National College of Ireland

DEVELOPING EFFICIENT MACHINE LEARNING ALGORITHMS FOR ANOMALY DETECTION IN NETWORK TRAFFIC FOR CYBERSECURITY

MSc Research Project
MSc Cyber Security

Ganeshkumar Prabakaran
Student ID: 23410311

School of Computing
National College of Ireland

Supervisor: Michael Prior

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Ganeshkumar Prabakaran

Student ID: 23410311

Programme: MSc Cybersecurity **Year:** 2026

Module: Practicum Part 2

Supervisor: Michael Prior

Submission Due Date: 28th January 2026

Project Title: Developing Efficient Machine Learning Algorithms for Anomaly Detection in network traffic for cybersecurity

Word Count:6317 **Page Count:** 23

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: Ganeshkumar Prabakaran
Date: 28th January 2026

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):	

DEVELOPING EFFICIENT MACHINE LEARNING ALGORITHMS FOR ANOMALY DETECTION IN NETWORK TRAFFIC FOR CYBERSECURITY

Ganeshkumar Prabakaran
23410311

Abstract

In this study, efficient machine learning algorithms to detect network traffic anomalies are developed to enhance the current cybersecurity systems. The conventional rule-based intrusion detection is not able to detect the changing and unknown attacks, making it necessary to adopt adaptive and data-driven solutions. In this study, the systematic preprocessing with the NSLKDD benchmark dataset, exploratory analysis and implementation of the supervised and unsupervised models, Decision Tree, Random Forest and K-Means clustering are applied. Accuracy, precision, recall, F1-score, ROC-AUC and purity measures are used to evaluate model performance. It has been found that supervised models vastly outperform the unsupervised methods and that the results of the random forest are nearly perfect in detecting and having exceptional discriminatory ability. K-Means showed mediocre clustering as well as poor ROC. All in all, the results reveal that optimised supervised learning, in this case, Random Forest, is a very capable and computationally efficient approach to anomaly detection in networks and enhancing real-time systems of intrusion detection.

Table of Contents

Table of Figures

1 Chapter 1: Introduction

1.1 1.1 Background

Due to the digital phenomenon of changes, the interaction within modern networks faces growing loads of traffic, which can be easily susceptible to advanced cybercrime threats at present. Conventional methods of intrusion detection, such as static rule-based intrusion detection, are unable to keep up with the dynamism of attack patterns (Jain and Mitra, 2025). This causes high false alarms and is poorly responsive to unknown threats. Machine learning, in its turn, provides the potential of learning behaviour automatically with the data available and detecting an abnormality that can be a sign of malicious intent. Thus, the ML-based anomaly detection is becoming a promising approach to enhance cybersecurity. However, complex network environments are still issues because of efficiency and scalability (Hephzipah *et al.* 2023). The purpose of this dissertation is to investigate and streamline machine learning algorithms to enhance their accuracy and computation efficiency of recognising anomalies in network traffic.

1.2 1.2 Aim and Objective

Aim

This dissertation focused on the creation and optimisation of machine learning algorithms that can predict anomalies in network traffic precisely and efficiently to maximise the quality of cybersecurity, minimise the number of false alarms, and better detect real-time threats.

Objectives

- To review critically other kinds of machine learning methods that are in use to detect anomalies in network traffic.
- To create and apply to the chosen machine learning models on the benchmark cybersecurity datasets like CICIDS2017 and NSL-KDD.
- To compare the performance of the individual models in terms of accuracy, precision, recall, F1-score, AUC and computational efficiency.
- To suggest a streamlined machine learning solution that can enhance real-time anomaly detection and powerful cybersecurity applications.

1.3 1.3 Research Question

Q1. Which machine learning algorithms may be formulated or optimised to enhance the precision and effectiveness of network traffic anomaly detection?
Q2. What are the performance rates of various supervised and unsupervised machine learning-based models on benchmark data, including CICIDS2017 and NSL-KDD?
Q3. What are the performance variations between the models tested based on accuracy, precision, recall, F1score, AUC and images?
Q4. What is the most suitable machine learning approach that can be advised to be used to increase the cybersecurity intrusion detection systems in real-time?

1.4 1.4 Problem Statement

Cyberattacks are advancing in terms of intelligence and dynamism, but most of the traditional intrusion detection systems continue to utilise the assistance of fixed rules and signatures that are unable to identify new or emerging threats. The effect of these limitations on the rate of detection and false positives is usually high, making security gaps in very important network environments (Goswami, 2024). Despite its good potential, machine learning has had several challenges that remain to be solved by most of the models; these include poor scalability, computational and real-time models. Consequently, there is an urgent requirement to come up with more precise, optimised and efficient machine learning algorithms to be able to detect network traffic anomalies.

1.5 1.5 Summary

The chapter clarified the importance of network anomaly detection in current cybersecurity and the shortcomings of the conventional approaches. It provided an introduction to machine learning as a promising remedy, presented the aim, objectives and research questions of the research, and reasoned that it is necessary to come up with effective and accurate machine learning models in terms of detecting cyber threats.

2 Chapter 2: Literature Review

2.1 2.1 Introduction

This chapter uses a critical review of intrusion detection evolution, which saw the shift of traditional rule-based systems to sophisticated machine learning techniques. It examines data, metrics of evaluation, and contemporary obstacles of real-time anomaly detection to lay the theoretical background and research rationale of creating effective ML-based models of cybersecurity.

2.2 2.2 Evolution of Intrusion Detection Systems and Limitations of Traditional Approaches

The first form of Intrusion Detection Systems implemented signature-based processes wherein known types of attacks were compared with the incoming data. This approach was successful against all attacks that were known in the past since it relied on pre-defined signatures, making it ineffective to operate with new attacks and zero-day and polymorphic attacks (Iyer, 2021). This weakness is increased in the case of IoT environments due to the disseminated nature of devices and resources, their dynamism, and their generation of dynamic traffic, some of which are beyond the capability of conventional IDS architecture to accommodate the heterogeneous and fast-evolving behaviour (Heidari and Jamali, 2023). Signature-based IDS is also not very adaptable and, in most cases, will give false positives based on predefined rule sets.

Machine learning has been investigated to enhance the detection of threats based on behavioural patterns learning. Nonetheless, ML-based strategies are not flawless as well because they can only generalise to unknown dangers, based on the quality of the training data and computer resources (Muneer *et al.* 2024). Deep learning also enhances representation learning and needs large amounts of labelled data, which is not available in real-world networks. Despite federated learning providing benefits in terms of data privacy, it is more expensive to communicate with. In general, the literature demonstrates a shift in the paradigm of detection based on rules to adaptive models, but still, there is no unanimous solution balancing between accuracy, efficiency, scalability and real-time responsiveness in contemporary network settings.

