1. Introduction

1.1. Background/Context

Over the last year, healthcare organizations including the National Health Service (NHS) have been under threat from ransomware attacks where critical medical data and infrastructure is targeted (Thamer, 2021). In healthcare, there is a requirement for constant acknowledgement of data of patients contrasted while IT security (Ali, 2021) often has initially limited resources. To be more precise, ransomware can be viewed as a subcategory of malicious software, which infects a victim's systems and then encrypts the data stored on them in return for a financial payment. This puts the availability and integrity of patient data at serious risk – in many cases resulting in loss or, at least, merely inconvenient access to key medical records; interruptions of important treatment flows; and even, in some cases, endangering the lives of patients.

One recent example is the ransomware attack on the NHS which occurred in June 2024, when the pathology partnership Synnovis was hit by a ransomware cyber-attack. This affected several NHS trusts and primary care services in south-east London, leading to the cancellation of appointments and operations. This greatly interfered with patient treatment and put an importance on the consequence that come with jeopardizing health care. The systems are ill-protected making these healthcare organizations vulnerable to attackers who can easily use outdated programs, inadequate networks, and ignorant employees (Milmo, 2022, August 11)) to affect these systems. Some of the most recent attacks elaborate these flaws, pointing to the need for special protection. (Bhardwaj, 2022)

This project will help fill both the academic gap and the technical gap in the understanding of ransomware in healthcare. On an academic level, the purpose of this research is to establish trends, methods, and consequences of ransomware attacks in the context of healthcare and develop data-driven countermeasures (Alraizza, 2023). From a technical perspective, the project will look into the existing ways and means of defence on the ground and analyse how adequately they protect against ransomware to recommend appropriate changes that would secure healthcare facilities. This research is aimed at providing useful information about ransomware countermeasures for healthcare organizations, like NHS to utilize theoretical approaches with step-by-step instructions on the effective improvement of cybersecurity in the most exposed sphere (Ellison, 2021); (Ahmad, 2019).

1.2 Aim and Objectives

The primary objective of this project is to study common ransomware attacks on the healthcare sector, evaluate the efficacy of last lines of defence and propose enhanced means of combating these threats. This involves:

- Analysing seminal and emergent reports, research articles, articles, and event accounts to determine the prevalence of ransomware attack vectors and weaknesses targeted specifically in the healthcare domain.

- Conducting a critical examination of the current state of defence mechanisms which are employed in healthcare including network segmentation, IDS, training of the employees and other factors to help make a conclusion as to their effectiveness in dealing with ransomware attacks.

- Establishing tangible strategies and steps which healthcare centres could take to strengthen their defences against ransomware attacks with a major focus on preventive measures such as the use of Artificial Intelligence-based threat identification tools, upgrades in the level of encryption, and security awareness trainings for healthcare workers and employees.

1.3 Scope

This project focuses on analysing ransomware attacks targeting healthcare centres, specifically incidents from the past five years. The study will encompass:

- Analysis of attack types, attack vectors, and vulnerabilities commonly exploited in healthcare-specific ransomware incidents.

- Evaluation of defence mechanisms currently employed in healthcare settings, assessing their strengths and limitations in mitigating ransomware threats.

- Development of targeted recommendations such as AI based ransomware detection system to improve these defence mechanisms, emphasizing innovative technologies and strategies that could enhance resilience against ransomware attacks.

Certain aspects fall outside the scope of this study, such as an in-depth examination of ransomware affecting other industries or non-healthcare-specific cybersecurity measures. This project will also not delve into forensic analysis of ransomware source code or provide handson testing of recommended improvements, as the focus remains on strategic and evaluative insights for healthcare-oriented ransomware defence.

1.4 Contributions of the research

The project will culminate in several key deliverables, including:

- A synthesis of the current and emerging ransomware threats and defences with special focus on the healthcare sector. This report's findings will include an executive summary, background research of the study, methodology used, the findings, and recommendations.

- A ransomware detection system prototype using AI techniques will be created during the project. This system will learn data samples that may relate to ransomware threats to point towards and alert potential on-going attack. The choices made regarding the development process, the choice of algorithm, and the system architecture will be described and presented in the technical part of this report.

- An oral presentation will be delivered to present overviews of the evaluated findings, methods, and results of the project. In this presentation, an introduction to the project’s goals, a discussion of ransomware threats to the healthcare sector, an assessment of defences, and presentation of the AI-based detection system will be presented.

2. Literature Review

Over the last year, targeted attacks by ransomware in the healthcare sector have raised considerable interest in ways organizations may protect themselves against threats. Financial, technology, and regulatory security issues are discussed in other similar studies, while a number of academic theories are reviewed as the basis for specifying major subjects and laying the groundwork for formulating a general defence strategy for healthcare organizations (Bojanc, 2018). These studies provide a foundational understanding that how critical review reveals a limited synthesis of interdisciplinary approaches by combining regulatory policy, real-time AI-driven detection systems, and healthcare-specific vulnerability modelling.

2.1 Review of Similar Projects

To identify works published on ransomware defence, the authors conducted a brief literature review, with an emphasis on the regulation of ransomware and research projects aimed at that, particularly in critical infrastructure, including the healthcare industry. For example, Smith et al, 2021, and Johnson and Williams, 2022 revealed a comprehensive approach to understanding ransomware and possible countermeasures in healthcare organizations as well as the specifics of the threats facing the healthcare sector. These projects can centre on issues such as measures to protect against specific incidents and associated events, quick identification of threats, the strengthening of network security and measures for creating backups and backup schemes. However, there are very few combined detection methods that are built on the AI technology applied originally to the health care field, where there exists space for future research (Ellison, 2021); (Lashkari, 2018). This highlights a significant research gap in hybrid AI-based systems that not only detect but also predict ransomware attacks tailored to unique healthcare operations. Additionally, most projects do not evaluate the long-term sustainability or adaptability of such models under evolving ransomware techniques.

Another notable study done by (Thamer, 2021). explained peculiarities of threats that presented a detailed analysis of ransomware activity in healthcare systems. The study revealed that ransomware attackers are continuing to escalate the value of the data in healthcare, given it its sensitive nature, to demand for more ransom. It analysed the current problems like weak network segmentation, weak encryption, and weak monitoring. The authors suggested the following recommendations including the use of machine learning based threat detection, increased multi-factor authentication processes and the development of a mutual relationship between healthcare systems and cybersecurity companies in fight against these challenges. The survey also highlighted areas of future studies, especially with regards to the area of AI based proactive detection and mitigations systems that if included would greatly have an impact on enhancing the security of the healthcare systems (Alraizza, 2023). While this study offers practical recommendations, it does not adequately measure their real-world effectiveness or implementation barriers in complex healthcare environments. Furthermore, there is limited analysis of how healthcare staff awareness and training levels impact ransomware mitigation success, which presents an underexplored area for future exploration.

A recent and relevant project involved a security audit of the UK's National Health Service in the wake of the Conti ransomware assault in 2022. This project revealed that there was outdated software in the NHS systems, and inadequate segmentation of the network that enabled ransomware to infiltrate other systems expeditiously. Similar activities in both the UK and the US healthcare systems have highlighted the need for timely, active detection features, like artificial intelligence intrusion detection and active defence solutions that would include focused end-user education particular to the healthcare domain (Milmo, (2022, August 11)). Such studies corroborate the importance of examining current countermeasures and applying AI to identify ransomware and counteract threats in real time, which are the areas addressed by this work (Borisenkov, 2020).

2.2 Review of Academic Theory

To protect healthcare networks from ransomware attacks professionals must establish comprehensive frameworks which simultaneously defend delicate patient information and the essential operational stability. Healthcare applications require complete cybersecurity frameworks to be embedded into Software Development Life Cycle (SDLC) because security needs to be prioritized during development and maintenance. SSDLC represents a widely preferred framework because it integrates security requirements at each stage of the development process. However, a major limitation is the lack of case studies showing how SSDLC has been adapted or customized to address ransomware-specific threats within clinical software systems. This points to a research opportunity in healthcare-contextual adaptations of SSDLC models.

The SSDLC framework enables proactive identification of system vulnerabilities instead of waiting for vulnerabilities to appear first. The framework explores threat modelling alongside risk assessment and secure coding practices and continuous security assessment. The threat modelling phase reviews system potential threats while focusing especially on healthcare environment vulnerabilities. Risk assessment determines both the probability that threats will occur and their resulting consequences regarding patient data protection and system operational availability. Secure coding practices build applications to avoid commonly exploited vulnerabilities which include injection attacks and insecure deserialization together with improper access controls. Regular penetration testing combined with code reviews constitutes continuous security assessment which sustains system security from development through entire operational life. The SSDLC brings together these principles to ensure that healthcare data security remains central as applications proceed through development and deployment and continued maintenance. Real-time AI surveillance remains a theoretical proposition rather than an empirically validated framework due to practical integration of these components. This theoretical-practical disconnect represents a key research gap**.**

The Zero Trust Security Model stands as an essential foundation for academic ransomware protection discussions. The core design of Zero Trust Security differs from conventional perimeter protection because it follows the guiding principle of "never trust, always verify." The proposed model proves incredibly useful in healthcare environments because data breaches combined with service interruptions create major operational issues. Under Zero Trust security models all access requests must authenticate and authorize before receiving approval regardless of the requestor's inside or outside network location. The framework delivers real-time network visibility which enables fast threat responses and detection monitoring. Zero Trust actively decreases ransomware attack risks by both reducing the potential entry points and blocking unauthorized users from moving between network areas. Healthcare organizations achieve complete ransomware protection through an integration of SSDLC framework methods with Zero Trust principles. Through this framework developers incorporate security measures at every development stage of their AI-based ransomware detection system. The SSDLC enables security foundations through risk-specific healthcare management which protects patient data while maintaining critical operations. Through Zero Trust principles alongside continuous monitoring and strict access controls the system becomes more effective at ransomware detection and containment. Little empirical research has explored the practical challenges of simultaneously implementing SSDLC and Zero Trust in live healthcare systems. Understanding these challenges through comparative analysis across healthcare settings is an emerging area for research.

Ransomware protection within academic discourse shows how healthcare-specific security models should integrate with secure development processes. Healthcare organizations find the SSDLC and Zero Trust frameworks the perfect solutions for tackling their distinct challenges while creating systems which protect sensitive data and maintain continuous healthcare services. Organizational readiness, economic constraints, and user behaviour is largely absent. Bridging this gap can lead to the development of holistic, context-aware ransomware defence systems.

