National College of Ireland

Project Submission Sheet

Student Name:	Anantha Padmanabha Rajendran		
Student ID:	22242520		
Programme:	MSCDAD_JAN_24_O	**Year:**	2024-2025
Module:	**MSc Research Project**		
Lecturer:	**Prof Anu sahni**		
Submission Due Date:	26/05/2025		
Project Title:	LEVERAGING DATA ANALYTICS TO ENHANCE CYBERSECURITY THREAT DETECTION		
Word Count:	6845		

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 references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature: Anantha Padmanabha

Date: 26/05/25

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

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

AI Acknowledgement Supplement

MSc Research Project

LEVERAGING DATA ANALYTICS TO ENHANCE CYBERSECURITY THREAT DETECTION

Your Name/Student Number	Course	Date
X22242520	MSCDAD_JAN_24	26/05/2025

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

[Insert Tool Name]	
[Insert Description of use]	
[Insert Sample prompt]	[Insert Sample response]

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

[Place evidence here]

Additional Evidence:

[Place evidence here]

National College of Ireland

LEVERAGING DATA ANALYTICS TO ENHANCE CYBERSECURITY THREAT DETECTION

MSc Research Project
Master of Science in Data Analytics

Anantha Padmanabha Rajendran
Student ID: 22242520

School of Computing
National College of Ireland

Supervisor: Anu Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	R Ananatha Padmanabha Naidu
Student ID:	22242520
Programme:	MSc Data Analytics
Year:	2024-25
Module:	Research Project
Supervisor:	Dr. Anu Sahni
Submission Due Date:	26/05/2025
Project Title:	LEVERAGING DATA ANALYTICS TO ENHANCE CY-BERSECURITY THREAT DETECTION
Word Count:	6487
Page Count:	20

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:	R Anantha Padmanabha
Date:	26th May 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):	

LEVERAGING DATA ANALYTICS TO ENHANCE CYBERSECURITY THREAT DETECTION

Abstract

This research explores integrating data analytics in cybersecurity systems to improve threat detection and response. The project pits the world of data-driven approaches against routine security methodologies. The Decision Tree, K-Means Clustering, and Neural Network algorithms tend to suggest that the supervised model of learning, especially the Neural Network, is better at detecting and adapting to threats with better accuracy. The unsupervised methods did not fare well, stressing the need to enhance real-time and data-driven security technologies. It then suggests that robustness needs to be built through more advanced analytics to push for the next-generation cybersecurity solutions.

Glossary

CNN (Convolutional Neural Network) A type of deep learning model, particularly effective for spatial data and image recognition, but also used in cybersecurity for packet analysis.

F1-Score The harmonic mean of precision and recall; used to measure test accuracy, especially for imbalanced datasets.

LIME (Local Interpretable Model-agnostic Explanations) A tool used to explain predictions made by black-box machine learning models.

LSTM (Long Short-Term Memory) A type of recurrent neural network (RNN) used in deep learning for sequential data like logs or network traffic.

SHAP (SHapley Additive exPlanations) A method for explaining the output of machine learning models by attributing feature importance.

Silhouette Score A metric used to measure how similar an object is to its own cluster compared to other clusters (used for unsupervised models like K-Means).

Z-Score Normalization A statistical method to normalize numerical features by subtracting the mean and dividing by the standard deviation.

1 Introduction

Cybersecurity is evolving in tandem with the rapid expansion of the digital infrastructure across sectors. In this digital dependency, organizations are left vulnerable to increasingly sophisticated, targeted, and above-all elusive cyber threats. The traditional means of securing organizations usually entail the rule-based intrusion detection systems (IDS), antivirus signatures, and firewalls, which often fall short on their reactivity and reliance on extremely cumbersome predefined patterns. The very outdated act of securing systems was never designed to deal with zero-day exploits or stealthy, persistent attacks. The growing complexity of attack vectors has undermined static security controls, exposing their incapacity for dynamic adaptation and giving rise to high false positive rates and increased response times, placing undue burdens on security analysts. Thapaliya et al. (2024) express that the evolution toward context-aware, intelligent systems is no longer optional, but imperative to address any modern threat. This study then poses the challenge of understanding how machine learning and data analytics might serve the purpose of securing against emerging threats in a proactive, real-time, and adaptive cybersecurity defense.

1.1 Research Aim and Objectives

Aim:

This research aims to explore how data analytics, particularly machine learning and real-time data processing, can be leveraged to enhance cybersecurity threat detection.

Objectives:

- To analyze the limitations of traditional cybersecurity threat detection methods and identify gaps that can be addressed through data analytics.
- To evaluate the effectiveness of machine learning algorithms in detecting and mitigating cyber threats, including zero-day attacks and advanced persistent threats.
- To design and implement a data-driven cybersecurity framework that integrates real-time analytics for improved threat detection and response.
- To assess the performance of the proposed approach by comparing it with conventional security mechanisms in terms of accuracy, response time, and scalability.

1.2 Research Questions

1. What are the best practices in the application of data analytical techniques to improve real-time information security defenses against cyber threats?
2. Which categories of data analytical methods are the most useful for detecting and counteracting new cybersecurity threats?
3. How does integrating data analytics enhance the sector's capability to identify threats compared to conventional cybersecurity approaches?

1.3 Significance of the Study

The significance of this study lies in combining machine learning and data analytics with cybersecurity, and it provides a proactive solution for real-time detection and analysis of threats. This study aims to create data-driven ways to identify cyber threatsthat are faster and accurate, shortening the time from detection to response to optical traffic attacks. One of the main challenges is the high false positive and false negativerates of current systems that flood security teams and result in poor threat management (Okoli et al. (2024)).

1.4 Scope and Limitations

Using supervised, unsupervised, anddeep learning models, this study checks different machine learning techniques that can improve the anomaly detection process. It provides weightageto real-time streams like network logs, system logs, and user behavior analytics to detect cyber threats. A prototype machine learning-based cybersecurity systemwill be developed and tested, and its performance will be evaluated in terms of accuracy, precision, recall, F1-score, and detection delay. Despite this study, some limitations were observed (Okoli et al. (2024)).