2.3 2.3 Machine Learning Techniques for Network Anomaly Detection

Machine learning has emerged as a prominent approach in the process of identifying abnormal behaviours in network traffic since it has the ability to provide flexibility not available in traditional rule-based methods. Wang *et al.* (2021) also emphasise that machine learning is able to be used to identify known and unknown attacks in different network topologies, although its ability to do so will depend on the algorithm and the quality of data on the network. Labelled datasets are also well suited to supervised models like Random Forests, SVM and Gradient Boost algorithms, which are reported to deliver strong classification performance. Nevertheless, the need to have labelled data restricts its application in dynamic settings. The methods that can be classified as unsupervised, such as clustering and autoencoders, work better in situations of unknown attacks, particularly in IoT ecosystems where novel behavioural patterns are being generated constantly (Haji and Ameen, 2021).

Recent methods discuss hybrid and ensemble methods of increased robustness, reduced false alarms and using complementary model strengths. According to Yaseen (2023), adaptive machine learning and enhanced feature engineering are becoming more crucial towards enhancing the quality of detection. Nevertheless, its implementation in combination with deep learning, ensembles and privacy-preserving models remains associated with the issue of increased computational load and resource constraints (Kurniabudi *et al*. 2019). Thus, machine learning has become an extremely promising approach, but the choice of the appropriate algorithm is determined by the peculiarities of the data, the complexity of the network, and the required realtime performance of the operation.

2.4 2.4 Benchmark Cyber security Datasets Used for Model Evaluation

The data sets applied to the analysis of cyber security models have changed considerably, especially with the introduction of more sophisticated attack patterns and the extensive AI integration. However, conventional anomaly detection research generally employs network data, including the CICIDS2017, the NSL-KDD and the

UNSW-NB15, to evaluate detection quality. Nevertheless, the more sophisticated and context-oriented benchmark data have a tendency to change according to recent literature. As an example, Tihanyi *et al.* (2024), proposed a large human-verified dataset named CyberMetric, and its purpose was to assess the AI models in relation to cybersecurity knowledge, showing that nowadays benchmark datasets evolve beyond the domain of packet flows to one of domain reasoning.

Liu *et al*. (2024) continue this line of thought and created a tool (CyberBench) that can be used to test multiple tasks that are cybersecurity-oriented, so that the assessment of LLM is more holistic. In the same light, Yu *et al.* (2024) developed CS-Eval, an open bilingual evaluation, which classifies questions into cognitive levels and offers a platform to evaluate questions in a structured format over 42 categories of cyber security. Even though the presented recent datasets predominantly aid the purpose of benchmarking the language models, they can be regarded as the continuation of the large-scale trend: performance testing in cyber security is becoming a multi-dimensional phenomenon, necessitating datasets that not only test the ability to detect attacks, but also testing the ability to be adaptable, reasoning-based, and contextually understanding. It means that it has a gap with traditional datasets on network traffic, which are still significant to the anomaly detection models, yet there are new benchmarks with broader evaluation areas.

2.5 2.5 Performance Evaluation Metrics in ML-Based Intrusion Detection Research

The metrics of performance evaluation are important in establishing the effectiveness of a machine learningbased IDS model under different cyber threat conditions. Several other studies have provided explicit pointers to the universal inadequacy of common ML metrics involving accuracy alone, particularly in a low-resource IoT or edge computing deployment setting (Jamshidi et al., 2025). Measures like CPU load, memory usage, and energy usage have to be examined as well since models used in the edge may impair the performance of the devices when handling real-time attacks.

Nevertheless, traditional research tends to focus on the accuracy, precision, recall, F1-score and detection rate to rank the various algorithms on the benchmark datasets. In fact, as the example of Al Farsi et al. (2024), unlike ensemble methods, which performed exceptionally high in terms of accuracy and the ability to detect minority-class attacks on CICIDS2017, other methods do not consistently perform on all subsets of attacks, which is why it is necessary to assess various metrics instead of only accuracy. Conversely, in Hidayat *et al.* (2023), the accuracy, as well as precision and recall, are used to compare two types of models, which are ML and DL models with the TON_IoT dataset and justify that low false alarms and balanced results are key factors in the current IDS. In general, it can be concluded that recent literature indicates that detection performance and computational efficiency are required to ensure that studies in the field of IDS can be meaningful in the real world.

2.6 2.6 Current Challenges and Future Directions for Real-Time Anomaly Detection

Even with the fast development of anomaly detection studies, real-time application is a challenging task because of the complexity of data and practical limitations. Yang and Zhang (2023) say that the environment of the IoT necessitates solutions that could handle heterogeneous, noisy, and very high-volume streams of data in milliseconds, yet the existing models do not provide a stable level of real-time responsiveness at the cost of accuracy. Also, privacy-sensitive situations further inhibit data sharing in training to constrain generalisation. Similar challenges can be noted in the field of time series, where Arslan *et al*. (2023) state that in highdimensional streaming data, a high-dimensional data model inevitably results in performance bottlenecks and delayed responses when alerts are required, thus requiring support beyond a static ML model. The current deep learning models have improved representation learning, yet they are computationally intensive and subject to model drift, which necessitates that models must always be retrained. The application of real-time computer vision anomaly detection to industrial contexts by Alzarooni *et al.* (2025) indicates that improved pipelines are needed in data acquisition, pre-processing and decision making. Hence, any further work should focus on adaptive online learning, lightweight inference based on edges towards privacy protection, federated learning, and self-evolving models that have the ability to adapt to novel threats independently. In general, the literature demonstrates that scalable answer to explainable and resourceefficient frameworks with the ability to work autonomously under changing conditions are in great demand.

2.7 2.7 Research Gap

Despite the evidence of the literature describing the transformation of signature-based IDS into machine learning-based and hybrid models of detection, in no case there is no single solution that satisfies the criteria of accuracy, computational efficiency, scalability, and real-time responsiveness (Gopalsamy, 2022). Most studies choose to investigate definition or resource optimisation only, and not both. Hence, an obvious disconnect in creation and testing optimised ML models with the ability to effectively identify abnormalities and at the same time be lightweight enough to run in real time in the complex but dynamic network scenarios can be seen.

2.8 2.8 Summary

Overall, in the literature, it is found that machine learning enhances the detection abilities of anomalies; however, there are considerable gaps in the attainment of accuracy, scalability, and real-time operational performance. The old datasets, measures and frozen models are too few. Consequently, to improve the current network security and facilitate efficient anomaly detection, optimised and adaptive ML solutions are needed.

3 Chapter 3: Methodology

3.1 3.1 Introduction

The chapter is describing the methodological framework which will be used in the study outlining the philosophy of research, approach, design, data collection, preprocessor, model implementation and evaluation techniques. It introduces the systematic experimentation of machine learning algorithms to benchmark datasets to generate reliable, objective, and reproducible results regarding network traffic anomaly detection.