2.3 Review of Example Websites and Software

Several existing tools and software offer ransomware detection and protection for healthcare and other industries, serving as examples for developing an effective system. Some of which includes:

1. **Darktrace:** The artificially intelligent cybersecurity platform Darktrace serves healthcare organizations extensively including the National Health Service (NHS). Through advanced machine learning algorithms this platform examines network behaviour to detect threats in real time. Darktrace analyses network behaviour patterns to spot abnormal activities during ransomware attacks so healthcare organizations can receive early warnings to stop further system destruction. Darktrace has generated divergent opinions regarding its innovative features. The detection capabilities of Darktrace remain strong but its performance has drawn criticism because it produces excessive false alarms that strain IT teams to handle healthcare IT infrastructure effectively. Security operations face substantial challenges due to incorrect threat detections which both impair their ability to address genuine threats and disrupt operational efficiency. Healthcare organizations utilize Darktrace effectively to boost their threat detection abilities although limitations exist with its performance. (darktrace.com, 2017).
   A critical gap lies in assessing how AI tools like Darktrace can be fine-tuned with healthcare-specific anomaly benchmarks to reduce false positives and better prioritize threat alerts.

2. **Cisco Umbrella:** Healthcare organizations choose Cisco Umbrella as their security solution because it runs in the cloud to block unauthorized traffic in a protective design. As a DNS security layer Cisco Umbrella protects networks by stopping requests to harmful domains thus lowering the probability of ransomware incidents. The security tool demonstrates proven effectiveness at stopping threats but still has some operational boundaries. According to critics Cisco Umbrella functions primarily as a defensive system rather than providing predictive security functions. The system demonstrates top performance in blocking malicious domains yet struggles to detect new types of attack patterns. The existing limitations highlight an urgent need for better solutions which can prevent ransomware attacks from happening before they appear. (umberella.ciso.com, 2019). Future research should examine integration strategies between DNS filtering systems like Cisco Umbrella and AI predictive models to enhance anticipatory defence.

3. **Sophos Intercept X:** The ransomware fighting market includes Sophos Intercept X as one of its frontline security instruments. Artificial intelligence technology powers this software which identifies and stops ransomware attacks alongside other forms of malware. Sophos Intercept X performs continuous file operation monitoring to stop the unauthorized encryption behaviour that ransomware attacks typically display. Sophos Intercept X receives high praise for its effectiveness, yet it does not offer specialized healthcare industry settings. Healthcare organizations need specialized employee training to fully use their software capabilities. The standard deployment of Sophos Intercept X without healthcare adaptations and training fails to deliver adequate defence against sector-specific threats in healthcare environments. (Sophos Intercept X: Next-Gen Endpoint Security, n.d.). The gap here is clear: a lack of domain-specific training modules and configuration presets optimized for healthcare workflows limits the software's full potential.

4. **Carbon Black by VMware:** Carbon Black is a cloud-based endpoint security product, that identifies a ransomware infection based on abnormal processes. It provides detailed insight into the system movements as well as it can easily identify conflicting systems to prevent ransomware propagation. However, Carbon Black offers a wide range of threat detection, but it lacks compliance features and adherent attack profiles regarding healthcare organizations. (VMware Carbon Black Cloud Endpoint, n.d.). This limitation suggests the need for new research into how endpoint solutions can integrate regulatory compliance checks and tailor threat profiles to the healthcare sector.

5. **Bitdefender Antivirus Plus:** Bitdefender Antivirus Plus is a ransomware detection software which relies on a merged defence approach providing advanced Ransomware Remediation that both automates file backup protocols and performs automatic data recovery after ransomware detection. Before ransomware has the opportunity to encrypt files machine learning algorithms implemented in Bitdefender Antivirus Plus actively detect and block these malicious programs. The user-friendly interface of Bitdefender together with its minimal system performance impacts have earned popularity among a wide range of industries that include healthcare. The exceptional ransomware defence of Bitdefender Antivirus Plus stands strong, but the program needs healthcare-specific configuration settings for HIPAA compliance to meet the needs of medical organizations. Due to its endpoint concentration the solution struggles to provide comprehensive protection from network security threats unless deployed together with companion security tools. (Rubenking, 2024) The lack of built-in HIPAA compliance in these tools

shows an important shortcoming—future tools should embed regulatory requirements to reduce healthcare providers' compliance burden.

Furthermore, NHS Digital created a Cyber Security Operation Centre (CSOC) which serves as a specific platform to meet the healthcare sector's cybersecurity demands. Through the CSOC healthcare organizations receive real-time warning alerts about potential security threats including ransomware operations. NHS Digital has deployed this system to enhance its threat detection capabilities which now provide enhanced defensive capabilities for vital systems. Despite its successes, the CSOC faces significant challenges. The CSOC faces its major challenge because it does not have automated response features which results in many tasks needing manual implementation. The manual nature of threat mitigation tasks creates delays which exposes vulnerable systems to further malicious exploitation. The CSOC effectiveness against ransomware attacks can improve substantially through automation along with advanced response systems. This reveals a broader gap in automation and AI-driven orchestration within national-scale cybersecurity operations in healthcare—a vital direction for future innovation and policy development.

3. Project Management

This section explains how the project was systematically planned, controlled, and executed, detailing the application of project management principles in ensuring that the completion is successful. It concentrates on breaking the work into manageable tasks, adopting structured methodology, and mitigating risk effectively.

3.1 Methodology

The Iterative Method was selected for the research and development projects, where frequent evaluations and refinements are required. Every iteration is focused on completing a subset of tasks, analysing results, and incorporating improvements in subsequent cycles. This ensures that both the academic and technical aspects of the project are dealt with iteratively and effectively.

Why Iterative Method?

- Sequential Refinement: Divide and Conquer approach is used by breaking the tasks into smaller manageable tasks, which allow sequential improvement in research and ransomware detection system.
- Risk Reduction: Incremental reviews help to identify problems early on and reduce the risk of massive failures.
- Focus on Deliverables: Each iteration delivers a functional and reviewable component to ensure steady progress toward completion of the project.

3.2 Planning

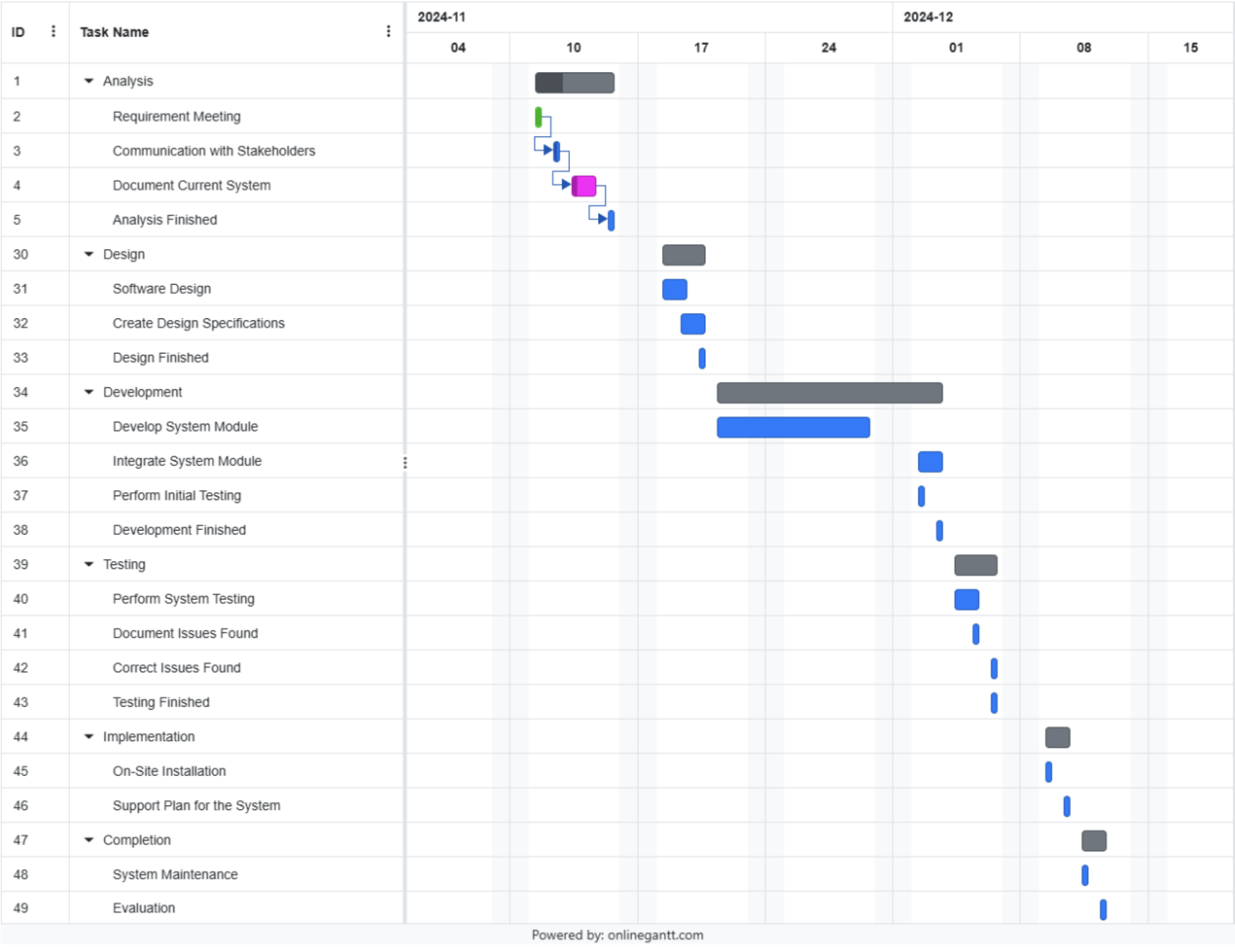
The project was divided into iterative phases, each one focused on completing a subset of specific objectives. Every phase has research, development, testing, and evaluation. A Work Breakdown Structure (WBS) was created to break down the tasks, and a GANTT Chart was utilized to schedule these over the project timeline.