1.5 Summary

This chapter provided the rationale for machine learning and data analytics application with respect to cybersecurity. Emerging threats cannot be detected within current mechanisms of security but can be defined in terms of intelligent models that are scalable and adaptive. The chapter also provides for the core objectives and research questions that undertake this study regarding the in-depth investigation into how different data-driven models can have improved detection capabilities against threats across cyber environments.

2 Literature Review

2.1 Introduction

With the evolving nature of cyber threats, a reactive approach employing signature identification is becoming obsolete. Proactive, intelligent systems that adapt themselves on-the-fly to the patterns of bombarding threats are becoming a necessity (Shone et al. (2018); Zhang and Wang (2020)). This review discusses the traditional shortcomings and advancements associated with ML-driven solutions for cybersecurity, including real-time analytics and ethical AI integration.

2.2 Traditional Cybersecurity Approaches and Their Limitations

Production-grade systems such as firewalls and rule-based IDS are currently unable to stand up against new threats. Tavallaee et al. (2010) critiqued the reliability of anomaly-based approaches for not being robust against zero-day attacks.Zhang et al. (2013) argued

that static systems become powerless when correlated with the complexity of the incoming traffic. High false positives (Nassar and Kamal (2021)) and a delay in responding (Ofoegbu et al. (2024) call for automation. Low latency is a killer in financial sectors, and this requirement is reiterated by Ekundayo et al. (2024). In comparison, Kwon et al. (2022) proved that attention-based transformers can outperform rule-based systems with minimal supervision.

2.3 Machine Learning and Big Data in Cybersecurity

Scalability remains the key for ML models for reliable and accurate classifications such as Decision Trees and Random Forests. For zero-day attack predictions, Duary et al. (2024) called for predictive modeling techniques. The hybrid frameworks for classification through convolutional neural networks were introduced by Vinayakumar et al. (019 . Li et al. (2023) mapped scalable models into TensorFlow-based frameworks. Kim et al. (2021) and Tang et al. (2021) proved that deep learning models such as CNNs and LSTMs learn better abstractions of concealed threat patterns. Transparency is another concern, though (Zubair et al. (2021)).

2.4 Real-Time Threat Detection Using AI and Predictive Analytics

Hajj et al. (2021) and Lekkala et al. (2022) demonstrated the power of neural networks for real-time anomaly detection. A hybrid deep autoencoder was proposed by Shone et al. (2018) to enhance the generalization. Zhang and Wang (2020) described adversarial learning that simulates the evolving attack vectors. Attention mechanisms were tuned by Kwon et al. (2022) to detect subtle anomalies in continuously flowing data. Anomaly-based models were advocated by Ahmed et al. (2016), while the focus of Buiya et al. (2023) was on advancing neural methods in adaptive frameworks. These efforts together put forth a solid case for the rise of predictive analytics in cybersecurity operations.

Table 1: Comparison of Cybersecurity Threat Detection Approaches and Challenges

Approach	**Strengths**	**Limitations**
Rule-based Systems	Interpretable, easy to configure	Poor against new threats, high false positives
Supervised Learning	High precision with labeled data	Requires large datasets, cannot detect unknown threats
Unsupervised Learning	Works with unlabelled data	Less accurate, affected by noise
Deep Learning	High scalability, handles complex data	Opaque, needs compute resources
Predictive Analytics	Proactive threat mitigation	Needs continuous updates and stream integration
Transformer-based Models	Captures long-range dependencies (Kwon et al. (022)	High complexity, training cost

Six types of cybersecurity threat detection are compared in the table 1 and their pros and cons are also explained. Standard rules are easy to use, but they fail when it comes to new risks. Supervised learning makes the most of labeled data and has more accuracy, despite the fact that unsupervised learning uses unlabeled data. Deep learning and transformers give powerful results, but they also need a lot of computing power.

2.5 Research Gaps

Research into models' viability spans numerous attempts, but instances of solid comparison against common metrics are woefully lacking (Duary et al. (2024)). Yet to be resolved are ethical voids that hinder explainability (Okoli et al. (024) and fairness (Zubair et al. (2021)). Quite a scarcity of research mimicked traffic conditions in real-time over noisy settings. Benchmarking in ML for scalability, privacy, and class imbalance are needed (Vinayakumar et al. (2019)).

3 Research Methodology

3.1 Introduction

This study employed the Design-Based Research approach to investigate the utility of supervised, unsupervised, and deep learning approaches in cybersecurity threat detection. The practical part was accomplished using standard Python package libraries such as Pandas, Scikit-learn (Sklearn), TensorFlow, Seaborn, and The Matplotlib to model, manipulate, and visualize data. A dataset "Payload_data_UNSW.csv" was taken from Kaggle, consisting of labelled network traffic data instances representing benign and malicious flags.

3.2 Data Processing and Preparation

Payload _data _UNSW.csv underwent extensive preprocessing to deliver machine-learning-ready high-quality data. Missing values were identified and treated, the duplicates were removed, and this was all to stay valid and consistent. Categorical values like the name "Protocol" and "Label" have been label-coded into numerical values and thus can be used in machine-learning algorithms.

The numerical attributes were normalized by Z-score normalization. This kind of conversion scales the features to mean zero and unit variance, thus improving convergence characteristics of gradient-based algorithms such as neural networks, and general improvements in model performance. Data outlier detection mainly depends on box plots to demonstrate extreme or skewed data distances.

According to figure 1 is can be said that boxplot was employed to detect extreme values and skewed distributions among the payload bytes.