3.2 3.2 Research Philosophy

The research is assuming a positivist research philosophy since it is continuously basing on objective, measurable, and evidence that are data-driven to investigate machine learning methods in detecting anomalies. The study is also making its emphasis on the patterns of observable trends within benchmark network-traffic data sets and use statistical evaluation to obtain objective results (Buzzio-García *et al*. 2024). This position is putting the study on track to ensure that findings are interpretable, measurable, and devoid of personal interpretation, and a systematic approach of exploring issues of cyber security, as well as its ongoing success, is maintained.

3.3 3.3 Research Approach

The study adopting a deductive and quantitative research method since the study is constantly testing predetermined theories regarding the effectiveness of machine learning in the process of anomaly detection. In applying the deductive approach, the study is implementing the known cyber security and ML concepts to scorecard datasets and is analysing the applicability of the findings to the theoretical means (Abdulhamid *et al.* 2025). This is what can assist the study, as it offers a well-organized, unambiguous direction in which algorithms will be compared, assumptions tested, and objective, evidence-based inferences made regarding making detection efficiency improvements.

3.4 3.4 Research Design

The current research is using a quantitative, experimental research design in which three variants of machine learning algorithms are implemented, tested, and evaluated on the standardised cyber security data. The architecture is paying attention to the controlled and systemized procedures like data pre-processing, feature engineering, model training, and performance evaluation (Sarker, 2021). This structure is important because it enables the study to get consistent measurements, objective comparison as well as replicable findings. The research design is assisting because it has offered a strict guideline to determine the algorithms that give the best accuracy, efficiency, and reliability of detecting anomalies in network traffic.

3.5 3.5 Data Collection Method

The research is gathering information based on the secondary data sources, i.e., the benchmark cyber security datasets, including NSL-KDD and CICIDS2017. The reason these datasets are being chosen is that they both have labelled and unlabelled network traffic data, comprising normal behaviour as well as several types of attacks, as is required to assess machine learning models (Maseer *et al*. 2021). The instrument used in the data collection is that the study uses large-scale, diverse, and realistic samples of traffic. The secondary datasets that are being used in the research are useful in that they offer reliable publicly validated data that will facilitate accurate testing, comparison and reproducibility of the anomaly-detection algorithms.

3.6 3.6 Data Analysis Technique

The paper is utilising various data analysis methods to assess the efficacy of machine learning models in identifying network traffic anomalies. It is commencing with cleaning, legalisation and reduction of data sets by correlational filtering and Principal Component Analysis. There is also several algorithms undergoing implementations, such as random forests, decision trees, k-Means clustering and a hybrid model that merges supervised and unsupervised learning environments (Khanan *et al.* 2024). Quantitative indicators, including accuracy, precision, recall, F1-score, ROC-AUC, training time and computational efficiency, are used in analysing each model. The techniques are assisting the research by providing comparisons in measurable and structured

forms, finding out which algorithm performed best, and supporting the results that anomaly detection is improving in benchmark datasets.

3.7 3.7 Ethical Consideration

The current research is making sure that the ethical standards remain high by only referring to the opensource and anonymised cyber security datasets that do not include personal, or identifiable information. The study is conducted on patterns of network traffic without any information on actual users and hence ensures privacy and confidentiality. Every process in machine learning is recorded in an open and responsible manner to promote responsible and reproducible AI (Mvula *et al*. 2023). The research is also making sure that the developed models are operated in pure usage as related to defensive purposes of cyber security, and no misuse or inadvertent damage is observed.

3.8 3.8 Summary

This chapter has outlined the steps in an organised manner of analysing the network traffic with the help of machine learning methods. The productive methodology will provide accuracy, consistency, and transparency because of the use of quantitative techniques, secondary data, and strong indicators of evaluation. It forms the basis of creating an effective, similar and extendable framework of detecting abnormalities in cyber security setups.

4 Chapter 4: Results and Discussion

4.1 4.1 Introduction

In this chapter, the author reveals the findings of the machine learning models used on the NSL-KDD dataset, with the help of the visualisation and the study of the data. It examines the performance of the models, the comparison of the supervised and unsupervised models, and gives a discussion on their performance regarding the detection of anomalies in the network traffic.

4.2 4.2 Results

Use of important Libraries

```
import warnings
warnings.filterwarnings("ignore")

import numpy as np
import pandas as pd
from scipy.io import arff
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.impute import SimpleImputer
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score, classification_report
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt
```

Data set import and showing the data head

```
data = pd.read_csv("KDDTrain+.txt")
df = data.copy()
df.head()
```

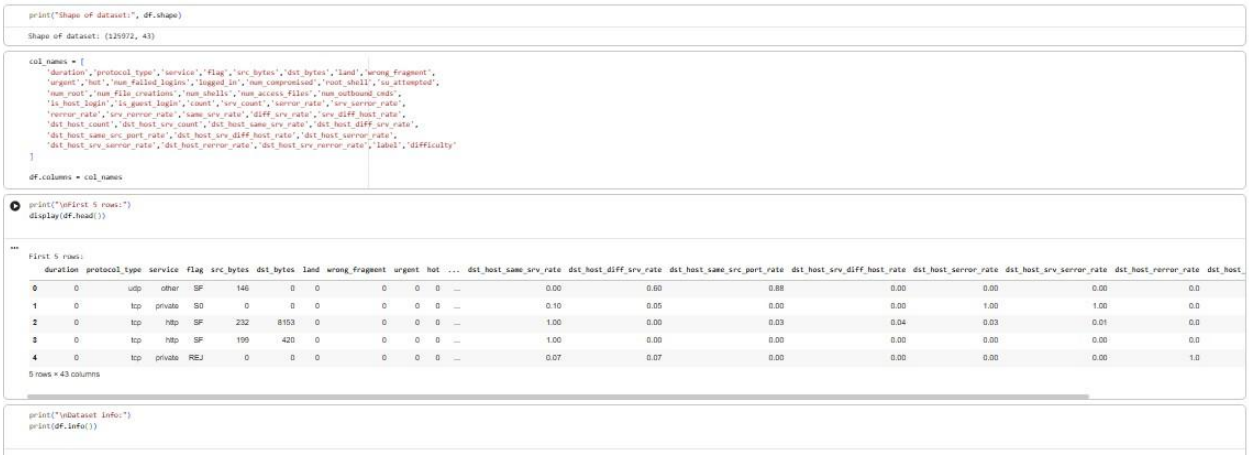
	0	tcp	ftp_data	SF	491	0.1	0.2	0.3	0.4	0.5	...	0.17	0.03	0.17.1	0.00.6	0.00.7	0.00.8	0.05	0.00.9	normal	20
0	0	udp	other	SF	146	0	0	0	0	0	...	0.00	0.60	0.88	0.00	0.00	0.00	0.0	0.00	normal	15
1	0	tcp	private	S0	0	0	0	0	0	0	...	0.10	0.05	0.00	0.00	1.00	1.00	0.0	0.00	neptune	19
2	0	tcp	http	SF	232	8153	0	0	0	0	...	1.00	0.00	0.03	0.04	0.03	0.01	0.0	0.01	normal	21

