Resilient Ransomware Detection and Logging Using Machine Learning and Blockchain

Jacob Bott

22019050

BSC IN COMPUTER SCIENCE (CYBER SECURITY)

5/5/2025

SCHOOL OF COMPUTER SCIENCE AND MATHEMATICS

Keele University

Keele

Staffordshire

ST5 5BG

Keele
UNIVERSITY

SUBMISSION DECLARATION FORM

MODULE NUMBER: CSC-30014-2024-SEM1-2-A **MODULE TITLE: Third Year D**

PROJECT TITLE: Resilient Ransomware Detection and Logging Using Machine Learning a

NAME (BLOCK CAPITALS): JACOB BOTT

STUDENT NUMBER: 22019050	**YEAR OF STUDY: 2025**

By submitting this assignment I confirm that:

(a) That the above project is my own account, based upon work actually carried out by me, and tha not resulting from my own experimentation, observation or specimen collecting, including o been clearly indicated.

(b)

- I have read and understood University Regulation D4 'Student Academic Misconduct' https://www.keele.ac.uk/legalgovernancecompliance/governance/actcharterstatutesordina ulationsandpoliciesindex/regulationd4/

- I understand the term ‘plagiarism’ and that this assignment is not plagiarised;
- I have not colluded with anybody in its preparation and production;
- This assignment (or sections of it) has not been submitted for assessment in this or any oth

Please tick here if you have used a proofreader for this assignment □

- I understand that inappropriate use of a proofreader as outlined in the University’s proofre (https://www.keele.ac.uk/media/keeleuniversity/sas/governancedocs/Proofreading%20gui ents%20%20-%20May%202017.pdf and http://www.keele.ac.uk/studentacademicconduct academic misconduct.

(c) That I have read, understood, and abided by the terms of University Regulations below:

Regulation D4: Student Academic Misconduct:

https://www.keele.ac.uk/legalgovernancecompliance/governance/actcharterstatutesordinancesandre ndpoliciesindex/regulationd4/

Student Academic Misconduct Code of Practice:

https://www.keele.ac.uk/media/keeleuniversity/policyzone20/studentandacademicservices/Student Misconduct%20CoP%20-%20June%202024.pdf

DATE: 5/5/2025	**SIGNATURE: jacob bott**

Contents

Abstract

This project proposes a dual-layered defense system combining machine learning (ML) for early ransomware detection with blockchain technology for tamper-proof forensic logging. A Random Forest (supervised) and Isolation Forest (unsupervised) were evaluated using both synthetic data and the CSE-CIC-IDS2018 dataset. The Random Forest model achieved high recall and F1-score, while the Isolation Forest showed strong AUC despite lower recall on real-world data. A blockchain proof-of-concept was deployed using Ethereum smart contracts to immutably log detection events. A simulated BlackByte case study demonstrated that this integrated system could enhance detection, preserve critical logs, and reduce post-attack recovery challenges.

1.0 Introduction

Ransomware is a ty malware that encrypts a victim's files and demands payment (often in cryptocurrency) for restoration. This form of malicious attack is one of the most dangerous and financially damaging threats facing modern organisations. According to IBM's 2024 cost of data breach report, the average cost of a ransomware attack in 2024 exceeds $2.7 million when considering ransom payments, downtime and recovery (IBM, 2024). Traditional forms of prevention like antivirus software and firewalls are becoming increasingly ineffective at ransomware prevention despite significant investment. A study by cybersecurity vendor PurpleSec in 2024 revealed that 75% of companies hit by a ransomware attack had up-to-date endpoint protection solutions in place (purplesec, 2024). This helps to highlight the ability that advanced ransomware has to evade traditional signature-based detection; through techniques such as polymorphism, stealth operations and zero-day exploits. The amount of these attacks continues to grow, a Cyberprint report shows a significant increase in attacks, with 5,414 published incidents in 2024, this is an 11% rise when compared to previous years (Cyberprint, 2024). As the frequency of ransomware attacks continues to rise, so does the evolution of more advanced ransomware strains, making it important that defenders continue to adopt and implement more intelligent and resilient detection and prevention approaches.