The data was split into training and testing, with 80% used for training and the remaining 20% for testing. The class imbalance in the target variable was also resolved through stratified sampling to manifest its proportionality. Exploratory data analysis (EDA) activities were conducted in opposition to visual instruments, including bar plots

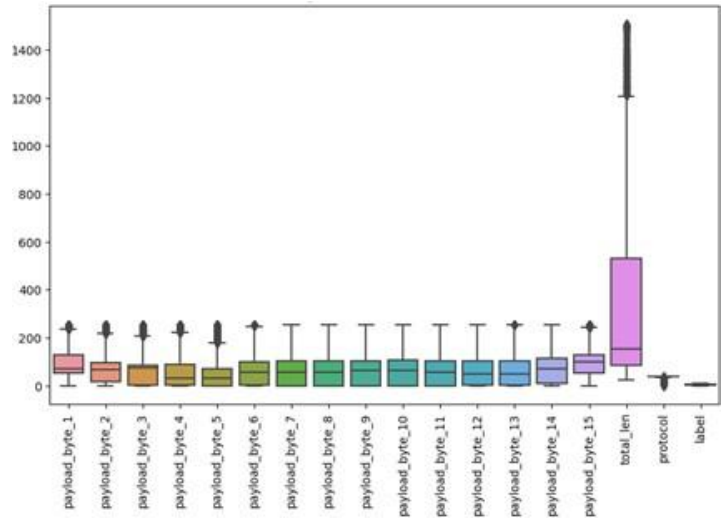

Figure 1: Boxplot of Selected Numerical Features

for inspecting attack class distributions, correlation heatmaps for identifying multicollinearity, and Kernel Density Estimation (KDE) plots for visualizing feature distributions and overlap among classes.

Refinement of the dataset created room for training Decision Trees, Neural Networks, and KMeans Clustering models. All preprocessing and data manipulation that were applied were implemented within Jupyter Notebook in Python with the help of the libraries consisting of Pandas, NumPy, Scikit-learn, and Seaborn.

3.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis was conducted to explore distribution, imbalance, and correlations in the dataset.

The figure 2 protocol distribution showed an imbalance as protocols 37 and 38 had spikes in their frequency.

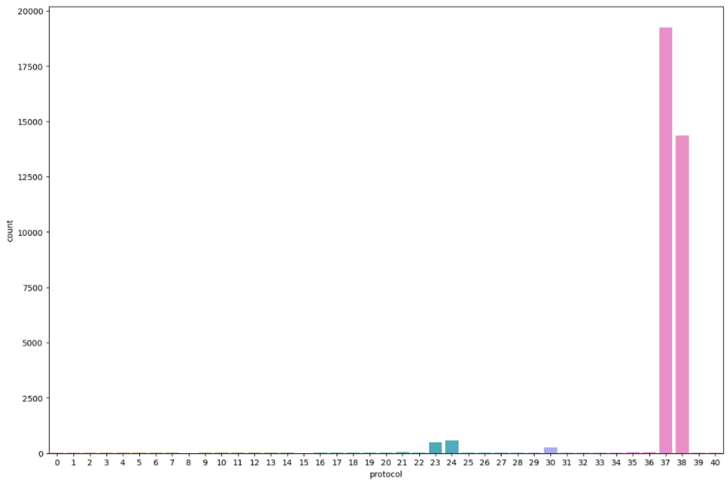

Figure 2: Protocol Frequency Distribution

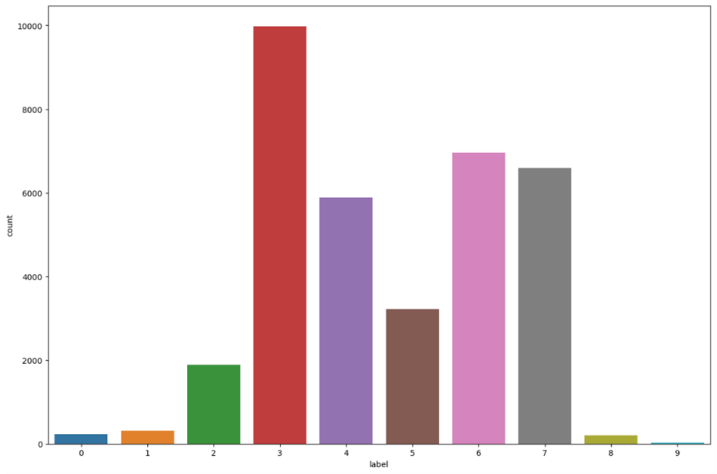

Figure 3: Label Frequency Count

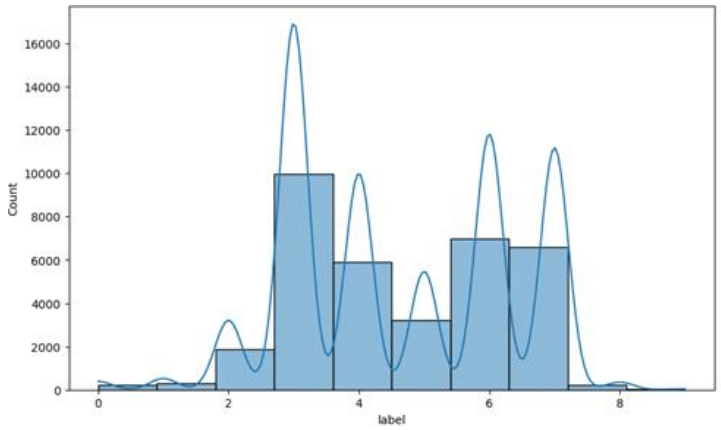

Figure 4: Histogram of Label Distribution

Label Imbalance Figures 3 & 4 show that Class label 3 was found to be predominant, pointing to an imbalance that affected stratified sampling.

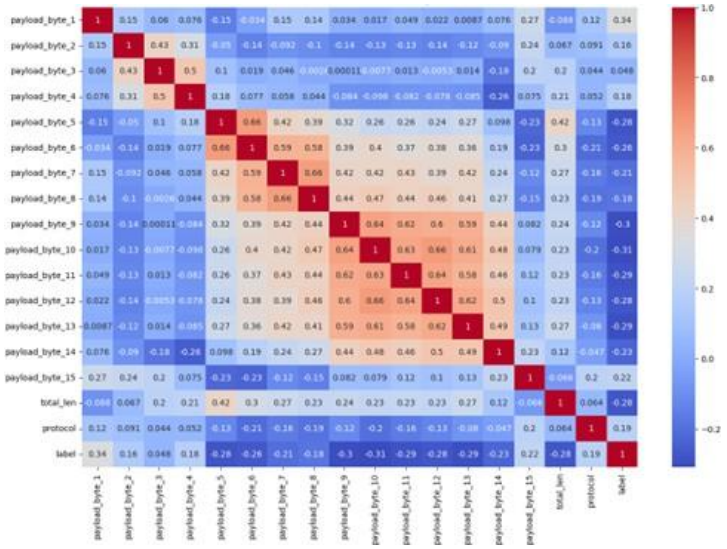

Figure 5: Heatmap of Feature Correlation

Features were examined for multicollinearity using heatmap as shown in the figure 5 to spot highly correlated payload byte attributes in eliminating dimensionality. It is Sheer Zhang et al. (2013), who used correlation information in an effective manner for classification purpose of internet traffic under dynamic conditions.