4.2.1.1.1 Figure 4.1: Use of Important libraries and data head shows

(Source: Self-Created)

In this section, all the necessary Python libraries necessary to process data, machine learn, visualise and evaluate are imported. Data structures are handled by libraries like NumPy and Pandas, Scikit-Learn handles such data, preprocessing as well as the implementation of the models, and is generated by Matplotlib are the visualizations. They are used collectively to build a full-fledged environment regarding the analysis of the NSLKDD dataset and application of the anomaly-detection models.

Pandas loads the dataset in the form of “KDDTrain+.txt” file, and the first five lines are shown to verify that it was loaded successfully. This preview allows checking the column alignment, data types and existence of such features as protocol type, service, flag, number of bytes, and labels. It makes sure that the dataset is adjusted to the format of NSL-KDD prior to further analysis (Yaseen, 2023).



```
print("Shape of dataset:", df.shape)

Shape of dataset: (125972, 43)

col_names = [
    'duration','protocol_type','service','flag','src_bytes','dst_bytes','land','wrong_fragment',
    'urgent','hot','num_failed_logins','logged_in','num_compromised','root_shell','su_attempted',
    'num_root','num_file_creations','num_shells','num_access_files','num_outbound_cmds',
    'is_host_login','is_guest_login','count','srv_count','serror_rate','srv_serror_rate',
    'rerror_rate','srv_rerror_rate','same_srv_rate','diff_srv_rate','srv_diff_host_rate',
    'dst_host_count','dst_host_srv_count','dst_host_same_srv_rate','dst_host_diff_srv_rate',
    'dst_host_same_src_port_rate','dst_host_srv_diff_host_rate','dst_host_serror_rate',
    'dst_host_srv_serror_rate','dst_host_rerror_rate','dst_host_srv_rerror_rate','label','difficulty'
]

df.columns = col_names

print("\nFirst 5 rows:")
display(df.head())

...
First 5 rows:
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_same_srv_rate	dst_host_diff_srv_rate	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	dst_host_
0	0	udp	other	SF	146	0	0	0	0	0	...	0.00	0.60	0.88	0.00	0.00	0.00	0.0	
1	0	tcp	private	S0	0	0	0	0	0	0	...	0.10	0.05	0.00	0.00	1.00	1.00	0.0	
2	0	tcp	http	SF	232	8153	0	0	0	0	...	1.00	0.00	0.03	0.04	0.03	0.01	0.0	
3	0	tcp	http	SF	199	420	0	0	0	0	...	1.00	0.00	0.00	0.00	0.00	0.00	0.0	
4	0	tcp	private	REJ	0	0	0	0	0	0	...	0.07	0.07	0.00	0.00	0.00	0.00	1.0	

5 rows × 43 columns

```
print("\nDataset info:")
print(df.info())
```

4.2.1.1.2 Figure 4.2: Steps of Exploratory Data Analysis

(Source: Self-Created)

The data set has 125973 records and 43 features which were properly named and complete. Deck previewing confirms that it is well-structured, and the categorical and numerical attributes of the first rows have meaning. There is no data loss in the dataset, so preprocessing, modelling, and analysis of anomalies have clean and consistent and machine-learning-ready data.

```
print("\nMissing values per column:")
print(df.isnull().sum())

Missing values per column:
duration                     0
protocol_type                0
service                      0
flag                         0
src_bytes                    0
dst_bytes                    0
land                         0
wrong_fragment               0
urgent                       0
hot                          0
num_failed_logins            0
logged_in                    0
num_compromised              0
root_shell                   0
su_attempted                 0
num_root                     0
num_file_creations           0
num_shells                   0
num_access_files             0
num_outbound_cmds            0
is_host_login                0
is_guest_login               0
count                        0
srv_count                    0
serror_rate                  0
srv_serror_rate              0
rerror_rate                  0
srv_rerror_rate              0
same_srv_rate                0
diff_srv_rate                0
srv_diff_host_rate           0
dst_host_count               0
dst_host_srv_count           0
dst_host_same_srv_rate       0
dst_host_diff_srv_rate       0
dst_host_same_src_port_rate  0
dst_host_srv_diff_host_rate  0
dst_host_serror_rate         0
dst_host_srv_serror_rate     0
dst_host_rerror_rate         0
dst_host_srv_rerror_rate     0
label                        0
difficulty                   0
dtype: int64
```

4.2.1.1.3 Figure 4.3: Checking if the data set contains null values

(Source: Self-Created)

This part substantiates the fact that there is no missing value within all 43 features. NSL-KDD is clean and is benchmark-clean, and its completeness ensures that imputation is not required. This also makes sure that model training is not influenced by data discontinuities, as well as ensuring the sound testing of decision trees, random forests, and clustering algorithms (PM and Soumya, 2024).