Current ransomware detection tools although continuing to improve, mostly rely on known malware signatures or behavioural heuristics, these methods are not foolproof and can be bypassed by novel or stealthy attack techniques. Many modern attackers will modify their code/behaviour to avoid detection, helping to circumvent signature-based detection systems (Sotero, n.d.). Once ransomware infects a system, it can quickly encrypt files, shut down security services, and delete event logs to hide its presence, complicating the work of investigators and security tools attempting to trace a malware’s origin and behaviour (VMRay, 2024).

Current tools tend to only focus on either detection or incident response, this creates a fundamental gap in cybersecurity strategy: with many solutions either detecting ransomware too late or failing to protect critical forensic evidence once an attack is underway. This leads to attackers not only encrypting a victim's files’ but also altering/deleting system logs, affecting the forensic process. This leaves organisations, vulnerable not only during the attack but also in the aftermath.

This project aims to address this gap in cyber security tools by proposing a dual-layered defence framework that combines machine learning and blockchain technology. Machine learning helps to enable early-stage ransomware detection by analysing behavioural anomalies, like spikes in CPU usage, unusual file changes or any irregular network activity that shows any indication of encryption or lateral movement. This approach offers the ability to adapt to unknown attack variants, giving a more robust and proactive method of identifying attacks in real-time, unlike the signature-based approach.

Blockchain technology complements the proposed detection layer, acting as a tamper-resistant logging system and evidence-preservation mechanism. When any suspicious activity is detected, relevant metadata and system events can be recorded immutably on a blockchain ledger. This ensures critical forensic data remains available for post-incident analysis, response, or even legal proceedings by preventing any attackers from covering their tracks by erasing logs. Combining early detection using machine learning and ensuring evidence integrity by using the blockchain, this Project aims to provide a new method of ensuring technical and operational resilience to ransomware.

My suggested machine-learning model will be tested on both simulated data and real-world data. Ransomware data will be simulated using elevated CPU usage, high file modification level and abnormal network traffic. The provided synthetic environment allows for controlled experimentation where the characteristics of benign and malicious activity are clearly defined, making it ideal for initial model training and baseline comparisons.

To show how my system would perform in realistic conditions, this project utilises the CSE-CIC-IDS2018 dataset, widely considered a benchmark in intrusion detection, containing examples of ransomware, botnets, infiltration, and other advanced persistent threats. This project will be using the “infiltration” subset which includes ransomware behaviour based on the CryptoLocker family. This real-world dataset helps to provide a challenging test environment for my machine-learning model, due to its scale, variability and noise, helping to reflect the actual complexity of real enterprise networks/organisations.

A blockchain-based proof of concept (PoC) will also be included to address the issues of forensic tampering. Once any ransomware activity is detected, the PoC system will immutably log key data like timestamps, detection details, or system responses onto a blockchain ledger. This ensures that logs cannot deleted or modified and that a secure, tamper-resistant copy of the logs will remain after an attack for analysis. This PoC will be implemented using smart contracts in solidity, to show how decentralised technologies can reinforce cyber defence strategies.

2.0 Machine Learning and Blockchain for Ransomware Defence: A Literature Review

2.1 Evolution and growing complexity of ransomware

Ransomware continues to rapidly evolve, from simplistic beginnings to a highly sophisticated and disruptive cyber threat. Early ransomware in the late 1980s and 1990s was primitive, however, modern strains have continued to develop, employing strong cryptography, advanced propagation methods, and even multi-phase extortion strategies. This "complete historical journey of ransomware" spans from "simple rudimentary" attacks to today's well-organised operations with "advanced encryption methods and modern distribution techniques" (Nagar, 2024). Key milestones in this evolution include:

- Ransomware-as-a-Service (RaaS): The rise of RaaS platforms has lowered the barrier of entry for cybercriminals. By the late 2010s, even unskilled actors had the ability to launch attacks using turnkey ransomware kits(Nagar, 2024). This commercialization greatly expanded the volume and diversity of attacks, transforming ransomware into "a very profitable business for cybercriminals" with an impact worldwide (Nagar, 2024).
- Targeted Attacks and Double Extortion: Overtime ransomware tactics shifted from indiscriminate "spray-and-pray" infections to targeted intrusions against lucrative organisations. Since 2019, groups such as Maze have pioneered double extortion – "as well as encrypting data, maze exfiltrated sensitive information and demanded a ransomware to not release information", thereby increasing the pressure on the victims. (Nagar, 2024). This tactic spread rapidly across ransomware families, with attackers also conducting careful

reconnaissance and timing to hit critical infrastructure and enterprises. "gone were the days" of purely opportunistic attacks; recent cases (e.g. the 2021 Colonial Pipeline incident) show ransomware "has the capacity to halt essential services" (Nagar, 2024), even threatening national security (Alqahtani and Sheldon, 2022).

- Increasing sophistication and defence evasion: Each new generation of ransomware has introduced new techniques to outmanoeuvre defences. Often modern ransomware disables backups, evades antivirus, and leverages anonymized cryptocurrency payments to avoid tracing (Nagar, 2024). Often described as an "arms race", every phase of this progression has "added new strategies and methodologies that have increased the capability and efficiency of ransomware, and it has become even harder to defend against it" (Nagar, 2024). Fundamentally, ransomware has evolved into an adaptive, "continuous threat" which requires equally adaptive countermeasures (Nagar, 2024).

The trends mentioned above clearly present why ransomware is one of "the most notorious malware… with financial loss of billions of dollars" reported by (Nagar, 2024). Its continued growth in complexity and impact generates an increased need for innovative defences, as will be discussed next.

2.2 Machine learning techniques for ransomware detection and mitigation

The evolving and evasive nature of ransomware has required new techniques like machine learning (ML), which show promise in detection and mitigation. ML allows for identification of anomalies in real time, this is dissimilar from traditional signature-based antivirus solutions. Even those associated with previously unseen or obfuscated variants can be detected using this technique, by learning behavioural patterns from system and network data (Alqahtani and Sheldon, 2022a; Fernando, 2023).

2.21 Supervised Learning

Supervised classification techniques are widely applied, being used to train models on labelled examples of ransomware and benign behaviour. Algorithms such as Support Vector Machines, Random Forests, and neural networks are commonly used to analyse file access patterns, API call sequences, and network traffic (Alqahtani and Sheldon, 2022a). Models like these generally achieve high accuracy when trained using historical data; for example, supervised models detecting crypto ransomware via memory and filesystem activity have demonstrated >90% precision and recall. Sgandurra et al.'s approach, which used behavioural data from the first 30 seconds of execution, enabled rapid identification before encryption began. This demonstrates how time-sensitive patterns can support early-stage mitigation (Alqahtani and Sheldon, 2022b).

2.22 Unsupervised and Anomaly Detection

Unsupervised techniques, such as Isolation Forests or clustering algorithms, are particularly useful for identifying novel or zero-day ransomware. These models learn a baseline of normal system behaviour and flag deviations, like unusual file modifications or spikes in CPU/network activity, as potential threats (Hyuk et al., 2023). This differs from the supervised approach, not relying on prior labelled data, which makes them ideal for detecting emergent or stealthy ransomware. However, false positives still pose an issue, meaning they require careful calibration to balance sensitivity and precision.

2.23 Hybrid Methods and Behavioural Monitoring

A combination of both static and dynamic analysis, taking advantage of both code inspection and runtime behaviour to improve robustness. ML models can be trained on multiple drastically different feature sets or operate in ensemble configurations, this helps to improve generalisability and resilience to evasion attempts. Modern systems often implement layered ML pipelines, where a fast lightweight model flags suspicious activity and a deeper analysis follows if needed (Alqahtani and Sheldon, 2022b). Feedback loops for periodic retraining are also becoming common, helping the system adapt to new threats over time.