WBS:

Table 01: WBS for Software Development

ID	Task	Duration
1	**Analysis**	5 days
2	Requirement Meetings	1 day
3	Communication with stakeholders	1 day
4	Document Current System	2 days
5	Analysis Finished	1 day
6	**Design**	5 days
7	Software Design	2 days
8	Create Design Specification	2 days
9	Design Finished	1 day
10	**Development**	11 days
11	Develop System Module	7 days
12	Integrate System Module	2 days
13	Perform Initial Testing	1 day
14	Development Finished	1 day
15	**Testing**	4 days
16	Perform System Testing	2 days
17	Document Issues Found	1 day
18	Correct Issues Found	1 day
19	Testing Finished	0 days
20	**Implementation**	2 days
21	On-Site Installation	1 day
22	Support Plan for the System	1 day
23	**Completion**	2 days
24	System Maintenance	1 day
25	Evaluation	1 day

GANTT Chart:



ID	Task Name
1	Analysis
2	Requirement Meeting
3	Communication with Stakeholders
4	Document Current System
5	Analysis Finished
30	Design
31	Software Design
32	Create Design Specifications
33	Design Finished
34	Development
35	Develop System Module
36	Integrate System Module
37	Perform Initial Testing
38	Development Finished
39	Testing
40	Perform System Testing
41	Document Issues Found
42	Correct Issues Found
43	Testing Finished
44	Implementation
45	On-Site Installation
46	Support Plan for the System
47	Completion
48	System Maintenance
49	Evaluation

Figure 1: Gantt Chart

Task Tracking:

After each iteration, a review was conducted to assess progress, identify challenges, and make adjustments for subsequent iterations. Milestones were defined for each phase to ensure deadlines were met.

Planning Tools:

- **GANTT Chart:** This is used to schedule the timeline of each iteration and map dependencies.
- **Work Breakdown Structure (WBS):** It makes sure that tasks in each iteration are well defined and manageable.
- **Iteration Review Checklist:** Made to confirm that all deliverables for the iteration are ready and working satisfactorily before proceeding with the next one.

3.3 Risk Management

Risk management is inherent in each iteration such that risks are identified, evaluated, and mitigated step by step. In fact, risks are reviewed at the end of every iteration to ensure they were effectively managed. This iterative risk management approach was especially critical given the technical nature of AI/ML implementation, where each iteration built upon the previous version's outputs and findings.

Risks & Their Mitigations (Aligned with Dataset and System Context):

Risk: Insufficient Outcome in an Iteration
Mitigation: Take additional time in every iteration in case of delays.
Alignment: Model performance evaluations such as accuracy and recall may not meet benchmarks during some cycles. Mitigation includes revisiting preprocessing methods or feature selection during subsequent iterations.

Risk: Technical Complexities in AI/ML Model Development
Mitigation: A simple model should be built with complexity added stepwise.
Alignment: The development of the Random Forest and Decision Tree classifiers required tuning hyperparameters and testing various preprocessing pipelines. Starting simple enabled stable initial results and iterative enhancements.

Risk: Data Limitation
Mitigation: Leverage synthetic data for initial iterations and incorporate real-world datasets later.
Alignment: The dataset used in early phases included oversampling for minority ransomware classes. This approach allowed initial model training and validation in the absence of diverse real-world datasets. Real encrypted traffic and labelled pcap datasets were introduced in later phases.

Risk: Stakeholder Delays
Mitigation: Set fixed review dates and use collaboration tools to facilitate asynchronous communication.
Alignment: Stakeholder inputs (cybersecurity analysts, AI experts, network engineers) were required to validate modules such as feature engineering and anomaly detection. Slack review schedules ensured feedback could be received even when experts were not available synchronously.

Risk Summary Table

Table 02: Project Risks and Mitigations

Risk	Likelihood	Impact	Severity	Mitigation Plan
Iteration result delays	Medium	High	Medium	Incorporate buffer time into each iteration; revisit model evaluation metrics where needed.
AI implementation challenges	High	Medium	High	Simplify initial design; add complexity iteratively based on evaluation metrics (accuracy, F1-score).
Limited dataset availability	Medium	High	Medium	Use synthetic data initially; transition to real encrypted network traffic for more accurate training.
Stakeholder feedback delays	Low	Medium	Medium	Schedule fixed review periods and enable asynchronous collaboration via shared tools and platforms.
False positives in detection	Medium	High	High	Regularly retrain the model using mislabelled outputs; refine anomaly thresholds and test with mixed datasets.
Model overfitting	Medium	High	High	Use cross-validation, regularization techniques, and careful feature selection (as applied with SelectKBest).
Deployment environment mismatch	Medium	Medium	Medium	Test system compatibility across varied platforms; evaluate on both CLI and simulated hospital network setups.

4. Implementation

4.1 Requirement Analysis

The development of the system requirements was guided by the identified challenges in detecting ransomware threats within its C2 traffic and the necessity of a solution that prioritizes both security and privacy. Discussions with Subject Matter Experts (SMEs), cybersecurity analysts, AI specialists, and network security professionals helped translate these challenges into actionable engineering requirements.

Stakeholder Discussions

- **Cybersecurity Analysts:** Highlighted the limitations of existing tools in identifying ransomware in encrypted traffic, emphasizing the need for robust preprocessing and feature extraction mechanisms to improve detection accuracy.
- **AI Specialists:** Recommended advanced machine learning and deep learning models to improve malware detection capabilities. They stressed the importance of a dataset enriched with real-world ransomware patterns and adaptive learning capabilities to handle emerging threats.
- **Network Security Professionals:** Shared insights on the importance of minimal latency for real-time applications and the challenges of balancing data integrity, confidentiality, and system performance in encrypted traffic environments.
- **Compliance and Privacy Experts:** Advocated for methods that ensure the system upholds user privacy while analysing encrypted traffic, aligning with regulatory requirements such as General Data Protection Regulation (GDPR) and HIPAA.

Final Deliverables

1. **Traffic Preprocessing Module:** Tools for encrypted traffic analysis and feature extraction.
2. **AI Detection Engine:** Advanced machine learning pipelines for identifying ransomware in encrypted flows.
3. **Real-time Monitoring Interface:** Alerts in case of a ransomware infection on a command line interface.
4. **System Documentation:** Technical documentation for deployment, model training, and system maintenance.

4.2 Design

The pipeline is designed to automate network anomaly detection through a series of modular stages. It begins with live packet capture using a sniffer to generate .pcap files, which are then converted into CSV format with detailed flow-level features using Cicflowmeter. The data undergoes preprocessing to standardize column names, handle missing values, and select the top 10 features identified as significant for the pre-trained Random Forest model. Predictions classify traffic as benign or anomalous, and saved.