```
print("\nStatistical Summary:")
display(df.describe())
```

```
Statistical Summary:
```

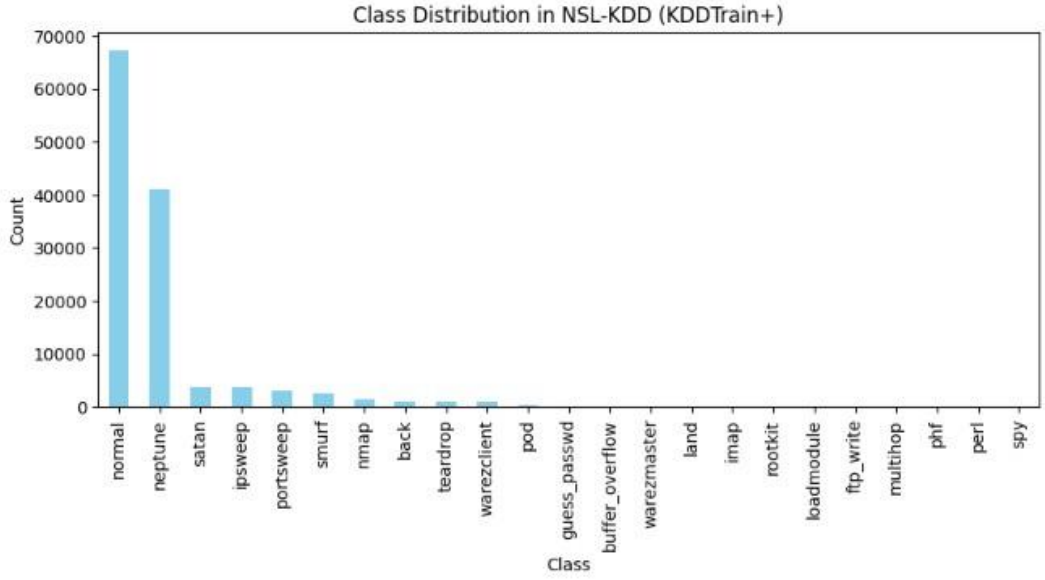
	duration	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logged_in	num_compromised	...	dst_ho
count	125972.000000	1.259720e+05	1.259720e+05	125972.000000	125972.000000	125972.000000	125972.000000	125972.000000	125972.000000	125972.000000	...	1
mean	287.146929	4.556710e+04	1.977927e+04	0.000198	0.022688	0.000111	0.204411	0.001222	0.395739	0.279253	...	
std	2604.525522	5.870354e+06	4.021285e+06	0.014086	0.253531	0.014366	2.149977	0.045239	0.489011	23.942137	...	
min	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	
25%	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	
50%	0.000000	4.400000e+01	0.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	
75%	0.000000	2.760000e+02	5.160000e+02	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	...	
max	42908.000000	1.379964e+09	1.309937e+09	1.000000	3.000000	3.000000	77.000000	5.000000	1.000000	7479.000000	...	

8 rows × 39 columns

4.2.1.1.4 Figure 4.4: Visualisation of descriptive statistics

(Source: Self-Created)

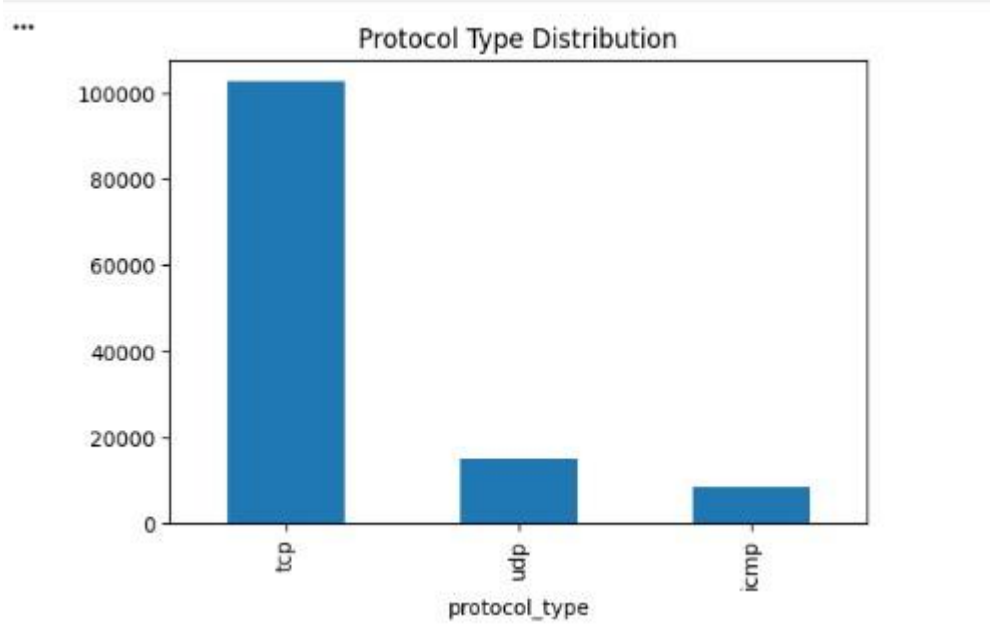
The statistical summary offers descriptive information of all the numerical characteristics, indicating their number, average, standard deviation and range. It points out to the high variability of traffic qualities whereby numerous characteristics are greatly skewed out. The summary aids in detecting normal behaviour patterns and large changes in the NSL-KDD data.



4.2.1.1.5 Figure 4.5: Class Distribution in NSL-KDD (KDDTrain+)

(Source: Self-Created)

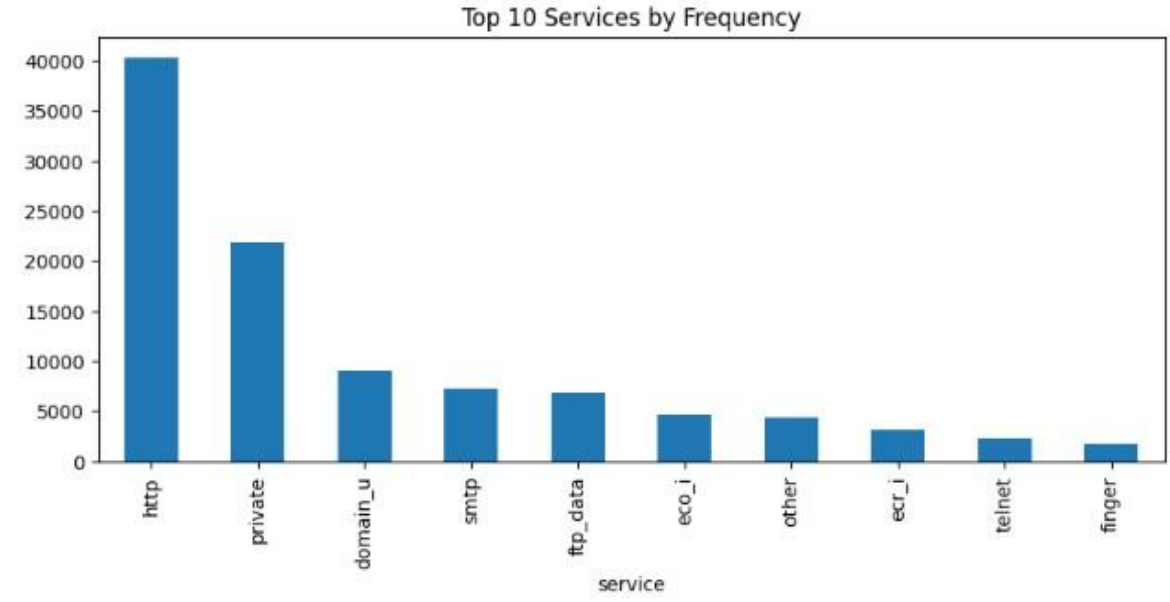
The plot of the class distribution indicates that the NSL-KDD dataset has a severe imbalance of the sample, with normal and “neptune” being the prevailing classes. A sort of amount of attack types are infrequent and few. This skew emphasises the problem of machine learning models because minor classes must be handled with great caution to avoid biased learning and decreased detection performance.



4.2.1.1.6 Figure 4.6: Class Distribution in NSL-KDD

(Source: Self-Created)

The protocol distribution indicates that TCP is the protocol to which most network connections are conducted, with a very small number of UDP and ICMP packets. This imbalance means that TCP-based traffic is prevalent in the dataset, and this will affect model learning; moreover, anomalies in the most popular communication protocol should be classified accordingly.



4.2.1.1.7 Figure 4.7: Protocol Type Distribution

(Source: Self-Created)

In the service frequency plot, it can be noted that the most frequent service is HTTP, then it is the private, domain_u and SMTP. The less common services, such as eco_i, telnet and finger, are infrequently listed. This variation is based on the fact that people have varied traffic behaviours and the models have the ability to distinguish between regular patterns and possible attacks related to services.

