity: On the use of machine learning approaches for unbalanced datasets. In 2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE) (pp. 1-6). IEEE.

Azam, Z., Islam, M.M. and Huda, M.N., 2023. Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision trees.IEEEe Access, 11, pp.80348-80391.

Aziz, M.N., 2023. Finding Patterns of Cyber-Attacks and Creating A Detection Model to Detect Cyber-Attacks Using Machine Learning. Journal of Artificial Intelligence, Machine Learning and Neural Network, 3(01), pp.8-24.

Bold, R., Al-Khateeb, H. and Ersotelos, N., 2022. Reducing false negatives in ransomware detection: a critical evaluation of machine learning algorithms. Applied Sciences, 12(24), p.12941.

Hakizimana, G. and Ledezma Espino, A., 2025. Nomological Deductive Reasoning for Trustworthy, Human-Readable, and Actionable AI Outputs. Algorithms, 18(6), p.306.

Hiremath, S., Shetty, E., Prakash, A.J., Sahoo, S.P., Patro, K.K., Rajesh, K.N. and Pławiak, P., 2023. A new approach to data analysis using machine learning for cybersecurity. Big Data and Cognitive Computing, 7(4), p.176.

Ismail, S., El Mrabet, Z. and Reza, H., 2022. An ensemble-based machine learning approach for cyber-attacks detection in wireless sensor networks. Applied Sciences, 13(1), p.30.

Jilcha, L.A., Kim, D.H. and Kwak, J., 2025. Temporal Decay Loss for Adaptive Log Anomaly Detection in Cloud Environments. Sensors, 25(9), p.2649.

Khayat, M., Barka, E., Serhani, M.A., Sallabi, F., Shuaib, K. and Khater, H.M., 2025. Empowering Security Operation Center with Artificial Intelligence and Machine Learning–A Systematic Literature Review. IEEE Access.

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.

Kumar, C.L., Betam, S., Pustokhin, D., Laxmi Lydia, E., Bala, K., Aluvalu, R., and Panigrahi, B.S., 2025. Metaparameter optimized hybrid deep learning model for next generation cybersecurity in a software-defined networking environment. Scientific Reports, 15(1), p.14166.

Latif, N., Ma, W. and Ahmad, H.B., 2025. Advancements in securing federated learning with IDS: A comprehensive review of neural networks and feature engineering techniques for malicious client detection. Artificial Intelligence Review, 58(3), p.91.

Odeh, A. and Abu Taleb, A., 2023. Ensemble-based deep learning models for enhancing IoT intrusion detection. Applied Sciences, 13(21), p.11985.

Okoli, U.I., Obi, O.C., Adewusi, A.O. and Abrahams, T.O., 2024. Machine learning in cybersecurity: A review of threat detection and defense mechanisms. World Journal of Advanced Research and Reviews, 21(1), pp.2286-2295.

Olateju, O., Okon, S.U., Igwenagu, U., Salami, A.A., Oladoyinbo, T.O. and Olaniyi, O.O., 2024. Combating the challenges of false positives in AI-driven anomaly detection systems and enhancing data security in the cloud. Available at SSRN 4859958.

Otoum, Y., Liu, D. and Nayak, A., 2022. DL-IDS: a deep learning–based intrusion detection framework for securing IoT. Transactions on Emerging Telecommunications Technologies, 33(3), p.e3803.

Oyinloye, T.S., Arowolo, M.O. and Prasad, R., 2025. Enhancing cyber threat detection with an improved artificial neural network model. Data Science and Management, 8(1), pp.107-115.

Ozkan-Okay, M., Akin, E., Aslan, Ö., Kosunalp, S., Iliev, T., Stoyanov, I. and Beloev, I., 2024. A comprehensive survey: Evaluating the efficiency of artificial intelligence and machine learning techniques on cybersecurity solutions. IEE Access, 12, pp.12229-12256.

Prinzi, F., Insalaco, M., Orlando, A., Gaglio, S. and Vitabile, S., 2024. A YOLO-based model for breast cancer detection in mammograms. Cognitive Computation, 16(1), pp.107-120.