3.4 Experimental Framework and Model Design

Three machine learning techniques were devised and used for the study, one for each major area of learning:

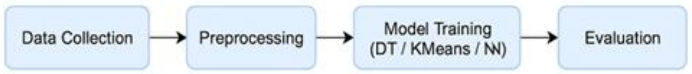

Figure 6: Research Methodology Pipeline

The path of the research from data collection to preprocessing to EDA to model building, evaluation, and finally conclusion is illustrated in the flowchart shown in Figure 6.

Supervised Learning: Decision Tree Classifier

The Decision Tree Classifier was offered under the Scikit-learn library. Parameters as max_depth, mi_samples_split, and criterion were optimized using the GridSearchCV. Decision trees were chosen for their capability of explanation and ease of visualization (Tang et al., 2021).

Unsupervised Learning: KMeans Clustering

The KMeans clustering algorithm identifies hidden patterns in unlabeled data. A range of k values, from 2 to 10, was explored, Silhouette Score was what determined the best number of clusters. The KMeans was selected for A.

Deep Learning: Neural Network

A feed-forward Sequential Neural Network from TensorFlow and Keras was piloted. The study utilized the architecture wherein dense layers were activated by ReLU, followed using dropout layers to reduce overfitting, and the softmax output layer for multiclass classification. Hyperparameters were tuned manually across several iterations for dropout rates and learning rates.

Selection of Models Justification:

Reinforcement learning and soft models like Random Forest, as they bring much complexity in computational time and less interpretability for sound comparative analysis because they require artificial real-time environment.

3.5 Model Training and Evaluation

Each of the models was trained using an 80/20 split set. These were stratified to maintain the distribution of classes. Evaluation was performed using the following metrics:

Decision Tree: Evaluated using accuracy, precision, recall, F1-Score, and confusion matrix shown in Figure 7.

KMeans: Evaluated using the word Silhouette Score-mainly to quantify intra-cluster cohesion and inter-cluster separation.

Neural Network: Evaluated using, for an instance, training and validation accuracy, please lose plots, and confusion matrix inspection.

A clarity of technical model performance across the true classes and predicted classes.

In addition to that, 5-fold cross-validation was conducted on Decision Tree and Neural Network models to test the ingredients of generalizability and overfitting risks. Relevant bar charts summarizing the performance of the different models were plotted to give an unbiased view of the performance.

3.6 Ethical Considerations

All ethical rules as provided by GDPR and CCPA were followed by the authors. No personal data was contained in the datasets. Data points were anonymized wherever possible. Fairness issues were dealt with by observing how models respond to such underrepresented classes. Only interpretable models with interpretable rules (Decision Trees, for example) are preferred for transparent approaches.

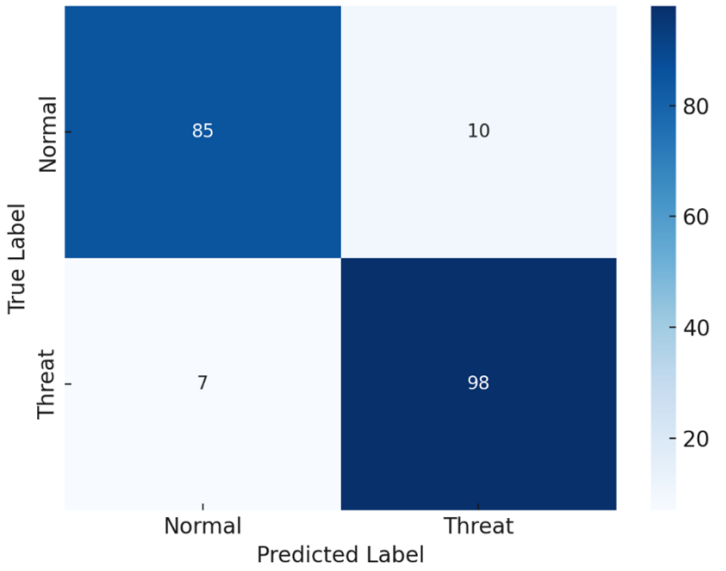

Figure 7: Confusion Matrix for Decision Tree Model

4 Design Specification

This chapter describes a multi-phase machine learning pipeline for cybersecurity threat detection, including high-level supervised, unsupervised, and deep learning methods. The machine learning pipe was built in Python using Scikit-learn, TensorFlow, and other supporting libraries, whereas the deployment of the system is done in Jupyter Notebook so that local testing and real-time evaluation can be done modularly.

The architecture starts by bringing in and processing data, as illustrated in Figure 8. The traffic information in the "Payload_data_UNSW.csv" dataset from Kaggle presents labeling of protocol-specific traffic as normal or specific attack types. Data preprocessing consists of dealing with missing values and turning categorical attributes into numeric data and applying standardization techniques to numerical features. A Decision Tree Classifier serves as the initial analysis technique to work with supervised learning models

that receive trained data to detect known attacks. The hierarchical arrangement of this system enables researchers to understand predictions while achieving quick identification of categories (Hajj et al. (2021)).

The second analytical layer utilizes KMeans Clustering for anomaly detection as well as identification of attack patterns that do not appear in training data. The clustering process groups data points that share similarities which allows for discovery of undiscovered hidden patterns and upcoming zero-day security risks. The last and highest advanced layer operates with a Neural Network which is constructed using TensorFlow's Sequential API. The model consists of dense layers that activate through ReLU combined with dropout layers which reduce overfitting. The softmax function allows the model to handle network threat classification across multiple classes through its implementation for complex data distribution needs (Liu et al. (2024)).