```
if 'label' in df.columns:
    label_col = 'label'
else:
    label_col = df.columns[-2]   # second last column is always the label in NSL-KDD

print("Using label column:", label_col)

df['binary_label'] = df[label_col].apply(
    lambda x: 'normal' if str(x).strip() == 'normal' else 'attack'
)

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['y_binary'] = le.fit_transform(df['binary_label'])

print(df[['binary_label','y_binary']].head())
scaler = StandardScaler()
X_num_scaled = pd.DataFrame(scaler.fit_transform(X_num), columns=X_num.columns)

X = pd.concat([X_num_scaled.reset_index(drop=True), df_enc.reset_index(drop=True)], axis=1)
y = df['y_binary'].values

print("\nFinal feature matrix shape:", X.shape)
```

```
Using label column: label
  binary_label  y_binary
0       normal         1
1       attack         0
2       normal         1
3       normal         1
4       attack         0

Final feature matrix shape: (125972, 72)
```

4.2.1.1.8 Figure 4.8: Data preprocessing before model implementation

(Source: Self-Created)

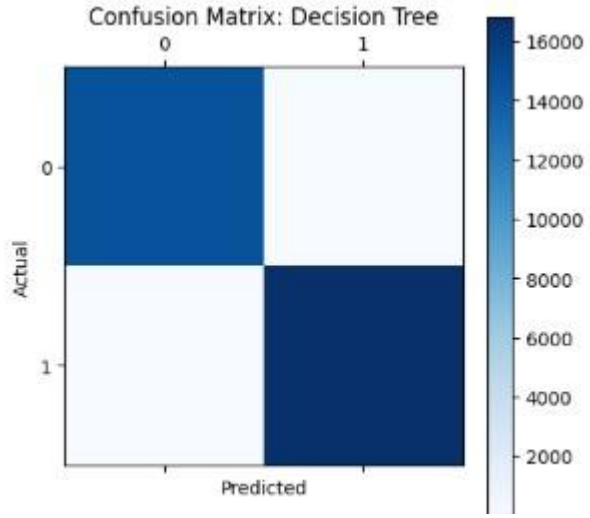
The preprocessing phase detects the appropriate column of labels, transforms the types of attacks into a binary label, and encodes the label. Scaling of the numerical features is done using StandardScaler, and the categorical features are one-hot encoded. Lastly, all of the built features are combined in one matrix that can then be trained into a model.

```
=== Decision Tree ===
Accuracy: 0.9981
Precision: 0.9976
Recall: 0.9988
F1-score: 0.9982

Classification report:
              precision    recall  f1-score   support

      attack       1.00      1.00      1.00     14658
      normal       1.00      1.00      1.00     16835

    accuracy                           1.00     31493
   macro avg       1.00      1.00      1.00     31493
weighted avg       1.00      1.00      1.00     31493
```



4.2.1.1.9 Figure 4.9: Decision tree Implementation

(Source: Self-Created)

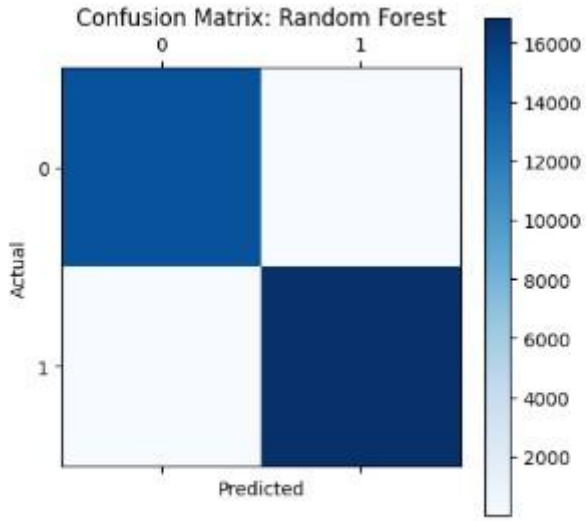
The Decision tree model is shown to have a very high predictive accuracy, and it is seen that the model has a 99.81% accuracy, a 99.87% precision, 99.88% recall, with a 99.82% F1-score. In the classification report, the separation of classes between attack and normal is perfect or close to being perfect. In the confusion matrix, the misclassifications are very minimal, which represents the fact that the model is effective in representing hierarchical decision rules based on the NSL-KDD data. The fact that the dominance of the near diagonal in the matrix is strong establishes a great ability to discriminate (Maddireddy and Maddireddy, 2024).

```
=== Random Forest ===
Accuracy: 0.9990
Precision: 0.9985
Recall: 0.9996
F1-score: 0.9991

Classification report:
              precision    recall  f1-score   support

      attack       1.00      1.00      1.00     14658
      normal       1.00      1.00      1.00     16835

    accuracy                           1.00     31493
   macro avg       1.00      1.00      1.00     31493
weighted avg       1.00      1.00      1.00     31493
```




```
ROC-AUC: 0.9999508991117669
```

4.2.1.1.10 Figure 4.10: Random forest implementation

(Source: Self-Created)

The strongest performer in this study is the Random Forest model since it has an accuracy of 99.90%, a precision of 99.85%, a recall of 99.96%, and an F1-score of 99.91. ROC-AUC score is 0.9999, which means that there is almost complete separation of attack and normal traffic. The confusion table indicates many few misclassifications, and this is a sign of the strength of the ensemble to work with noisy and complex patterns. The model decreases variance and overfitting by merging several decision trees, giving the model predictions of the highest stability (Abdullah *et al.* 2024). Because of its higher floating capacity to produce nonlinear interactions, Random Forest proves to be particularly effective in detecting cybersecurity anomalies in a highdimensional dataset format such as NSL-KDD.