Saied, M., Guirguis, S. and Madbouly, M., 2023. A comparative study of using boosting-based machine learning algorithms for IoT network intrusion detection. International Journal of Computational Intelligence Systems, 16(1), p.177.

Sowmya, T. and Anita, E.M., 2023. A comprehensive review of AI AI-based intrusion detection system. Measurement: Sensors, 28, p.100827.

Subbiah, S., Anbananthen, K.S.M., Thangaraj, S., Kannan, S. and Chelliah, D., 2022. Intrusion detection technique in wireless sensor network using grid search random forest with Boruta feature selection algorithm. Journal of Communications and Networks, 24(2), pp.264-273.

Tran, M.Q., Elsisi, M., Liu, M.K., Vu, V.Q., Mahmoud, K., Darwish, M.M., Abdelaziz, A.Y. and Lehtonen, M., 2022. Reliable deep learning and IoT-based monitoring system for secure computer numerical control machines against cyberattacks with experimental verification. IEEE Access, 10, pp.23186-23197.

Voutsas, F., Violos, J. and Leivadeas, A., 2024. Mitigating alert fatigue in cloud monitoring systems: A machine learning perspective. Computer Networks, 250, p.110543.

Yadulla, A.R., Kasula, V.K., Yenugula, M. and Konda, B., 2023. Enhancing Cybersecurity with AI: Implementing a Deep Learning-Based Intrusion Detection System Using Convolutional Neural Networks. European Journal of Advances in Engineering and Technology, 10(12), pp.89-98.

Bibliography

Prasath, S., Sethi, K., Mohanty, D., Bera, P., and Samantaray, S.R., 2022. Analysis of continual learning models for intrusion detection systems. *IEEE Access*, *10*, pp.121444-121464.

Schlette, D., Böhm, F., Caselli, M. and Pernul, G., 2021. Measuring and visualizing cyber threat intelligence quality. *International Journal of Information Security*, *20*(1), pp.21-38.

Hazman, C., Guezzaz, A., Benkirane, S. and Azrour, M., 2023. lIDS-SIoEL: intrusion detection framework for IoT-based smart environments security using ensemble learning. *Cluster Computing*, *26*(6), pp.4069-4083.

Thawait, N.K., 2024. Machine learning in cybersecurity: Applications, challenges, and future directions. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol*, *10*(3), pp.16-27.

Nwachukwu, C., Durodola-Tunde, K. and Akwiwu-Uzoma, C., 2024. AI-driven anomaly detection in cloud computing environments. *International Journal of Science and Research Archive*, *13*(2), pp.692-710.