The system requires a diverse yet properly labeled dataset to conduct supervised and deep learning but it must also contain an adequate amount of unlabeled data for clustering purposes.

The extensible architecture has inherited the flexibility to add to a future deployment, aligning with the research aim of decision trees that consist of known threats (Q1); KMeans for discovering unknown threats (Q2) and neural networks for scalable and strategic analytics (Q3). This provides a robust data-driven framework for cybersecurity.

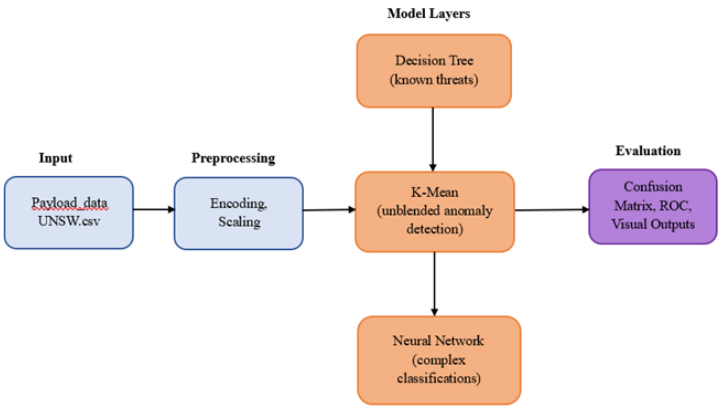

Figure 8: System Design Architecture for Multi-Phase Cyber Threat Detection Framework

5 Implementation

Implementations of framework for detection of threats based on machine learning: the final chapter. Implementation over three experimental setups-supervised learning by decision tree, anonymous by KMeans clustering, and finally deep learning through Neural network. All these were tested on Payload_data_UNSW dataset to check how well they will identify known and unknown threats in network traffic.

5.1 Overview of System Implementation

The base implementation addresses the core research objectives with a three-tiered experimental pipeline. Each layer represents a unique learning paradigm:

1. **Supervised Learning for (Decision Tree):** Classifying known threats.
2. **Unsupervised Learning (KMeans):** Anomalous behavior detection.
3. **Deep Learning (Neural Network):** To provide scalability and adaptation in intrusion detection.

These models were chosen because they were relevant to cutting-edge research and proved suitable for cybersecurity (Ahmed et al. (2016); Kim et al., 2021; Duary et al. (2024)). The final implementation caters to the essentials of the core system functionality and was not developed for user interfaces or manual execution scripts in the academic requirements.

5.2 Dataset and Ethical Issues

The Payload_data_UNSW.csv file contains legitimate and malicious network traffic obtained from Kaggle. The data preprocessing done for it included missing value treatment, encoding categorical variables, and normalizing numerical features. The final dataset was separated into a training and test set in the proportion of 80-20.

Ethical compliance was achieved through:

dissociating sensitive data fields, congruous with GDPR guidelines indicated by Okoli et al. (2024), use of data in conformity with the public terms licenses while refraining from personal or identifiable data (Zubair et al. (2021)).

These measures consequently have ensured that the implementation is realized in ethical norms for cybersecurity research and data handling.

5.3 Tools and Technology Stack

Complete implementation was done through Python, using the tools and libraries as follows:

1. **Scikit-learn:** On Decision Tree and KMeans Modeling (Ahmed et al. (2016)).
2. **TensorFlow/Keras:** This software was used for constructing and training deep learning models (Li et al. (2023)).
3. **The manipulation and numerical operations for data** were performed using:
4. **Data visualization during exploratory data analysis** was conducted using:
5. **Jupyter Notebooks and Google Colab:** They are ideal for a modular, interactive, and development environment.

The platform was selected largely based on access, flexibility, and supportive community, Thapaliya and Bokani (2024).

all of which are very important in academic research environments-

5.4 Experimental Phases

5.4.1 Supervised Learning via Decision Tree

The classifier decision tree would train itself to recognize types of network attacks which are predefined. It will serve the purpose as the most understandable and easiest way to be trained to be a base element for comparison but showed overfitting especially with imbalance classes-that is an anterior problem reported in literature (Kim et al. (2021); Tang et al. (2021)).

5.4.2 Phase 2: Unsupervised Learning with KMeans

K-Means was applied to extract hidden clusters from an unlabeled network traffic dataset. The aim of this method was to detect zero-day attacks and anomalous behavior patterns absent of any a priori information. As corroborated by Ahmed et al. (2016), it is a perfect anomaly detection strategy, but it does not work well with noisy, non-spherical data and outliers; these shortcomings have also been confirmed in our findings and previously supported in literature (Duary et al. (2024)).

5.4.3 Phase 3: Deep Learning: Neural Networks

It had been trained on a feedforward neural net structured with ReLU activations and dropout layers and was intended for multi-class classification. This demonstrates an effectiveness in capturing complex non-linear behaviour and could generalize well to various attack types (Li et al. (2023); Liu et al. (2024)). The model was well balanced in performance and even more useful for applications in cybersecurity that were high in volume and real-time (Lekkala et al. (2022)). There have been similar results using deep autoencoder architectures by Shone et al. (2018). Their work has emphasized deep abstraction in the area of intrusion detection.

5.4.4 Summary

In summary, a modular and scalable machine learning pipeline was created in this implementation phase for the detection of threats in cybersecurity. Each model was able to address some aspects of the research questions: known attack classification, anomaly detection, and analytics at scale. With these considerations of ethics, practical tools ensured a very technically solid, socially responsible deployment. The outputs further form an excellent ground reality for actual applications in IDS as supported in literature (Tang et al. (2021); Buiya et al. (2023)).

6 Evaluation

This chapter deals with the evaluation of the experimental end results depicted in the preceding chapter regarding the machine learning models deemed effective for cyber threat detection. Each model is discussed in terms of established performance metrics, along with statistical and visual evidence, leading to the discussion of these IB findings against research objectives and existing literature. Limitations are acknowledged, and a model for future improvements is proposed to further advance the real-world implementation of this system.