```
n_clusters = 2  # binary: normal vs attack
kmeans = KMeans(n_clusters=n_clusters, random_state=42, n_init=10)
kmeans.fit(X)
clusters = kmeans.labels_

def cluster_purity(true_labels, cluster_labels):
    dfc = pd.DataFrame({'true': true_labels, 'cluster': cluster_labels})
    grouped = dfc.groupby('cluster')['true'].apply(lambda x: x.value_counts().max())
    return grouped.sum() / len(dfc)

purity = cluster_purity(y, clusters)
print(f"\nKMeans clustering purity: {purity:.4f}")
```

```
KMeans clustering purity: 0.9068
```

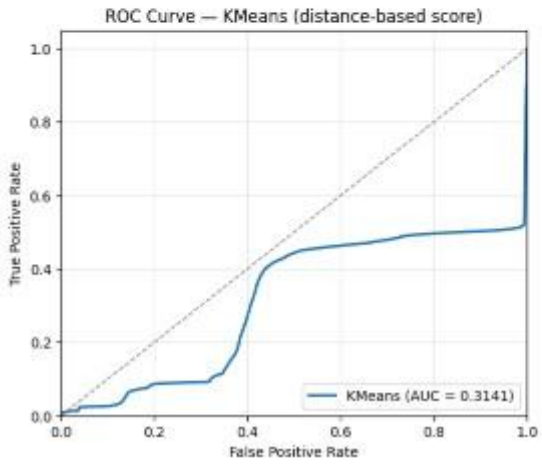
4.2.1.1.11 Figure 4.11: K-means model implementation

(Source: Self-Created)

The K-Means model that was set to one cluster of normal and one cluster of attack traffic had a purity measure of 0.9068, which is a fairly good separation. Though it is an unsupervised clustering, the purity of the clusters of high quality indicates purposeful grouping, yet it still does not perform as well as supervised clustering models such as Decision Tree and Random Forest.

```
distances = kmeans.transform(X)
scores = np.min(distances, axis=1)
fpr, tpr, thresholds = roc_curve(y, scores)
roc_auc = auc(fpr, tpr)