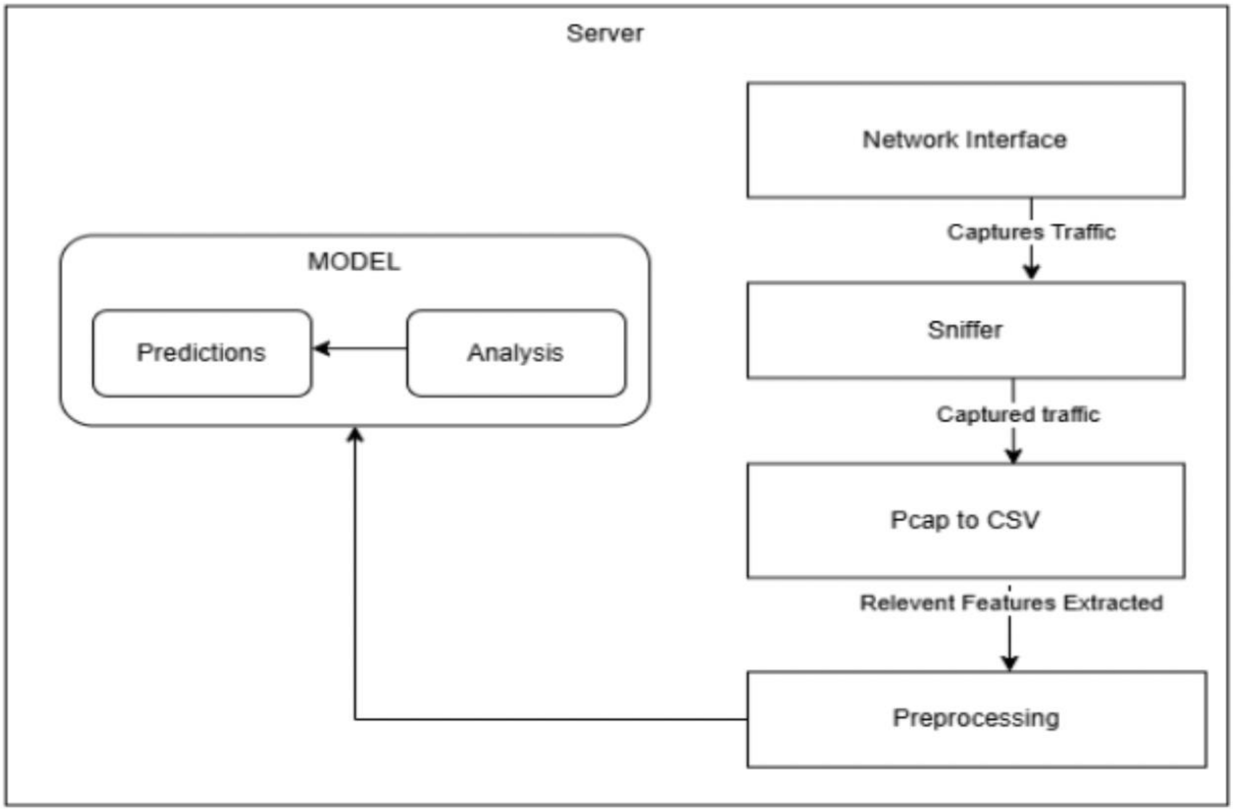


Figure 02: Context Diagram

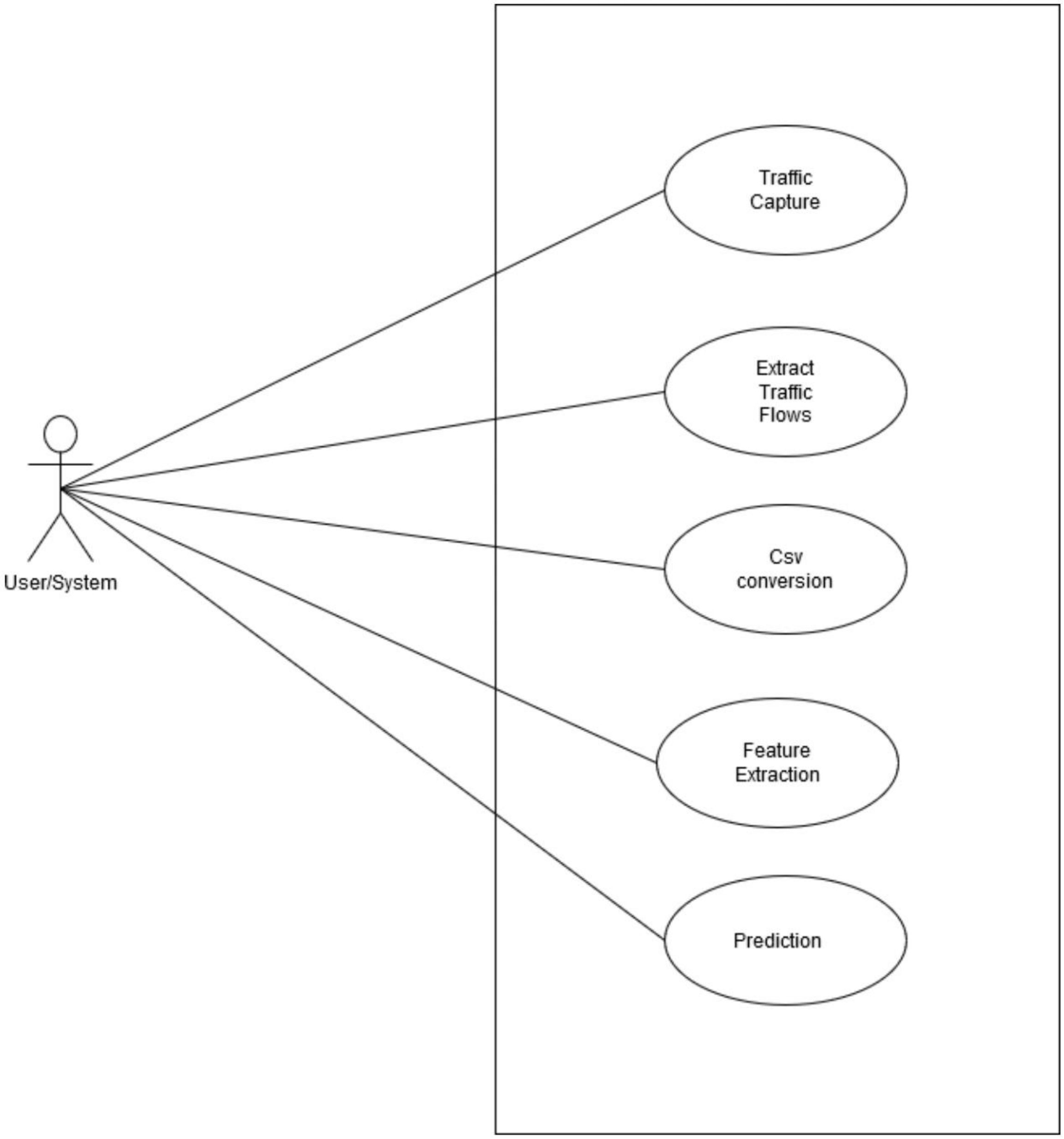


Figure 03: Use Case Diagram

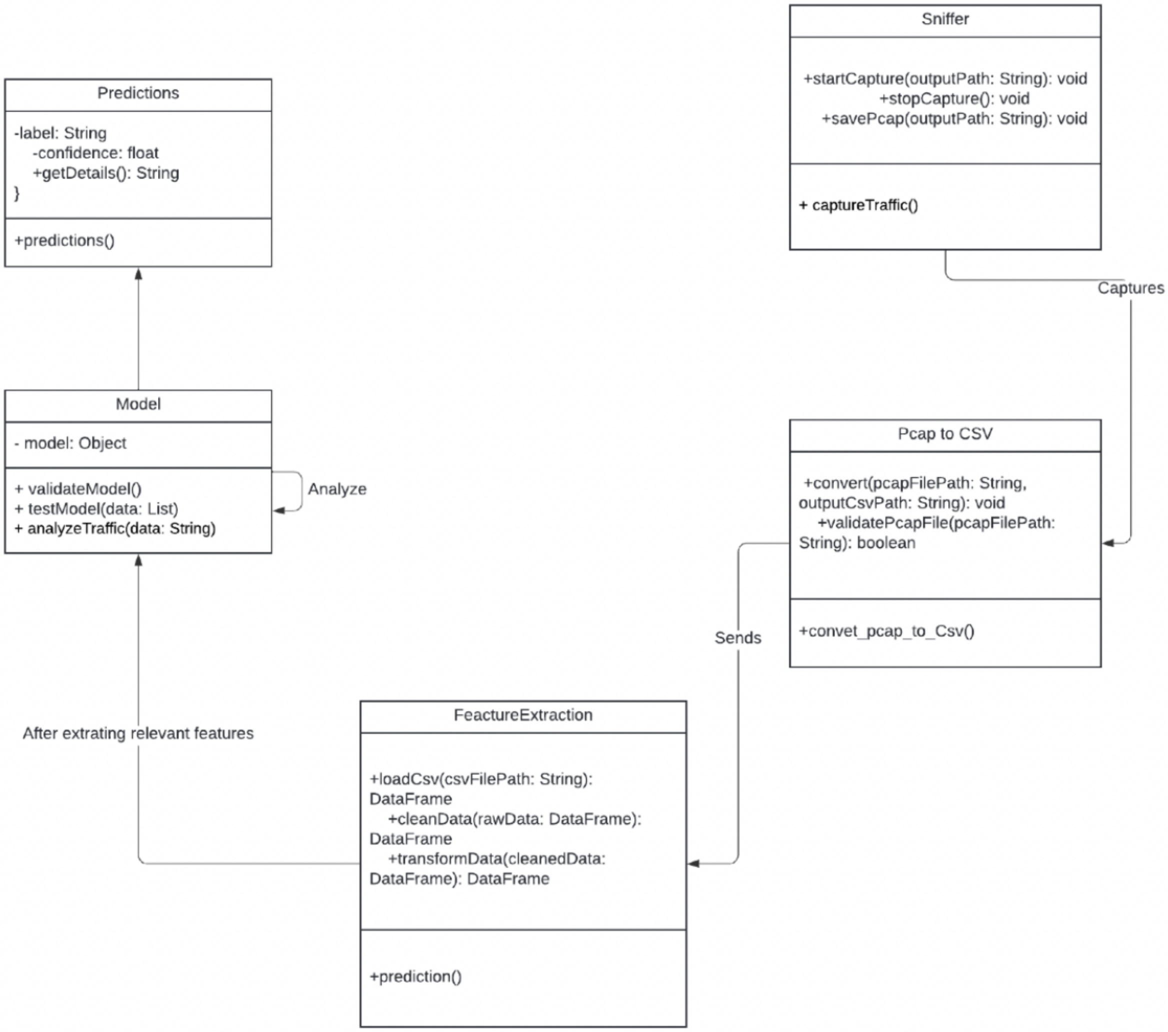

Sniffer
+startCapture(outputPath: String): void
+stopCapture(): void
+savePcap(outputPath: String): void
+ captureTraffic()
Captures
Predictions
-label: String
-confidence: float
+getDetails(): String
}
+predictions()
Model
- model: Object
+ validateModel()
+ testModel(data: List)
+ analyzeTraffic(data: String)
Analyze
Pcap to CSV
+convert(pcapFilePath: String, outputCsvPath: String): void
+validatePcapFile(pcapFilePath: String): boolean
+convet_pcap_to_Csv()
Sends
After extrating relevant features
FeactureExtraction
+loadCsv(csvFilePath: String): DataFrame
+cleanData(rawData: DataFrame): DataFrame
+transformData(cleanedData: DataFrame): DataFrame
+prediction()

Figure 04: UML Diagram

4.3 System Modules

Below are the modules for Ransomware Detection System:

- **Module 1:** D**ataset Feature Engineering**

  Extract network traffic features such as flow duration, packet length, and protocol usage that help in identifying ransomware-related traffic patterns.

- **Module 2: Data Capturing**

  Capture real-time network traffic using packet sniffing tools and preprocess the captured data into structured CSV format for further analysis.

- **Module 3: AI-based Model**

  Train a Random Forest classifier on labelled datasets, focusing on features that differentiate ransomware traffic from benign traffic.

- **Module 4: Monitoring for Ransomware Communications**

  Implement anomaly detection techniques to identify suspicious command-and-control (C2) communications commonly used by ransomware.

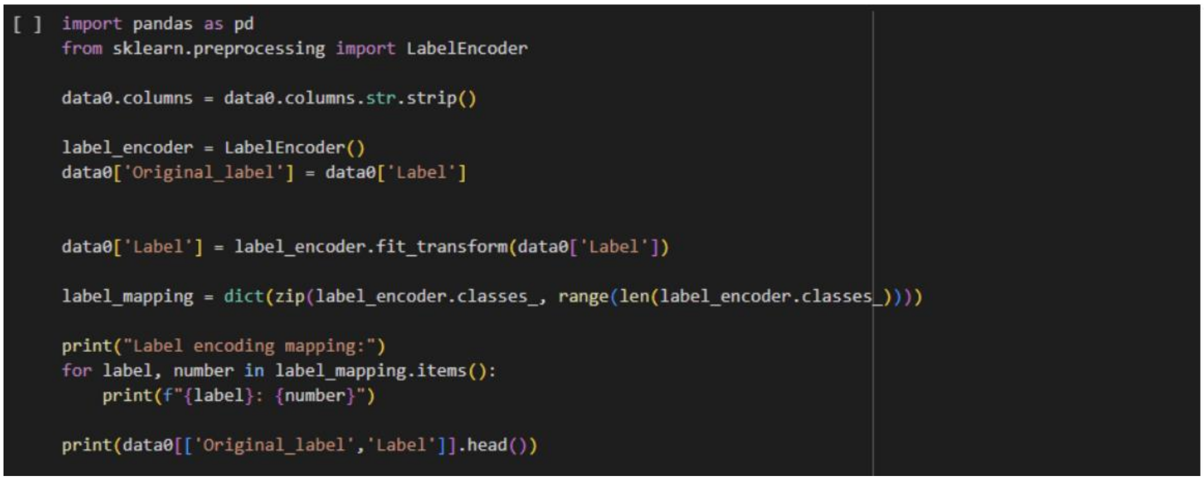
- **Module 5: Detection Algorithms**

  Develop detection algorithms to flag irregular traffic patterns indicative of ransomware activities, such as encrypted traffic or abnormal packet flows.