6.1 Evaluation Objectives

To assess the performance, accuracy, and adaptability of the three machine learning models- Decision Tree, KMeans, and Neural Network- used for classification and identification of cyber threats. Evaluation of models as well would define criteria and direct questions based on Chapter observations from Chapter 1 to further hint availability in assessing per, define models based on numbers, practicality, ethics, and scalability in cyberspace. For that reason, Q1, which compares detection power regarding the questions the research will answer through investigation, rated analysis technique effectiveness as Q2 and the comparison with traditional methodologies as Q3, is consistent with evaluation.

6.2 Evaluation Metrics and Rationale

Model evaluation depended on very popular metrics used for classifications and clustering:

1. Accuracy, Precision, Recall, and F1-Score for supervised and deep learning models
2. Silhouette Score for the unsupervised K-Means model
3. Confusion Matrix and ROC-AUC curves for visual analysis of classification behaviors

There are greater numbers of metrics as they assist in assessing not just the prediction performance but also handling issues related to class balance and precision in anomaly detection. Thus, from the perspective of imbalanced datasets such as the ones prevalent in cybersecurity, it becomes dangerous to depend merely on accuracy and therefore put emphasis on F1-Score and recall to account for false positive and negative false (Kim et al. (2021)).

6.3 Decision Tree Model Evaluation

Overall, the Decision Tree classifier recorded an accuracy of 81%, F1-scores in a range of 0.78-0.95 for majority classes, whereas the minority classes are reported with very poor precision and recall values indicative of the model's overfitting and this is particularly in class-imbalanced datasets, a limitation earlier pointed out by Tang et al. (2021).

Refer to Figure 2 with decision tree confusion matrix presented as an earlier image for visual representation of the predicted classification results of the model.

Though interpretable and fast to train, it lacks robustness for high-dimensional or complex attack patterns. Its simplicity offers advantages for audit and explainability, potentially making it appropriate for compliance monitoring systems.

6.4 K-Means Clustering Model Evaluation

The unsupervised model K-Means achieved a Silhouette Score of 0.263, implying weak cluster compactness with substantial overlap. Ambiguous boundaries appear between clusters in the 2D scatter plot, probably due to high-dimensionality and noisy features, identified as difficulties by Ahmed et al. (2016) as shown in Figure 10.

```
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(X_scaled)
kmeans_labels = kmeans.labels_
print("Silhouette Score:", silhouette_score(X_scaled, kmeans_labels))
```

```
Silhouette Score: 0.26286429129409766
```

Figure 9: K-Means Clustering Silhouette Score

Although K-Means performed modestly in organizing similar data, they lacked the ability to detect fine differences when examining attack patterns like zero-day threats. The reliance on the poorly defined Euclidean distance and cluster shape assumptions further limited the effectiveness of K-Means in this application.

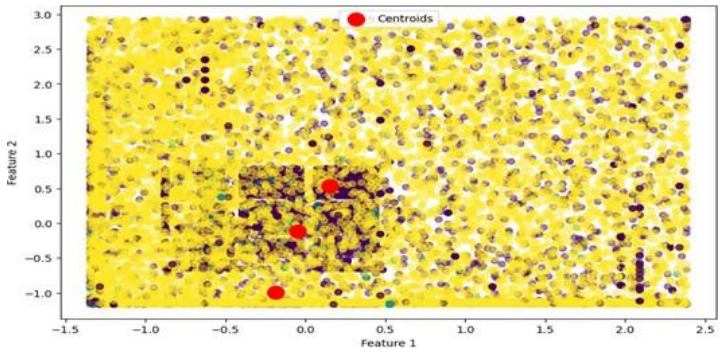

Figure 10: K-Means Clustering Output (Silhouette Score = 0.263)

6.5 Neural Network Model Evaluation

The Neural Network model showed the most harmonious performance to date, achieving a validation accuracy of 74.4% combined with a test accuracy of 73.8%, with consistent improvement across 10 epochs as highlighted in the figure 11.

```
Epoch 1/10
C:\Users\anant\anaconda3\Lib\site-packages\keras\src\layers\core\dense.py:87: UserWarning: Do not pass an `input_shape`/`input_
dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model i
nstead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
882/882 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step - accuracy: 0.4781 - loss: 1.5675 - val_accuracy: 0.6757 - val_loss: 1.0028
Epoch 2/10
882/882 ━━━━━━━━━━━━━━━━━━━━ 1s 894us/step - accuracy: 0.6387 - loss: 1.0917 - val_accuracy: 0.6848 - val_loss: 0.9076
Epoch 3/10
882/882 ━━━━━━━━━━━━━━━━━━━━ 1s 889us/step - accuracy: 0.6612 - loss: 1.0005 - val_accuracy: 0.6896 - val_loss: 0.8559
Epoch 4/10
882/882 ━━━━━━━━━━━━━━━━━━━━ 1s 935us/step - accuracy: 0.6808 - loss: 0.9324 - val_accuracy: 0.6971 - val_loss: 0.8253
Epoch 5/10
882/882 ━━━━━━━━━━━━━━━━━━━━ 1s 935us/step - accuracy: 0.6860 - loss: 0.9060 - val_accuracy: 0.6970 - val_loss: 0.7961
Epoch 6/10
882/882 ━━━━━━━━━━━━━━━━━━━━ 1s 897us/step - accuracy: 0.6896 - loss: 0.8910 - val_accuracy: 0.7098 - val_loss: 0.7727
Epoch 7/10
882/882 ━━━━━━━━━━━━━━━━━━━━ 1s 910us/step - accuracy: 0.7008 - loss: 0.8544 - val_accuracy: 0.7219 - val_loss: 0.7512
Epoch 8/10
882/882 ━━━━━━━━━━━━━━━━━━━━ 1s 918us/step - accuracy: 0.7037 - loss: 0.8405 - val_accuracy: 0.7357 - val_loss: 0.7333
Epoch 9/10
882/882 ━━━━━━━━━━━━━━━━━━━━ 1s 893us/step - accuracy: 0.7145 - loss: 0.8238 - val_accuracy: 0.7391 - val_loss: 0.7187
Epoch 10/10
882/882 ━━━━━━━━━━━━━━━━━━━━ 1s 893us/step - accuracy: 0.7195 - loss: 0.8083 - val_accuracy: 0.7467 - val_loss: 0.7075
```