# plot ROC curve
plt.figure(figsize=(6,5))
plt.plot(fpr, tpr, lw=2, label=f'KMeans (AUC = {roc_auc:.4f})')
plt.plot([0,1], [0,1], linestyle='--', lw=1, color='grey')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve — KMeans (distance-based score)')
plt.legend(loc='lower right')
plt.grid(alpha=0.3)
plt.show()
```



4.2.1.1.12 Figure 4.12: Visualisation ROC curve of the K-cluster model

(Source: Self-Created)

K-Means curve on ROC has low discriminatory potential as the AUC is 0.3141, implying that it is not a brilliant separator between an attack and normal traffic. K-Means is an unsupervised algorithm that is unable to represent acute patterns, which makes it a poor competitor to supervised classifiers in the dataset.

4.3 4.3 Discussion

The findings prove that the supervised machine learning models are far more successful than unsupervised in alerting against anomalies in the NSL-KADD dataset. Both Decision Tree and Random Forest models were extremely accurate, precise, recall and F1-score; however, Random Forest worked out better because of the ability to make an ensemble learning process, less variance, and the ability to revive noisy and highdimensional data (Roy, 2024). The fact that ROC-AUC is almost perfect further proves that it possesses good discriminatory power.

The K-Means model clustering model, with a relatively higher purity score, on the contrary, was weak when it comes to ROC performance, indicating the shortcomings of the unsupervised models when handling the intricate attack patterns. The imbalance of the dataset, which consisted of a majority of normal and “neptune” data, probably affected the performance of the model, which supports the significance of supervised learning in the presence of labelled data (Okoli *et al.* 2024). All in all, the results confirm that Random Forest is the best algorithm to use when anomalies in the network must be identified, and prove that the clustering method alone is not reliable to allow distinguishing between such sophisticated attack behaviours.

4.4 4.4 Summary

The chapter showed that the most accurate and detection capability of models were supervised models, and 1 Random Forest was the most accurate. Exploratory analysis verified the quality of data, and the metrics of evaluation revealed a high level of classification to make the conclusions that the most trustworthy tool to recognise network anomalies in NSL-KDD is the Random Forest.

5 Chapter 5: Conclusion

5.1 5.1 Conclusion

The study aimed at creating and testing effective machine learning algorithms that would be able to identify anomalies in network traffic with precision to improve cybersecurity. The study conducted a systematic experiment on NSL-KDD benchmark dataset and found that supervised learning models are much more successful and reliable compared to unsupervised learning methods. Thorough data exploration helped identify the integrity of the dataset and, therefore, facilitated the process of preprocessing and training the model. Random Forest was the most effective algorithm among those put in place as it had close to perfect accuracy, precision, recall, F1-score and ROC-AUC. The detail of its ensemble structure enabled it to achieve nonlinear attack behaviours and nonlinear attack behaviours, along with staying strong to noise and imbalance, so it is the most appropriate technique in intrusion detection. Decision Tree also fared very well but took slightly higher variance than the Random Forest. Conversely, the K-Means clustering model exhibited fair purity of clusters with a poor ROC value, which underscored the vulnerability of the unsupervised models in identifying complex cyber threats.

Broadly, the results prove that optimised supervised machine learning is an effective and useful solution to the current problem of anomaly detection. The paper supports the use of robust pre-processing, labelled training datasets, and ensemble-based solutions to counter advancing network assaults in strengthening real-time cybersecurity defence systems.

5.2 5.2 Recommendation

With regards to the findings of the study, organisations are advised to focus on supervised machine learning models, especially Random Forest, in detecting anomalies in their networks owing to the fact that they have a high degree of accuracy in reaching conclusions, robustness and flexibility with regard to the more complex attack patterns. Proper feature scaling, encoding, and balancing of the datasets among others are preprocessing measures that need to be continually taken to improve the performance of detection. Also, ensemble-based monitoring pipelines must be used to reduce false positives and enhance real-time threat detection by cybersecurity systems. Frequent retraining on new datasets is also recommended to guarantee the further relevance and reliability of the model.

5.3 5.3 Future Scope

This can be followed up in the future by incorporating deep learning models, like LSTM and autoencoders, to encode and decode temporal attack behaviours and increase the chances of detection of new threats. There should be a possibility to implement it in real-time on an edge or fog computing framework, which can enhance scalability and latency performance. Generalisation capabilities may be further tested by testing on other datasets such as CICIDS2017 and UNSW-NB15. It might also be beneficial to involve hybrid models of supervised, unsupervised, and online learning methods in order to enhance adaptability in a changing environment and enhance advanced intrusion detection systems.

6 References

Abdulhamid, A., Kabir, S., Ghafir, I., Lei, C., El Hindi, K. and Hammoudeh, M., 2025. Quantitative Cybersecurity Analysis Framework for Cyber Physical Systems: A Conceptual Approach. *IEEE Open Journal of the Computer Society*.

Abdullah, M.M., Khan, H., Farhan, M. and Khadim, F., 2024. An Advance Machine Learning (ML) Approaches for Anomaly Detection based on Network Traffic. *Spectrum of engineering sciences*, *2*(3), pp.502-527.

Al Farsi, A., Khan, A., Bait-Suwailam, M.M. and Mughal, M.R., 2024. Comparative Performance Evaluation of Machine Learning Algorithms for Cyber Intrusion Detection.

Alzarooni, A., Iqbal, E., Khan, S.U., Javed, S., Moyo, B. and Abdulrahman, Y., 2025. Anomaly detection for industrial applications, its challenges, solutions, and future directions: A review. *arXiv preprint arXiv:2501.11310*.

Arslan, F., Javaid, A. and Awan, M.D.Z., 2023. Anomaly detection in time series: current focus and future challenges.

Buzzio-García, J., Vergara, J., Ríos-Guiral, S., Garzón, C., Gutiérrez, S., Botero, J.F., Quiroz-Arroyo, J.L. and PérezDíaz, J.A., 2024. Exploring traffic patterns through network programmability: introducing sdnflow, a comprehensive openflow-based statistics dataset for attack detection. *IEEE Access*, *12*, pp.42163-42180.

Diana, L., Dini, P. and Paolini, D., 2025. Overview on intrusion detection systems for computers networking security. *Computers*, *14*(3), p.87.

Gopalsamy, M., 2022. Scalable anomaly detection frameworks for network traffic analysis in cybersecurity using machine learning approaches. *Int. J. Curr. Eng. Technol*, *12*(06), pp.549-556.

Goswami, M., 2024. AI-based anomaly detection for real-time cybersecurity. *International journal of research and review techniques*, *3*(1), pp.45-53.

Haji, S.H. and Ameen, S.Y., 2021. Attack and anomaly detection in iot networks using machine learning techniques: A review. *Asian journal of research in computer science*, *9*(2), pp.30-46.

Heidari, A. and Jabraeil Jamali, M.A., 2023. Internet of Things intrusion detection systems: a comprehensive review and future directions. *Cluster Computing*, *26*(6), pp.3753-3780.

Hephzipah, J.J., Vallem, R.R., Sheela, M.S. and Dhanalakshmi, G., 2023. An efficient cyber security system based on flow-based anomaly detection using Artificial neural network. *Mesopotamian Journal of Cybersecurity*, *2023*, pp.48-56.

Hidayat, I., Ali, M.Z. and Arshad, A., 2023. Machine learning-based intrusion detection system: an experimental comparison. *Journal of Computational and Cognitive Engineering*, *2*(2), pp.88-97.

Iyer, K.I., 2021. From Signatures to Behavior: Evolving Strategies for Next-Generation Intrusion Detection. *European Journal of Advances in Engineering and Technology*, *8*(6), pp.165-171.

Jain, V. and Mitra, A., 2025. Real-time threat detection in cybersecurity: leveraging machine learning algorithms for enhanced anomaly detection. In *Machine Intelligence Applications in Cyber-Risk Management* (pp. 315-344). IGI Global Scientific Publishing.

Jamshidi, S., Nafi, K.W., Nikanjam, A. and Khomh, F., 2025. Evaluating machine learning-driven intrusion detection systems in IoT: Performance and energy consumption. *Computers & Industrial Engineering*, *204*, p.111103.

Khanan, A., Mohamed, Y.A., Mohamed, A.H.H. and Bashir, M., 2024. From bytes to insights: a systematic literature review on unraveling IDS datasets for enhanced cybersecurity understanding. *IEEE Access*, *12*, pp.59289-59317.

Kurniabudi, K., Purnama, B., Sharipuddin, S., Darmawijoyo, D., Stiawan, D., Samsuryadi, S., Heryanto, A. and Budiarto, R., 2019. Network anomaly detection research: a survey. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, *7*(1), pp.37-50.

Liu, Z., Shi, J. and Buford, J.F., 2024, February. Cyberbench: A multi-task benchmark for evaluating large language models in cybersecurity. In *AAAI 2024 Workshop on Artificial Intelligence for Cyber Security*.

Maddireddy, B.R. and Maddireddy, B.R., 2024. Neural network architectures in cybersecurity: optimizing anomaly detection and prevention. *International Journal of Advanced Engineering Technologies and Innovations*, *2*(1), pp.238-266.

Maseer, Z.K., Yusof, R., Bahaman, N., Mostafa, S.A. and Foozy, C.F.M., 2021. Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. *IEEE access*, *9*, pp.22351-22370.

Muneer, S., Farooq, U., Athar, A., Ahsan Raza, M., Ghazal, T.M. and Sakib, S., 2024. A critical review of artificial intelligence based approaches in intrusion detection: A comprehensive analysis. *Journal of Engineering*, *2024*(1), p.3909173.

Mvula, P.K., Branco, P., Jourdan, G.V. and Viktor, H.L., 2023. A systematic literature review of cyber-security data repositories and performance assessment metrics for semi-supervised learning. *Discover Data*, *1*(1), p.4.

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.22862295.

PM, V.P. and Soumya, S., 2024. Advancements in anomaly detection techniques in network traffic: The role of artificial intelligence and machine learning. *Journal of Scientific Research and Technology*, pp.38-48.

Roy, S., 2024. A comprehensive Survey on Network Traffic Anomaly Detection Using Deep Learning. *Preprints*.

Sarker, I.H., 2021. CyberLearning: Effectiveness analysis of machine learning security modeling to detect cyberanomalies and multi-attacks. *Internet of Things*, *14*, p.100393.

Tihanyi, N., Ferrag, M.A., Jain, R. and Debbah, M., 2024. Cybermetric: A benchmark dataset for evaluating large language models knowledge in cybersecurity. *arXiv preprint arXiv:2402.07688*.

Wang, S., Balarezo, J.F., Kandeepan, S., Al-Hourani, A., Chavez, K.G. and Rubinstein, B., 2021. Machine learning in network anomaly detection: A survey. *IEEe Access*, *9*, pp.152379-152396.

Yang, M. and Zhang, J., 2023. Data anomaly detection in the internet of things: A review of current trends and research challenges. *International Journal of Advanced Computer Science and Applications*, *14*(9).

Yaseen, A., 2023. The role of machine learning in network anomaly detection for cybersecurity. *Sage Science Review of Applied Machine Learning*, *6*(8), pp.16-34.

Yu, Z., Zeng, J., Chen, S., Xu, W., Xu, D., Liu, X., Ying, Z., Wang, N., Zhang, Y. and Yang, M., 2024. CS-Eval: A Comprehensive Large Language Model Benchmark for CyberSecurity. *arXiv preprint arXiv:2411.16239*.