2.24 Challenges in ML-Based Detection

While ML techniques outperform many traditional tools, several challenges remain:

Evasion Techniques: Attackers using ransomware are increasingly taking advantage of tactics like delayed encryption, process masquerading, or activity throttling to bypass behaviour-based systems. Alqahtani and Sheldon (2022), "evasive mechanisms… often nullify the solutions that are currently in place." ML models can become obsolete as attackers evolve their tactics.

- Concept Drift: The statistical nature of ransomware behaviour shifts over time; this is a phenomenon known as concept drift. If models are not updated regularly, their performance degrades. Fernando (2023) highlights this in their FeSAD system, which dynamically updates feature sets to maintain detection rates across changing ransomware variants from 2016–2020. Without such adaptation, false negatives can increase over time as ransomware attacks adjust their approach.

- Data limitations: Access to high-quality, labelled ransomware datasets is not easy and continues to remain a barrier. Real-world data is often sensitive, limited, or proprietary, and may not reflect the latest threats. This constrains model training and validation (Hyuk et al., 2023). Synthetic datasets or testbeds like CIC-IDS2018 offer partial solutions but still lack complete real-world coverage (Zhang et al., 2025).

2.25 Relevance to this project

The Random Forest (supervised) and Isolation Forest (unsupervised) models were selected due to their widespread use, interpretability and resilience to noise. Both models were trained on synthetic and real-world datasets, with ransomware behaviours represented through elevated CPU usage, file modification and abnormal network activity. This aligns with the detection goals identified in the literature and reflects best practices for early-stage behavioural ransomware detection (Fernando, 2023; Alqahtani and Sheldon, 2022a).

Even though this project does not implement advanced adaption mechanisms like online learning, a solid foundation is laid to allow for enhancements, done through structuring the data pipeline to support retraining. Furthermore, the discussed limitations are reflected through the real-world model’s lower recall performance, helping to show the importance of continuous dataset expansion and system refinement.

2.3 Blockchain for tamper-proof event logging and data integrity

Blockchain technology has emerged as a promising tool for enhancing cybersecurity helping to preserve the integrity of logs and other critical data. This technology ensures that once data is recorded, it becomes extremely difficult to modify or erase without detection. This is done using an append-only ledger secured by cryptographic hashing and consensus mechanisms. These features make blockchain particularly well-suited for securing forensic records in the event of a ransomware attack.

A prominent application is immutable logging. Austin and Di Troia (2023) proposed a tamper-resistant logging framework using a private blockchain, where proof-of-work is applied to blocks of log messages. Even if an attacker gains administrative access, modifying log entries becomes infeasible without breaking the blockchain structure (Austin & Di Troia, 2023). This ensures strong non-repudiation and helps retain trustworthy evidence, addressing the well-known issue of ransomware deleting or corrupting logs to evade detection and analysis.

Another key use case is data integrity verification. Morillo Reina et al. (2023) developed a system in which critical logs are hashed and recorded on public blockchains such as Cardano or Solana. Any unauthorized changes to local data can be detected by comparing hashes against the on-chain version. This model guarantees "data integrity, immutability, and non-repudiation" and adds an additional layer of security, even against insider threats who might otherwise tamper with sensitive logs (Morillo Reina et al., 2023).

Blockchain logging is especially resilient to ransomware's common tactic of encrypting or deleting logs. By continuously writing telemetry or detection metadata to an external blockchain ledger, defenders can retain tamper-proof evidence, even if the local system is compromised (Morillo Reina et al., 2023). This logging method aligns with Zero Trust principles and has been shown to function effectively in systems where event volume is moderate and near-real-time integrity is crucial.

2.4 Integrating Machine Learning with Blockchain for Ransomware Defence

While blockchain ensures data integrity and traceability, machine learning enables intelligent threat detection and response. Combining both technologies for a dual-layered defence strategy has gained attention. This is particularly useful for ransomware attacks where fast detection and resilient logging are crucial.