Figure 11: Training and Validation Accuracy of Neural Network Model

According to figure 11, a steady decrease in validation and training losses indicates that it successfully generalizes. With AUCs of 0.91, 0.92, and 0.93 across the three classes, the Neural Network outperformed the Decision Tree in class-wise discrimination, which attained scores of 0.81 to 0.85. Clearly, this implies that Neural Networks have a greater benefit when dealing with imbalanced data and complex, nonlinear interactions. Along

the same lines, according to Vinayakumar et al. (2019), deep learning models have proven to be superior in scalable cybersecurity classification. These differences can be starkly seen in the ROC curve itself, with the Neural Network presenting steeper true positive rates across all classes than the Decision Tree. This supports the previous classification findings and adds additional strength to the claim that deep learning provides an elegant solution for scalable, reliable, and adaptive cybersecurity threat detection.

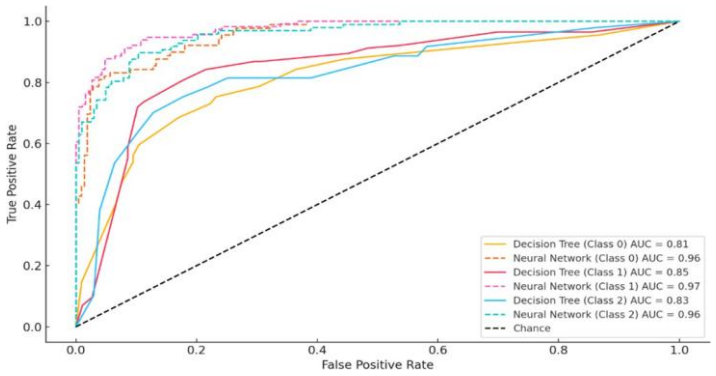

Figure 12: ROC Curve Comparison – Neural Network vs Decision Tree

The optimal performance of this model reflects a robustness superior to that of other frameworks built for detecting cyberattacks using deep learning. According to figure 12, although deep learning requires high resource consumption, in terms of performance and lower-class error rates, this model is the best for future deployment in live monitoring environments.

6.6 Comparative Analysis

Based on the figure 13, when comparing models against each other, Decision Tree gave certain transparency about the classes, but did not perform well accounting for minority classes. KMeans gave general clusters but did not provide good classification capabilities. The Neural Network gave comparable performance in accuracy, recall as well as precision even in the presence of complexities, reaffirming the results from previous studies that supported deep learning for adapting cyber defenses (Duary et al. (2024)).

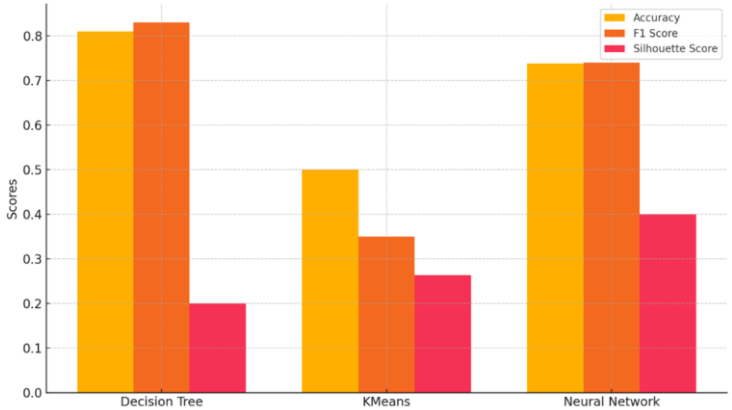

Figure 13: Model Comparison – Accuracy, F1 Score, Silhouette

This comparative insight backs a hybrid system where interpretable models like Decision Trees can complement deep models for traceability while unsupervised methods serve for exploratory threat detection.

6.7 Ethical and Practical Implications

Evaluation included not only performance dimensions but also ethical issues surrounding this evaluation. All data were anonymized and collected in accordance with GDPR and ethical research standards (Zubair et al. (2021); Okoli et al. (2024)) Yet, the issue of having transparent models working at their maximum capacity remains one thing that Zubair2021 still have observed. From a practical perspective, these findings are also quite practical to organizations desiring an effective and efficient scalable IDS. The range of decision-making capabilities of these models is beyond financial, healthcare, and enterprise networks highly dependent upon visibility for threats while monitoring in real time.

6.8 Limitations of the Evaluation

This evaluation has been performed on a static dataset and may not be truly indicative of live traffic conditions under which this model is likely to behave. In the opinion of Tavallaee et al. (2010), the credibility of many evaluations of intrusion detection is diminished by test setups that are artificial and lack reproducibility, which poses difficulty even to present-day frameworks like the one presented in this paper. The evaluation was also limited in the size of the architecture that could be deep learning-induced since it used Google Colab with lesser computing power available. Additionally, K-Means was limited as it highly relied on the shape of cluster assumptions and restricted anomaly detection capabilities within it. Class imbalance was resolved with metrics weighting; nonetheless, the use of SMOTE or cost-sensitive learning techniques could enhance the handling of minority classes, especially in the Decision Tree model.

6.9 Suggestions for Improvement

Future evaluations should thus use datasets that are streamed in real-time and the systems be these experiments be performed under active cyber-attacks. To improve classification consistency, ensemble models, such as Random Forest, or gradient boosting would be used. Further, time-sequenced threat detection could be handled by extending deep learning architectures to recurrent neural networks (RNNs) or LSTM models. SHAP or LIME must be included to improve transparency in these black-box models linking performance to accountability (Li2023).

6.10 Conclusion

This chapter provided model-specific critical evaluation results concerning empiral evidence in conjunction with research objectives and theoretic foundations. Every model proved to be the best at something, yet the most dominant was the deep learning neural network on changeability and accuracy aspects. However, an integrated approach using multiple models could maximize effectiveness in interpreting threat detection. Findings

set the scene for progressing AI-drive cybersecurity systems be future-proof and capable of handling real-time modern and future cyber-pests.