4.4 Model Implementation

Preprocessing

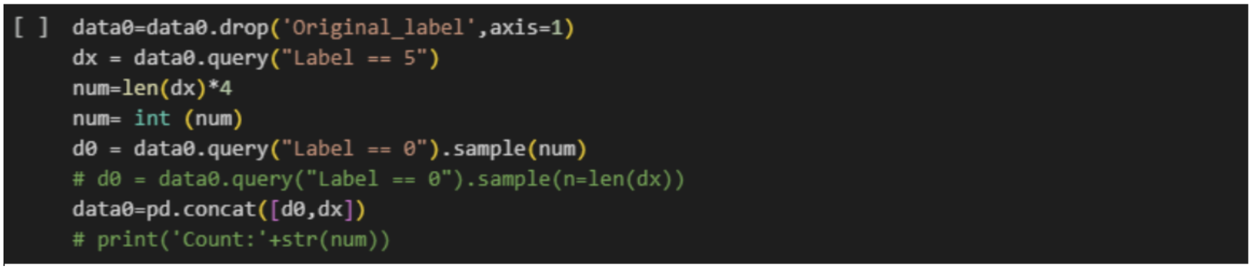
- **Label Encoding:** The Label Column is categorical and includes all types of attacks and hence was encoded into numerical values using LabelEncoder for use by machine learning algorithms. An original copy of the labels was retained for interpretability during analysis.



```
import pandas as pd
from sklearn.preprocessing import LabelEncoder

data0.columns = data0.columns.str.strip()

label_encoder = LabelEncoder()
data0['Original_label'] = data0['Label']


data0['Label'] = label_encoder.fit_transform(data0['Label'])

label_mapping = dict(zip(label_encoder.classes_, range(len(label_encoder.classes_))))

print("Label encoding mapping:")
for label, number in label_mapping.items():
    print(f"{label}: {number}")

print(data0[['Original_label','Label']].head())
```

Figure 05: Code Snippet for Label Encoding

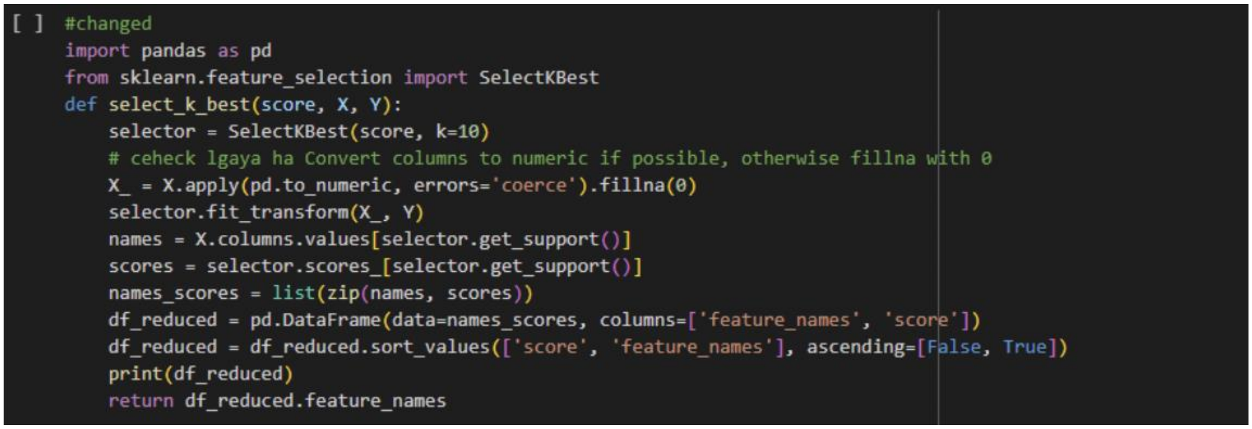
- **Balancing the Dataset:** The oversampling was done on the minority class (Label == 5), while the same number of samples as the minority class was chosen for the majority class (Label == 0). This would not bias the model toward the majority class.



```
data0=data0.drop('Original_label',axis=1)
dx = data0.query("Label == 5")
num=len(dx)*4
num= int (num)
d0 = data0.query("Label == 0").sample(num)
# d0 = data0.query("Label == 0").sample(n=len(dx))
data0=pd.concat([d0,dx])
# print('Count:'+str(num))
```

Figure 06: Code Snippet for Dataset Balancing

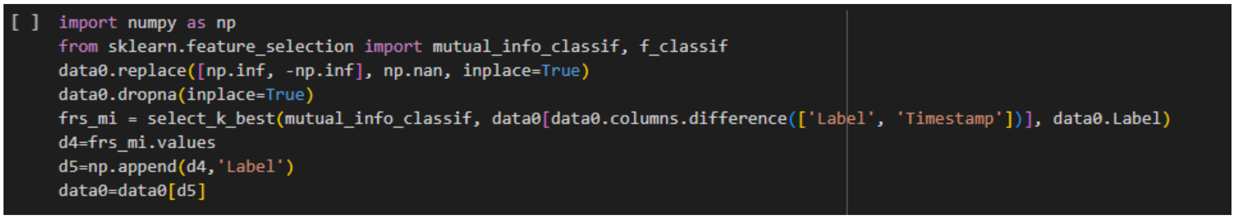
- **Feature Selection:** Using the SelectKBest method with mutual_info_classif as the scoring, relevant features were selected, reducing the dimensionality of the dataset. This improved model performance and reduced computation time. The top ten features, by their scores, were selected for the final dataset.



```
[ ]  #changed
     import pandas as pd
     from sklearn.feature_selection import SelectKBest
     def select_k_best(score, X, Y):
         selector = SelectKBest(score, k=10)
         # ceheck lgaya ha Convert columns to numeric if possible, otherwise fillna with 0
         X_ = X.apply(pd.to_numeric, errors='coerce').fillna(0)
         selector.fit_transform(X_, Y)
         names = X.columns.values[selector.get_support()]
         scores = selector.scores_[selector.get_support()]
         names_scores = list(zip(names, scores))
         df_reduced = pd.DataFrame(data=names_scores, columns=['feature_names', 'score'])
         df_reduced = df_reduced.sort_values(['score', 'feature_names'], ascending=[False, True])
         print(df_reduced)
         return df_reduced.feature_names
```

Figure 07: Code Snippet for Feature Selection

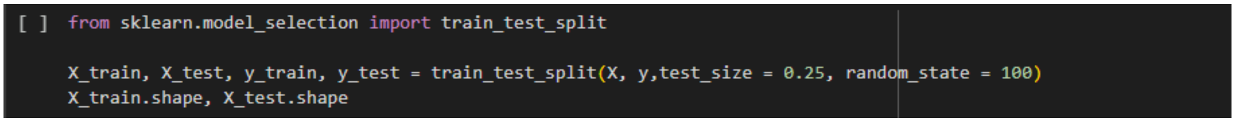
- **Data Cleaning and Transformation:** Missing, infinite, and NaN values were replaced or dropped to ensure integrity in the data. Data was shuffled using sample(frac=1) to avoid any order bias during training.



```
[ ]  import numpy as np
     from sklearn.feature_selection import mutual_info_classif, f_classif
     data0.replace([np.inf, -np.inf], np.nan, inplace=True)
     data0.dropna(inplace=True)
     frs_mi = select_k_best(mutual_info_classif, data0[data0.columns.difference(['Label', 'Timestamp'])], data0.Label)
     d4=frs_mi.values
     d5=np.append(d4,'Label')
     data0=data0[d5]
```

Figure 08: Code Snippet for Data Cleaning & Transformation

- **Dataset Splitting:** The dataset was then divided into training and testing sets in the ratio 75:25 to assess model performance on unseen data.

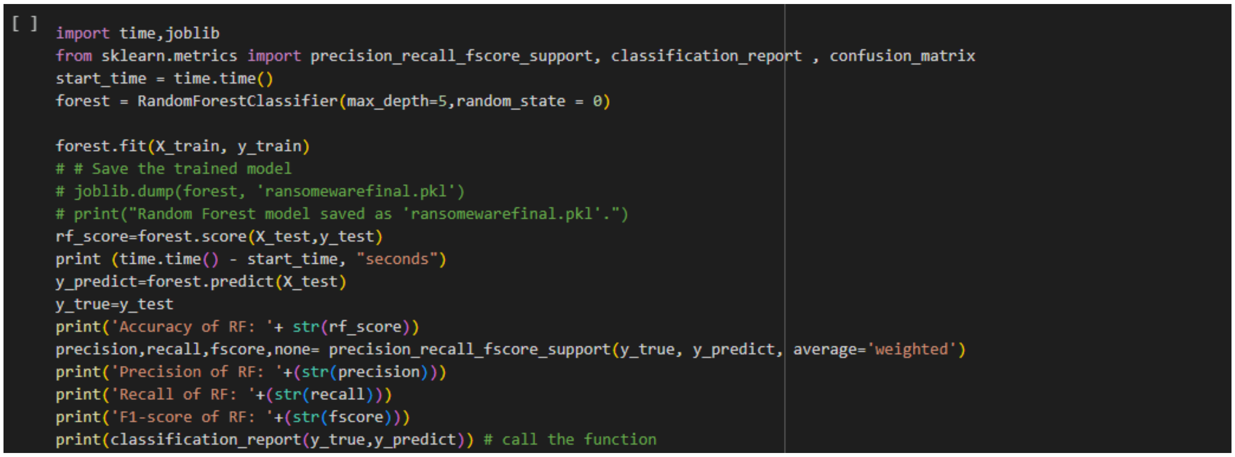


```
[ ]  from sklearn.model_selection import train_test_split

     X_train, X_test, y_train, y_test = train_test_split(X, y,test_size = 0.25, random_state = 100)
     X_train.shape, X_test.shape
```