Oz et al. (2022) identify artificial intelligence and blockchain as "the most prominent and promising solutions for ransomware-based cyber-attack detection.", with ML models offering rapid anomaly detection based on behavioural deviations—such as unusual CPU spikes or file modifications—while blockchain preserves detection outputs and system logs in a tamper-evident format (Oz et al., 2022). The combination of the two provides proactive identification and trustworthy record-keeping, helping to improve real-time response and post-incident analysis.

Sharma et al. (2024) provides a working example of this integration in IoT security. Their framework uses a deep learning model for intrusion detection and records detection logs immutably using a blockchain backend. The result of this was a system with a 98.9% detection accuracy and "tamper-proof logs of detected intrusions," this demonstrates how blockchain and ML can work together, blockchain reinforcing ML by securing the output of detection models (Sharma et al., 2024).

Other studies indicate blockchain technology can address common limitations of ML, such as concept drift or poisoning attacks. Sharing updated ransomware signatures or model parameters via a distributed ledger enables secure collaboration without centralized trust. What's more, ML can analyse blockchain transaction patterns (such as cryptocurrency ransom payments) to detect ransomware indirectly (Kayikci & Khoshgoftaar, 2024). Showing how integration can be mutually beneficial.

Despite showing promise, this dual-layered system introduces complexity. Performance overhead, privacy concerns, and system interoperability are ongoing challenges (Kayikci & Khoshgoftaar, 2024). However, the feasibility demonstrated in early prototypes provides a strong foundation for research into frameworks that unify intelligent detection with resilient data integrity.

2.5 Summary and research gap

The literature shown highlights the complementary strengths of both machine learning and blockchain technology for defence against ransomware threats. Machine learning enables intelligent, behaviour-based detection, and analysing anomalies in both system and network activity, making it well-suited for identifying both known and novel ransomware variants (Alqahtani and Sheldon, 2022a). Techniques such as supervised learning (e.g. Random Forests, neural networks) and unsupervised anomaly detection (e.g. Isolation Forest, clustering) have been shown to outperform traditional signature-based detection, often achieving high accuracy rates in experimental settings (Fernando, 2023; Hyuk et al., 2023). Limitations remain, however particularly concerning model evasion, concept drift, and limited access to real-world datasets (Alqahtani and Sheldon, 2022b; Fernando, 2023).

Blockchain, on the other hand, provides a robust solution for preserving the integrity of logs and forensic evidence. Its tamper-resistant, append-only structure ensures that once data is written, it cannot be altered or erased without detection, making it ideal for secure logging in cyberattack scenarios (Austin and Di Troia, 2023). Public and private blockchain implementations have been explored to store log hashes and event metadata, helping to prevent both insider and external tampering (Morillo Reina et al., 2023). This is especially relevant in ransomware contexts, where malware often targets system logs to conceal its presence.

Despite strong individual potential, current research often treats these technologies separately. Few practical implementations effectively integrate machine learning detection with blockchain-based evidence preservation, and even fewer apply this combination specifically to ransomware defence. This represents a clear gap in the literature. As Oz et al. (2022) and Sharma et al. (2024) highlight, the convergence of ML and blockchain can provide multi-layered resilience — combining proactive threat identification with tamper-proof post-attack analysis — however, this remains underdeveloped in practice.

This project directly addresses that gap by designing and evaluating a dual-layered ransomware defence framework. The first layer employs machine learning models — including an Isolation Forest and Random Forest classifier — tested against both simulated ransomware behaviour and real-world data from the CSE-CIC-IDS2018 dataset. The second layer introduces a blockchain-based proof of concept, logging detection events immutably using smart contracts to ensure that critical alerts and telemetry are not lost or altered during an attack. This integrated approach aims to deliver a technically sound and forensically reliable solution, aligning with recent calls for adaptable, tamper-resistant cyber defence systems (Alqahtani and Sheldon, 2022a; Kayikci and Khoshgoftaar, 2024).