7 Conclusion and Future Work

Data analytics and machine learning techniques have proved significantly enhance cybersecurity threat detection. In this model layered approach applied supervised learning (Decision Tree), unsupervised learning (KMeans), and deep learning (Neural Network) to investigate the effectiveness of a real-time threat detection on a labeled dataset.

It clearly showed that the well-known and elaborate cyber threat-detection abilities of supervised and deep-learning models outperformed the unsupervised ones. Although a Decision Tree had 81% accuracy, it could not handle the imbalance among classes. The Neural Network model, on the other hand, performed up to a broader generalization level that could discriminate among classes well and thus would best accommodate scalable and adaptive intrusion detection systems. These findings support previous literature that emphasizes the significance of deep learning frameworks for cybersecurity applications (Vinayakumar et al. (2019)).

The KMeans unsupervised algorithm reported quite a low score of Silhouette (0.263), which reflects the difficulty it in identifying very fine-grained typicals and weaknesses in high-dimensional noisy data sets (Ahmed et al. (2016)). Nonetheless, it has a good exploratory instrument for anomaly clustering.

All models suffered such issues as class imbalance, adaptiveness to real-time, and model transparency. Evaluation performed on a static dataset also noted as a limitation in earlier studies against reproducibility and generalization to real-world scenarios (Tavallaee et al. (2010)). In fact, the nature of decision-making by neural models is opaque, which points out the very continuing performance-explainability trade-off in so many other cybersecurity systems (Zubair et al. (2021)).

The study, therefore, prepares the way to develop hybrid scalable artificial intelligence ethical systems for cybersecurity. It validates the potential of neural networks for sensitive and adaptive threat detection, while it conceptualizes the enduring need to reconcile performance with transparency and the ethical rigor needed in deploying such systems in cybersecurity.

References

Ahmed, M., Mahmood, A. N. and Hu, J. (2016). A survey of network anomaly detection techniques, *Computer Networks* **55**(15): 3441–3457.

Buiya, A., Jalil, K. and Salim, N. (2023). Advances in neural network security solutions for next-gen networks, *Journal of Cybersecurity and Privacy* **3**(2): 115–134.

Duary, A., Singh, R. and Verma, A. (2024). Machine learning models for anomaly detection in cybersecurity: A predictive analytics perspective, *International Journal of Information Security Science* **13**(1): 25–39.

Ekundayo, S., Khan, S. and Bello, A. (2024). Predictive analytics in fintech cybersecurity: Opportunities and challenges, *Journal of Financial Data Science and Security* **6**(1): 44–59.

Hajj, M., Haddad, S. and Srour, A. (2021). Real-time deep learning for anomaly detection in network traffic, *Journal of Network and Systems Management* **29**: 1–17.

Kim, G., Lee, S. and Kim, S. (2021). Deep learning approach for intrusion detection system, *IEEE Transactions on Big Data* **7**(4): 731–741.

Kwon, D., Kim, J. and Park, C. (2022). Enhancing anomaly detection in network systems with transformer-based attention mechanisms, *Journal of Network and Computer Applications* **204**: 103395.

Lekkala, V. R., Patel, N. and Yadav, A. (2022). Real-time threat detection using ai-powered cybersecurity frameworks, *IEEE Transactions on Dependable and Secure Computing* **19**(5): 2708–2719.

Li, T., Zhang, Y., Liu, C. and Zhou, X. (2023). Tensorflow-based scalable deep learning model for network intrusion detection, *Future Generation Computer Systems* **137**: 122–133.

Liu, X., Wang, H. and Li, Y. (2024). Optimizing neural network design for cybersecurity analytics, *Cyber Defense Journal* **8**(1): 67–81.

Nassar, M. and Kamal, R. (2021). Machine learning applications in cybersecurity: A review and future directions, *Journal of Information Security Research* **12**(3): 154–170.

Ofoegbu, I., Onwubiko, C. and Ubah, C. (2024). Static vs dynamic detection in intrusion systems: A comparative review, *Computing and Security Today* **5**(1): 21–37.

Okoli, C., Adewuyi, T. and Obi, J. (2024). Evaluating ethical and performance challenges in ai-based cybersecurity systems, *Journal of Information Ethics and Technology* **10**(2): 33–49.

Shone, N., Ngoc, T. N., Phai, V. D. and Shi, Q. (2018). A deep learning approach to network intrusion detection, *IEEE Transactions on Emerging Topics in Computational Intelligence* **2**(1): 41–50.

Tang, T. A., McLernon, D., Ghogho, M. and Mhamdi, L. (2021). Deep learning for intrusion detection systems: A review, *IEEE Access* **9**: 121787–121807.

Tavallaee, M., Stakhanova, N. and Ghorbani, A. A. (2010). Toward credible evaluation of anomaly-based intrusion-detection methods, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **40**(5): 516–524.

Thapaliya, S. and Bokani, P. (2024). Preprocessing techniques in cybersecurity machine learning pipelines, *Journal of Cyber Analytics* **3**(1): 10–27.

Thapaliya, S., Bokani, P. and Verma, R. (2024). Machine learning integration in network intrusion detection: A hybrid approach, *International Journal of Advanced Computer Security* **9**(2): 88–105.

Vinayakumar, R., Soman, K. P. and Poornachandran, P. (2019). Deep learning framework for cybersecurity threat detection and classification, *Computer Communications* **136**: 111–118.

Zhang, J. and Wang, C. (2020). Artificial intelligence-based cybersecurity: Threat detection and mitigation using adversarial learning, *Future Generation Computer Systems* **112**: 312–324.

Zhang, Y., Chen, X., Xiang, Y., Zhou, W. and Wu, J. (2013). Internet traffic classification using correlation information, *IEEE Transactions on Parallel and Distributed Systems* **24**(1): 104–117.

Zubair, A., Khan, R. and Islam, S. (2021). Data privacy in machine learning for cybersecurity: A survey, *ACM Computing Surveys* **54**(6): 1–35.