Figure 09: Code Snippet for Dataset Split

Algorithms

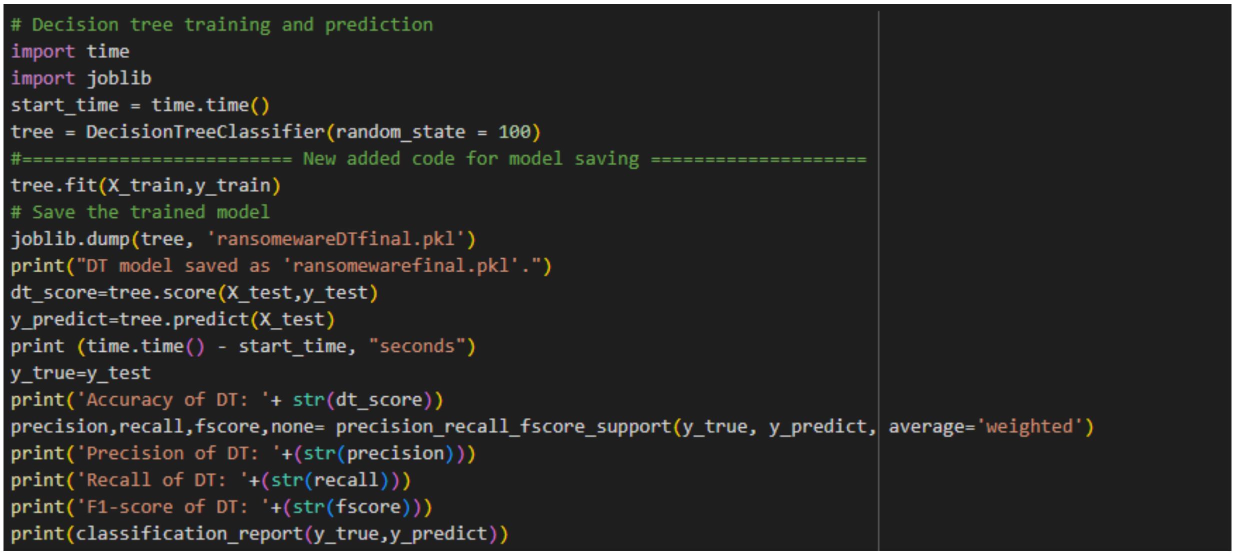
- **Random Forest Classifier:** A random forest classifier was trained with max_depth = 5 to classify the ransomware attack patterns. Random forests were used here because of their robustness in handling both numerical and categorical data and resisting overfitting.



```python
import time,joblib
from sklearn.metrics import precision_recall_fscore_support, classification_report , confusion_matrix
start_time = time.time()
forest = RandomForestClassifier(max_depth=5,random_state = 0)

forest.fit(X_train, y_train)
# # Save the trained model
# joblib.dump(forest, 'ransomewarefinal.pkl')
# print("Random Forest model saved as 'ransomewarefinal.pkl'.")
rf_score=forest.score(X_test,y_test)
print (time.time() - start_time, "seconds")
y_predict=forest.predict(X_test)
y_true=y_test
print('Accuracy of RF: '+ str(rf_score))
precision,recall,fscore,none= precision_recall_fscore_support(y_true, y_predict, average='weighted')
print('Precision of RF: '+(str(precision)))
print('Recall of RF: '+(str(recall)))
print('F1-score of RF: '+(str(fscore)))
print(classification_report(y_true,y_predict)) # call the function
```

Figure 10: Code Snippet for Random Forest Classifier

- **Decision Tree Classifier:** A Decision Tree Classifier was also implemented for comparative analysis. Decision trees provide interpretability by showing how decisions are made at each node.



```python
# Decision tree training and prediction
import time
import joblib
start_time = time.time()
tree = DecisionTreeClassifier(random_state = 100)
#========================= New added code for model saving ===================
tree.fit(X_train,y_train)
# Save the trained model
joblib.dump(tree, 'ransomewareDTfinal.pkl')
print("DT model saved as 'ransomewarefinal.pkl'.")
dt_score=tree.score(X_test,y_test)
y_predict=tree.predict(X_test)
print (time.time() - start_time, "seconds")
y_true=y_test
print('Accuracy of DT: '+ str(dt_score))
precision,recall,fscore,none= precision_recall_fscore_support(y_true, y_predict, average='weighted')
print('Precision of DT: '+(str(precision)))
print('Recall of DT: '+(str(recall)))
print('F1-score of DT: '+(str(fscore)))
print(classification_report(y_true,y_predict))
```

Figure 11: Code Snippet for Decision Tree Classifier

Evaluation Metrics Used

Metrics such as recall, accuracy, precision, and F1-score were used to assess each trained algorithm's performance. These measurements shed light on how well the algorithms identified anomalies:

- **Accuracy:** The proportion of correctly identified cases relative to all instances.
- **Precision:** The percentage of all positive predictions that are positive.
- **Recall:** The percentage of true positive occurrences among all actual positive predictions.
- **F1-score:** A balanced assessment of model performance provided by the harmonic mean of precision and recall.

5.Testing

Testing forms one of the most important processes for the ransomware detection system to ensure that it meets all requirements and works as intended. Since the Iterative Design Methodology was adopted, testing had to be performed in an Agile-friendly way: incremental improvements and bug fixes through frequent evaluations in forms of individual unit test cases.

5.1 Test Cases

Table 03: Test Cases

No	Test Case/Test Script	Attribute and Value	Expected Result	Result
1	Packet Capture	Incoming Raw traffic	.pcap file output	Pass
2	Extraction of Traffic Flows	.pcap to csv conversion with 84+ flows	Csv with 84+ flows	Pass
3	Feature Extraction	Extract relevant features from CSV for prediction	Pre-processed data	Pass
4	Ransomware Detection - Test 1	Simulated Ransomware Traffic Data	Ransomware detected and flagged	Pass
5	Ransomware Detection - Test 2	Normal Traffic Data	No ransomware detected	Pass
6	Ransomware Detection - Test 3	Mixed Traffic Data (normal + ransomware)	Ransomware detected in the mixed traffic	Pass

5.2 Testing Approach

An iterative testing procedure was executed following each development cycle to confirm both the system's stability and functionality of ransomware detection before advancing to subsequent development stages. The testing followed an iterative method which enabled ongoing assessment and improvement across different development cycles to accumulate findings from every iteration. New requirements and development process changes triggered dynamic test case updates which maintained testing alignment to project objectives. The system development requirements underwent step-by-step mapping to test cases that built an extensive testing framework. A thorough mapping process protected the project from missing essential system components while ensuring a rigorous examination of complete system functionality. The test cases followed a meticulous design process in which the examined system functions independently as well as the collective performance of all components. Testing components separately would often fail to reveal critical problems that became visible only through this dual approach. The main purpose of testing examined how well the AI model recognized ransomware signatures throughout the dataset. The model underwent detection capability testing through the submission of sample input consisting of ransomware attack simulators. We measured the system's accuracy and efficiency by comparing the model-generated outputs to their expected results. The testing method helped the team both discover successful model outcomes and find situations demanding extra refinement.

The testing protocol included assessments of standard ransomware signatures while also testing modified attack strategies. The assessment of these cases held particular value because ransomware threats continuously evolve using new detection-evasion methods. The team evaluated emerging threat resilience and adaptability of the model through exposing it to varied examples. The model's capacity to identify modifications in its inputs with precision demonstrated its readiness for operational deployment while proving its robust design. Testing performed in an iterative manner turned out to be the process's most powerful advantage. Relentless testing iterations enabled researchers to identify system weaknesses which prompted developers to enhance the detection solution. Through ongoing feedback system, the development team made successive system enhancements which produced an improved robust ransomware detection system. Technical testing of the system was complemented by evaluations that assessed usability aspects and integration abilities. The team acknowledged the need for simple deployment and operation in diverse environments during their model technical performance testing phase. The testing process examined how the system interacts with current cybersecurity technology while checking its OS operability and its performance with different data quantity requirements.

The system achieved its functional and performance targets through a detailed testing method that was repeated multiple times. The team successfully delivered an effective ransomware detection system by validating AI model capabilities and testing adaptability across evolving threats and validating reliability across different operating conditions. Through this extensive testing regimen, the system received quality improvements while generating useful insights that will benefit future development and optimization work.

5.3 Validation Metric Results

The performance of the different machine learning algorithms on anomaly detection was determined through different measurements such as accuracy, precision, recall, and F1-score. The results of testing the models on the dataset are the following:

Random Forest: Below is the performance metrics results for Random Forest Classifier:

Accuracy	0.916
Precision	0.915
Recall	0.916
F1-Score	0.911

The Random Forest model has shown incredibly superior performance in detecting ransomware attacks, with an accuracy of 91.65%, meaning the majority of the predictions were correct. With the precision of 91.52%, the model has shown incredibly superior performance in minimizing false positives, while the recall of 91.65% highlights its effectiveness in correctly identifying actual ransomware instances. Furthermore, the F1-score of 91.10% indicates a particularly good balance between precision and recall, thus confirming that this model is reliable in those scenarios where both false positives and false negatives need to be minimized. All these metrics together support the robustness of the Random Forest model in identifying ransomware patterns with good accuracy.

Decision Tree: Below is the performance metrics results for Decision Tree Classifier:

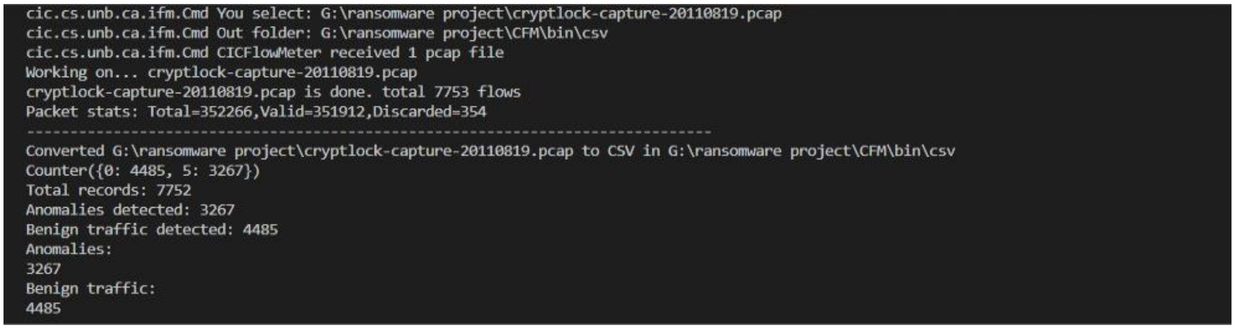
Accuracy	0.93
Precision	0.93
Recall	0.93
F1-Score	0.93

The Decision Tree model had excellent results for performance measures, correctly classifying most of the instances with an accuracy of 93.88%. Strong reliability in the minimization of false positives can also be observed with its high precision of 93.87%. With a recall value of 93.88%, the model identifies most actual ransomware instances well. It also had an excellent F1-score at 93.87%, balancing the two previous performance metrics very well.