Appendices

Appendix A: Summary Table

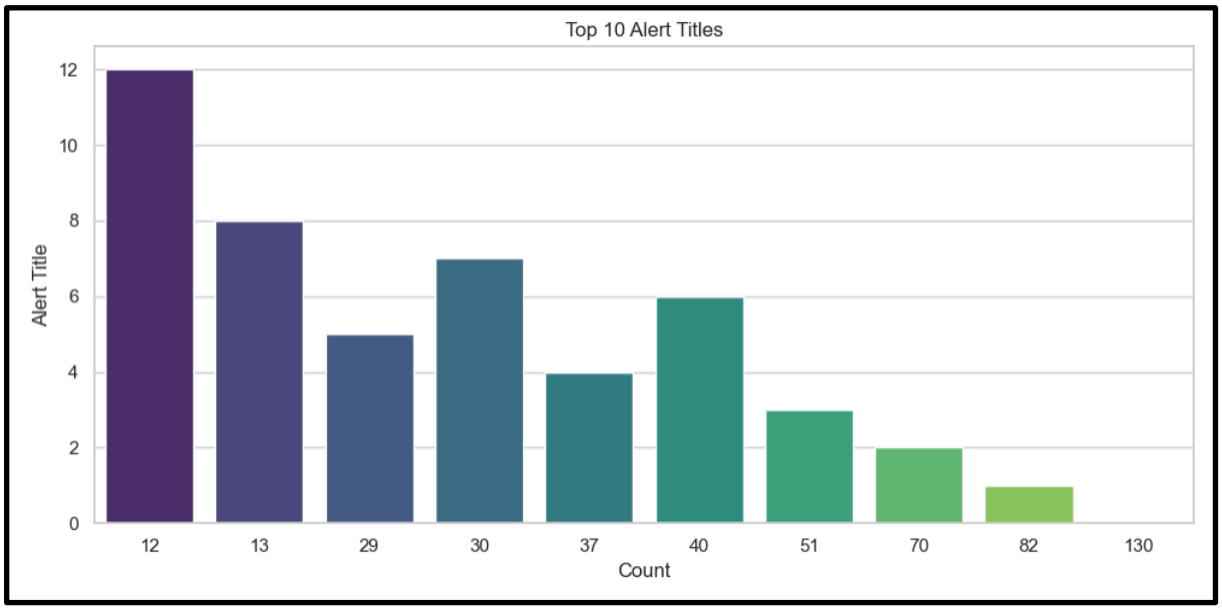
In-text citation	Title/Topic Summary	Strengths	Weaknesses
Ismail *et al.* 2022	Lightweight ML ensemble (WSS) for detecting cyber-attacks in Wireless Sensor Networks	- Efficient ensemble method tailored to energy-constrained environments - Dynamic classifier selection enhances adaptability	- Limited to WSN datasets, not SOC-specific - Simulated dataset may not reflect real-world data variety
Yadulla *et al.* 2023	Hybrid CNN-LSTM IDS for improved detection accuracy	- Effectively combines spatial and temporal analysis - High performance on both known and unknown threats - Scalable and adaptive design	- Lack of clarity on computational cost - May face real-time latency in high-throughput SOC settings
Kim *et al.* 2024	Human-in-the-loop framework for dynamic alert prioritization using HMMs	- Integrates analyst feedback for continuous learning - Significantly reduces alert fatigue and detection time - Uses advanced prioritization methods (Max Ratio/KL)	- Resource-intensive due to human involvement - May not scale well in large-volume SOCs without automation