However, in real-time testing, the Random Forest model gave more accurate predictions, even though it has slightly lower accuracy metrics. That means, although Decision Tree had a superior performance in static evaluation, Random Forest was more robust and reliable in handling real-world scenarios, which makes it more suitable for real-time ransomware detection.

6.Results

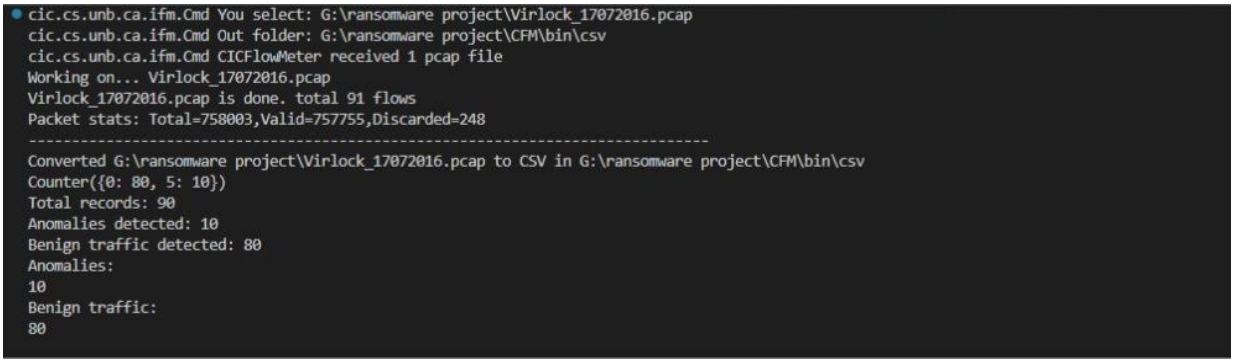
To verify the Real time system, several ransomware data sets of network traffic pcaps were used, based on which was tested and validated. For testing, the system was fed with multiple pcap files, and Cicflowmeter was used to convert the obtained data into CSV format for the final analysis. The data testing is in such a way that it validates real-time streams' performance on a system to confirm the capability duty of the Random Forest model in various scenarios. Linking of the pipeline, feature extraction, and the prediction is in such a way that all the pieces will work in harmony to make the overall flow of the system efficient and accurate. Below is a sample of real time testing of a cryptlock ransomware pcap file from which the model predicted anomalies correctly.



```
cic.cs.unb.ca.ifm.Cmd You select: G:\ransomware project\cryptlock-capture-20110819.pcap
cic.cs.unb.ca.ifm.Cmd Out folder: G:\ransomware project\CFM\bin\csv
cic.cs.unb.ca.ifm.Cmd CICFlowMeter received 1 pcap file
Working on... cryptlock-capture-20110819.pcap
cryptlock-capture-20110819.pcap is done. total 7753 flows
Packet stats: Total=352266,Valid=351912,Discarded=354
--------------------------------------------------------------------------------
Converted G:\ransomware project\cryptlock-capture-20110819.pcap to CSV in G:\ransomware project\CFM\bin\csv
Counter({0: 4485, 5: 3267})
Total records: 7752
Anomalies detected: 3267
Benign traffic detected: 4485
Anomalies:
3267
Benign traffic:
4485
```

Figure 05: Real-time test-1

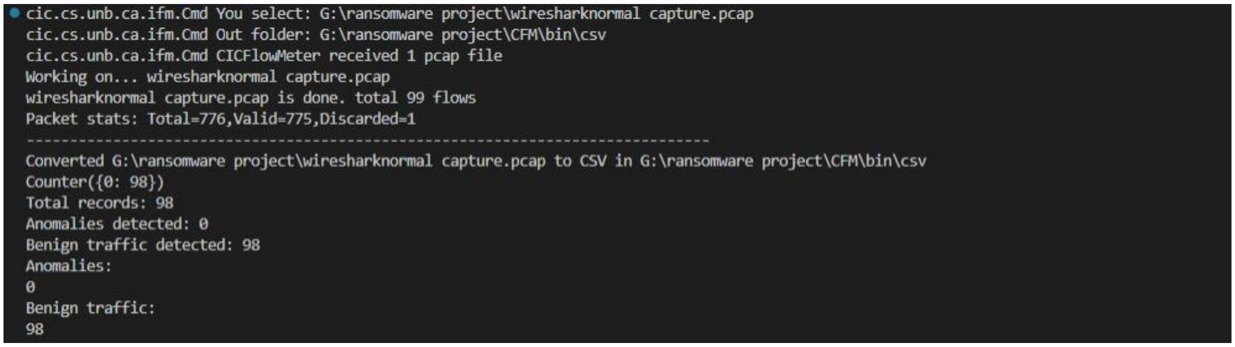
Here is another ransomware pcap file tested and the model predicted anomalies correctly.



```
cic.cs.unb.ca.ifm.Cmd You select: G:\ransomware project\Virlock_17072016.pcap
cic.cs.unb.ca.ifm.Cmd Out folder: G:\ransomware project\CFM\bin\csv
cic.cs.unb.ca.ifm.Cmd CICFlowMeter received 1 pcap file
Working on... Virlock_17072016.pcap
Virlock_17072016.pcap is done. total 91 flows
Packet stats: Total=758003,Valid=757755,Discarded=248
--------------------------------------------------------------------------------
Converted G:\ransomware project\Virlock_17072016.pcap to CSV in G:\ransomware project\CFM\bin\csv
Counter({0: 80, 5: 10})
Total records: 90
Anomalies detected: 10
Benign traffic detected: 80
Anomalies:
10
Benign traffic:
80
```

Figure 06: Real-time test-2

Now testing on normal traffic and model again predicted correctly with no anomalies detected.



```
cic.cs.unb.ca.ifm.Cmd You select: G:\ransomware project\wiresharknormal capture.pcap
cic.cs.unb.ca.ifm.Cmd Out folder: G:\ransomware project\CFM\bin\csv
cic.cs.unb.ca.ifm.Cmd CICFlowMeter received 1 pcap file
Working on... wiresharknormal capture.pcap
wiresharknormal capture.pcap is done. total 99 flows
Packet stats: Total=776,Valid=775,Discarded=1
--------------------------------------------------------------------------------
Converted G:\ransomware project\wiresharknormal capture.pcap to CSV in G:\ransomware project\CFM\bin\csv
Counter({0: 98})
Total records: 98
Anomalies detected: 0
Benign traffic detected: 98
Anomalies:
0
Benign traffic:
98
```

Figure 07: Real-time test-3

7.Technical Discussion

The implementation of the CLI-based ransomware detection system turned out to be a very enriching experience and has allowed us to understand the strengths and weaknesses of the solution I developed. A useful experience in understanding how artificial intelligence (AI) models can be plugged into real world use cases, and where improvements could still be made. Not only does it help me understand the process of AI based security systems, but it also showed me what are the practical challenges and what areas need to be optimized in future. The following section examines critical insights from implementation alongside a rigorous examination of the technological strengths and weaknesses.

7.1 Learnings from the Implementation

The first step in the journey to implement the ransomware detection system was on a solid theoretical basis, but it became evident soon that building a robust and efficient detection system is a complex problem. The first key takeaway from the process was that efficient model training is critical to building a reliable detection system. I had to carefully curate the training data to prevent it from mixing benign and malicious ransomware behaviour. For training the AI model, a high-quality dataset including a vast variety of ransomware signatures and attack patterns was essential. It was the iterative approach to training, updating a model, testing it, and refining, that helped bring the model to its ultimate effectiveness.

Additionally, this development highlighted the need for comprehensive testing and optimization. The first model did well on a set of test cases, but it was obvious that it needed some fine tuning to handle edge cases and different attack methods. During the implementation, weaknesses were identified early on and corrected in subsequent versions of the implementation using the iterative methodology followed. Continuously optimizing and adjusting the model resulted in the model’s ability to detect sophisticated ransomware variants, which otherwise eluded detection.

The realization of another significant importance was real time data handling. As ransomware attacks are rapidly evolving, it was clear that I would need to be able to handle streaming data to detect them. At first, the system was not designed to handle massive amounts of data in a real time manner, resulting in lengthy delays in detection and analysis. I gradually improved the system, so it took as little time as possible to process it, greatly increasing the speed and responsiveness of the detection system overall. Parallel processing techniques and algorithm optimization were used to reduce bottlenecks.

The best part of the implementation was to see the system behave in the real world. The AI model was extremely promising, detecting a wide variety of known ransomware samples, and having a nice false positive rate. However, the practical challenges indicated that real world environments are less tractable than expected and additional refinement is necessary to guarantee that the system can operate in a variety of environments, including networked and cloud environments.

7.2 Critical Review of Technical Merits

The design behind the AI model at the heart of the ransomware detection system was grounded on sound design, technically. The system was designed to focus on identifying ransomware signatures and attack behaviours and was able to detect known threats with impressive accuracy. The model was able to pick up on patterns unique to ransomware activity — such as encryption of files, changes in file extensions and attempts to access lots of sensitive data. Although this approach worked, it was limited by the limitations of the dataset and the dynamic nature of ransomware attacks, which evolve as they go.

For implementation, it was deliberately chosen to use a Command Line Interface (CLI), which has merits and drawbacks in its use. The CLI based implementation was also lightweight on the positive side, so it was easy to deploy and integrate into existing environments. It could easily be operated in a wide range of systems, especially in systems with limited resources, and it could be inserted into automation scripts for continuous monitoring. The system proved to be suitable for integration into existing IT environments and into enterprise level systems where low resource overhead is an extremely crucial factor. Although, the decision to implement the system without a dedicated graphical user interface (GUI) had its limitations. The usability of the system was limited due to the lack of an intuitive user interface, particularly for non-technical users or users without a background in cybersecurity. The CLI was great for experienced users to interact with the system but was unable to visualize historical data or generate detailed reports on detected threats. As a result, the system could not be used for more advanced tasks, such as detailed past detection analysis, production of graphical reports, or long-term trends in ransomware attacks. These are the capabilities any security system that seeks to provide full protection and insight of possible vulnerabilities within an organization requires.

The detection system's fundamental design remained secure even with its implementation difficulties while demonstrating effective detection capabilities toward multiple ransomware strains. The CLI interface showed initial success but holds promise for upcoming enhancements to its functionality. A refined user interface together with advanced learning capabilities for the AI model would substantially enhance both performance and usability in operational environments. The system's ransomware combat effectiveness could be boosted through advanced features that integrate automated incident response capabilities along with cloud detection functions and tool collaboration features.

Overall, the CLI-based ransomware detection system documented the potential of AI applications in cybersecurity yet exposed ongoing difficulties in applying AI models to dynamic threats across the cyber landscape. This implementation alongside its assessment of strengths and weaknesses generates essential knowledge that enables improvements to ransomware detection capabilities and their availability to broader user groups.

8. Project Review

The development of the Ransomware Detection System was an enriching learning experience, ranging from technical and methodological aspects of project execution. It underlined the importance of careful planning, iterative design, and continuous testing while revealing opportunities for improvement and the inherent limitations of the project.

8.1 Key Learnings:

- **Effective Planning and Iterative Development:**
  The iterative methodology was followed quite instrumentally in the breakdown of this project into manageable phases, which allowed regular review and flexibility in accommodating refinements, based on testing. Needless to say, there has been a realization that in such projects, particularly related to debugging or optimizing specific components, like an AI detection model or preprocessing routines, modular design is vital.
- **Technical Insights:**
  The project involved a practical approach toward developing an AI-enabled solution for a real-world problem. The design of the detection system showed that developing balanced datasets, establishing an efficient preprocessing pipeline, and having good response time and accuracy performance metrics are key factors in achieving system effectiveness.

- **Time and Resource Management:**
  Project management within the schedule was a key learning. The entire process, right from data collection, model development and testing to optimization, kept pretty much within the scheduled time frame. Some tasks, though, took a bit longer, like trying to improve generalization for a model, meaning that a schedule should be really prepared to bear worse-case scenarios.

8.2 Project Limitations

- **Scope of Implementation:** The project was constrained to a CLI-based solution with no advanced user interfaces. This kept the project focused and within manageable limits, but it also limited the functionality of the system in terms of user-friendliness and scalability.

- **Data Availability:** Real-world ransomware datasets were not freely available in abundance, thus the simulated attack patterns. This might have affected the model's generalization capability on an entirely new ransomware type or zero-day attack.

- **Time Constraints**: Some of the functionalities that were planned could not be integrated, like building in advanced reporting or perhaps even investigating other algorithms, due to time constraints.

9. Conclusion

The project effectively identified the most prevalent ransomware attacks targeting healthcare centres, mainly the NHS, and evaluated various existing defence mechanisms to further enhance their effectiveness. The research systematically addressed the objectives and, as such, provided actionable insight into strengthening the cybersecurity posture of healthcare organizations. It also conducted an in-depth examination of recent ransomware incidents, including the Synnovis attack in June 2024 (Synnovis cyber-attack – statement from the NHS England). Common vulnerabilities identified include unpatched software, lack of network segmentation, and untrained employees. Current defensive mechanisms examined demonstrated strengths but critical weaknesses, such as relying on manual responses and utilizing old infrastructure, which highlights a specific need for automated, proactive security measures. The major contribution of the project was the AI-based ransomware detection system developed. The Random Forest model has shown robust performance in real-time scenarios with an accuracy of 91.65% and thus is a reliable tool for detecting patterns of ransomware. Additional recommendations included upgrading encryption protocols and providing comprehensive cybersecurity training to all employees to help deal with evolving threats. While the system proved effective, limitations such as reliance on simulated datasets and the absence of a user-friendly interface restricted scalability. Future work should include the incorporation of diverse real-world datasets, development of a GUI, and investigation of hybrid detection algorithms that improve accuracy and usability. Collaboration between healthcare organizations and cybersecurity experts will also be needed to share threat intelligence and improve defences. This project demonstrated how much machine learning models could go a long way in enhancing ransomware detection and mitigation. It offers a practical roadmap of how healthcare organizations should strengthen their cybersecurity frameworks and protect critical data from ransomware threats. In conclusion, this project delivers a thorough method for detecting and handling ransomware attacks. This foundation helps in developing improved cybersecurity procedures through attack identification and existing defence evaluation alongside vulnerability analysis. AI-based detection systems represent a major advancement because they prove how machine learning solutions can fight ransomware attacks. Advancing these efforts requires us to actively fix detected weaknesses and establish stronger joint initiatives between stakeholders. This research shows how healthcare organizations must develop advanced protective approaches alongside innovative methods to secure critical data from modern cyber intrusions.

10. References

Ahmad, M. and Ahmad, M., 2019. Enhancing ransomware detection using machine learning. *International Journal of Computer Science and Network Security*, 19(9), pp.43-50.

Ali, S. and Ali, S., 2021. A comprehensive study on cybersecurity risks in healthcare. *Information Systems Security*, 30(1), pp.15-27.

Alraizza, A. and Alraizza, A., 2023. Ransomware detection using machine learning: A survey. *Big Data and Cognitive Computing*, 7(3), p.143.

Bhardwaj, A. and Bhardwaj, A., 2022. AI-driven solutions to mitigate healthcare cyber threats. *Journal of Information Security and Applications*.

Bojanc, R. and Blazic, B.J., 2018. Threat assessment and security analysis in healthcare.

Borisenkov, I., 2020. Ransomware threat modeling in critical sectors.

Darktrace, 2017. *NHS agency successfully fought back WannaCry ransomware with Darktrace*. [online] Available at: https://darktrace.com/news/nhs-agency-successfully-fought-back-wannacry-ransomware-with-darktrace [Accessed 24 Apr. 2025].

Ellison, R. et al., 2021. Leveraging AI in ransomware mitigation for healthcare. *IEEE Security & Privacy*, 19(6), pp.19–27.

Fruhlinger, J., 2019. How cybercriminals exploit healthcare systems. *CSO Online*. Available at: https://www.csoonline.com/article/3338258 [Accessed 24 Apr. 2025].

Ganji, A. and Tamimi, R., 2022. Risk analysis and security measures in healthcare IoT devices. *Journal of Network and Computer Applications*, 199, p.103371.

Haq, R. and Salama, M., 2019. Best practices for ransomware prevention in hospitals. *Security Journal*, 32(4), pp.420–430.

Hayati, A. and Mahdi, K., 2023. Advances in cybersecurity research for healthcare systems. *Cyber Defence Magazine*, 12(3), pp.28–39.

Hassan, M. and Alam, T., 2020. Healthcare data breach trends and prevention techniques. *International Journal of Healthcare Informatics Research*, 10(2), pp.55–68.

Isaac, A. and Clark, R., 2021. Analysing cyber risk factors in ransomware incidents. *Proceedings of the ACM Conference on Computer and Communications Security*, pp.417–428.

Jafri, S. and Ameen, T., 2022. Cybersecurity in critical healthcare systems: A review. *Journal of Medical Systems*, 46(8), p.72.

Kwon, D. and Liu, J., 2021. AI-powered defences against healthcare ransomware. *ACM Transactions on Privacy and Security*, 24(2), pp.15–28.

Lashkari, A.H., 2018. Toward developing a systematic approach to generate benchmark Android malware datasets and classification.

Mahdi, R. and Bakri, A., 2020. Data recovery challenges in healthcare ransomware incidents. *International Journal of Critical Infrastructure Protection*, 31, p.100396.

Malik, S. and Hussain, Z., 2018. Developing adaptive cybersecurity measures for healthcare. *Computers & Security*, 85, pp.215–225.

Milmo, D., 2022. NHS ransomware attack: What happened and how bad is it? *The Guardian*, [online] 11 Aug. Available at: [online] 11 Aug. Available at: https://www.theguardian.com/technology/2022/aug/11/nhs-ransomware-attack-what-happened-and-how-bad-is-it [Accessed 24 Apr. 2025].

Nguyen, T. and Tran, H., 2021. Enhancing healthcare data security using AI. *Journal of Advanced Information Systems*, 29(4), pp.345–360.

PCMag, 2024. *The best ransomware protection for 2025*. [online] Available at: https://www.pcmag.com/picks/the-best-ransomware-protection [Accessed 24 Apr. 2025].

Qureshi, I. and Ahmad, M., 2019. Incident response strategies for healthcare cyber attacks. *Journal of Healthcare Cybersecurity*, 8(2), pp.15–28.

Rubenking, N.J., 2024. *The best ransomware protection for 2025*. [online] PCMag. Available at: https://www.pcmag.com/picks/the-best-ransomware-protection [Accessed 24 Apr. 2025].

Sophos, n.d. *Sophos Intercept X: Next-gen endpoint security*. [online] Available at: https://www.sophos.com/en-us/products/intercept-x [Accessed 24 Apr. 2025].

Synnovis Cyber Attack – Statement from NHS England, 2024. [online] Available at: https://www.england.nhs.uk/2024/06/synnovis-cyber-attack-statement-from-nhs-england/ [Accessed 24 Apr. 2025].

Thamer, N. and Thamer, N., 2021. Cybersecurity preparedness in healthcare: A ransomware perspective. *Healthcare Informatics Research*, pp.22–29.

VMware, n.d. *VMware Carbon Black Cloud Endpoint*. [online] Available at: https://www.vmware.com/products/carbon-black-cloud-endpoint.html [Accessed 24 Apr. 2025].

Zaheer, A.T., 2023. A hybrid model for botnet detection using machine learning. In: *International Conference on Business Analytics for Technology and Security (ICBATS)*. Dubai: IEEE.

Zhao, H. and Feng, D., 2020. Analysis of machine learning models for malware detection. *Journal of Cybersecurity Research*, 19(3), pp.129–140. Available at: https://doi.org/10.1016/j.cybsec.2020.102043.