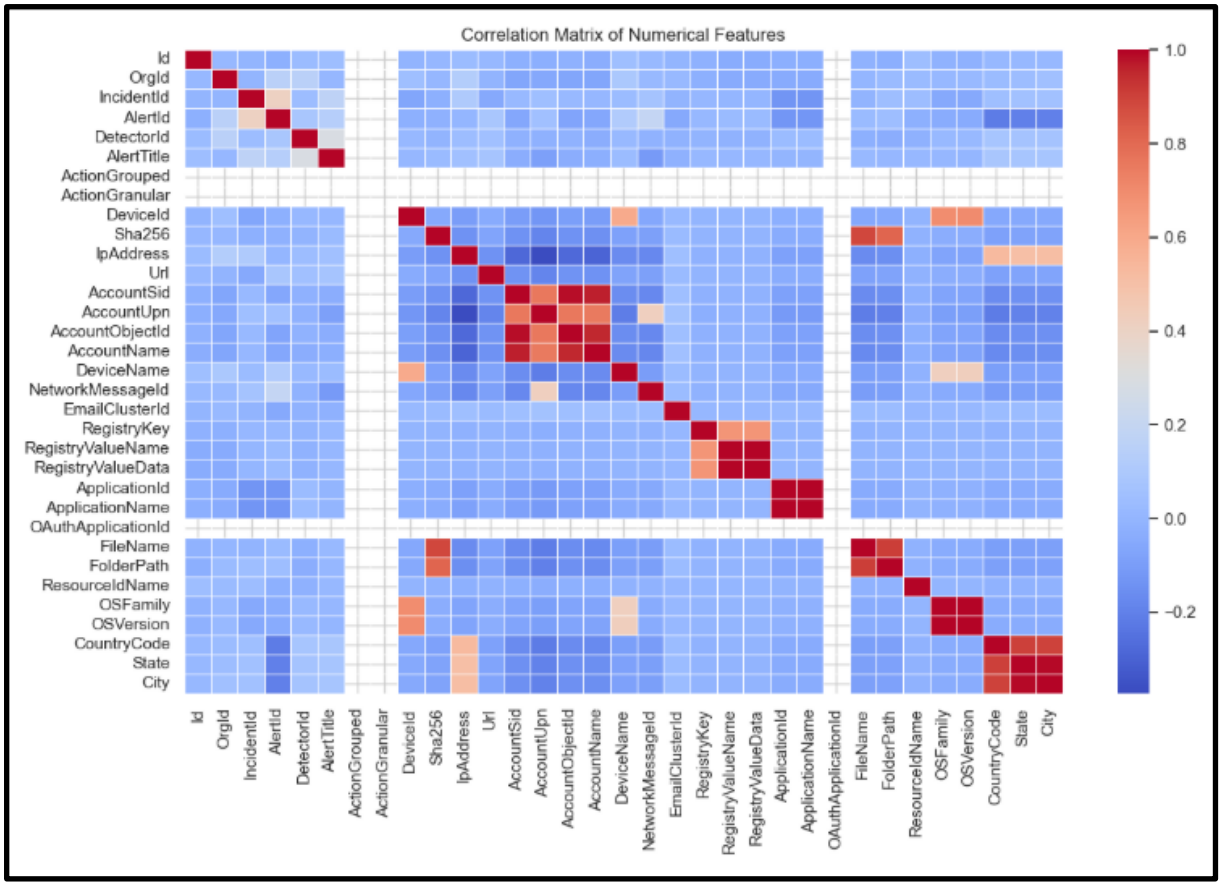
Oyinloye *et al.* 2025	Augmented ANN to handle unbalanced/noisy data for IDS	- Strong performance with 92% accuracy - Tackles class imbalance and data noise effectively - Random initialization boosts resilience	- Limited transparency on model interpretability - May lack generalizability to complex SOC scenarios
Alaeifar *et al.* 2024	CTI sharing: benefits, challenges, and architectures	- Comprehensive overview of CTI value in proactive defense - Discusses interoperability and legal compliance - Identifies gaps in schema standardization	- Theoretical focus with limited implementation insight - No AI/ML model introduced or tested
Odeh and Abu Taleb, 2023	Deep learning-based IoT intrusion detection using ensemble models (CNN-LSTM, CNN-GRU)	- Very high accuracy and F1-scores - Ensemble design enhances robustness - Addresses diverse IoT traffic types	- Primarily IoT-centric, not directly SOC-specific - Computational intensity not discussed in detail
Tran *et al.* 2022	IoT-DNN integration for CNC machine cybersecurity monitoring	- Real-world implementation in industrial settings - DNN detects both cyber and physical anomalies - Real-time monitoring using IoT sensors	- Focused on manufacturing environment, not SOC - Overemphasis on vibration analysis may limit applicability

Jilcha *et al.* 2025	Log anomaly detection using domain-specific PLM + Loss with Decaying Factor (LDF)	- Addresses zero-shot generalization and distributional shifts - Novel time-decay mechanism improves real-time relevance - High cross-dataset performance	- May require domain-specific PLMs for each new context - Implementation complexity may hinder wide adoption

Table 1: Summary Table

Appendix B: Visualizations




Correlation Matrix of Numerical Features
Id
OrgId
IncidentId
AlertId
DetectorId
AlertTitle
ActionGrouped
ActionGranular
DeviceId
Sha256
IpAddress
Url
AccountSid
AccountUpn
AccountObjectId
AccountName
DeviceName
NetworkMessageId
EmailClusterId
RegistryKey
RegistryValueName
RegistryValueData
ApplicationId
ApplicationName
OAuthApplicationId
FileName
FolderPath
ResourceIdName
OSFamily
OSVersion
CountryCode
State
City
1.0
0.8
0.6
0.4
0.2
0.0
−0.2