3.0 Methodology

3.1 Overview of proposed approach

This project proposes a dual-layered ransomware defence framework combining both machine learning (ML) for real-time detection and blockchain technology for tamper-resistant event logging. The ML component aims to identify early indications of ransomware activity through behavioural analysis, whilst the blockchain layer preserves critical system logs in an immutable format. This section focuses on the design, implementation, and evaluation of the machine learning models used.

3.2 Dataset Selection and Preparation

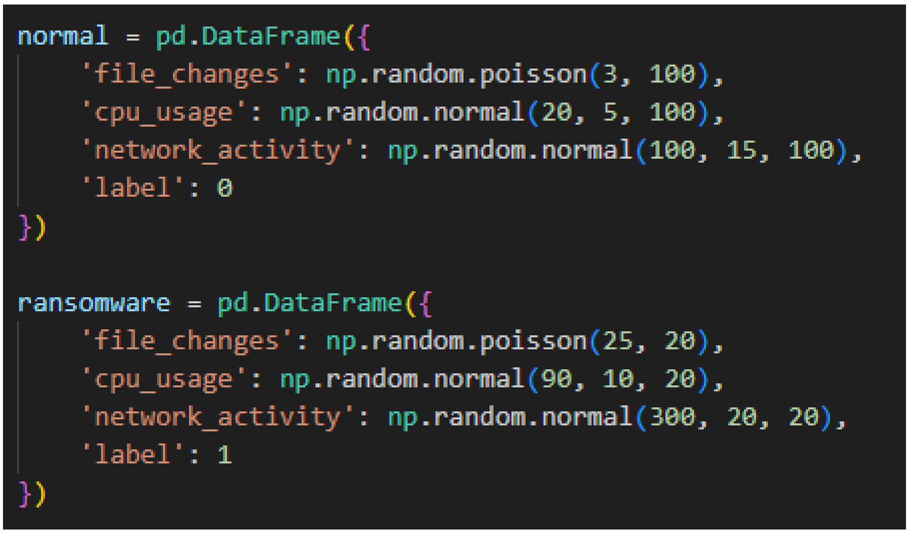
3.21 Simulated dataset

A synthetic dataset was generated, simulating typical ransomware activity in a controlled environment. This dataset consisted of two classes, benign and ransomware behaviour. The three chosen features were generated using random distributions, representing system and network activity.

- file_changes: Simulated using a Poisson distribution (λ=3 for normal, λ=25 for ransomware).

- cpu_usage: Simulated using a normal distribution (μ=20, σ=5 for normal; μ=90, σ=10 for ransomware).
- network_activity: Simulated using a normal distribution (μ=100 for normal; μ=300 for ransomware).

These features were selected to balance interpretability with computational efficiency and represent key behaviours commonly manipulated by ransomware. Using Synthetic data generation is a common practice in cybersecurity research, allowing testing in a controlled setting without the risk of testing live ransomware (Fernando, 2023).



```
normal = pd.DataFrame({
    'file_changes': np.random.poisson(3, 100),
    'cpu_usage': np.random.normal(20, 5, 100),
    'network_activity': np.random.normal(100, 15, 100),
    'label': 0
})

ransomware = pd.DataFrame({
    'file_changes': np.random.poisson(25, 20),
    'cpu_usage': np.random.normal(90, 10, 20),
    'network_activity': np.random.normal(300, 20, 20),
    'label': 1
})
```

Figure 1. example of code representing generation of ransomware and benign data

The generated dataset contains 100 benign and 20 ransomware samples. This allows simple testing to see how affective models can identify outliers or abnormal behaviour patterns.

3.22 Real-World Dataset (CSE-CIC-IDS2018)

To successfully validate the model in a realistic setting, the CSE-CIC-IDS2018 dataset was used this is Widely considered a benchmark in ransomware and intrusion detection (Hyuk et al., 2023; Zhang et al., 2025). Explicitly, the “Thursday” subset was selected, due to it containing ransomware behaviour classified under the Infiltration label. This dataset contains network traffic features such as:

- Flow Duration
- Tot Fwd Pkts
- Tot Bwd Pkts

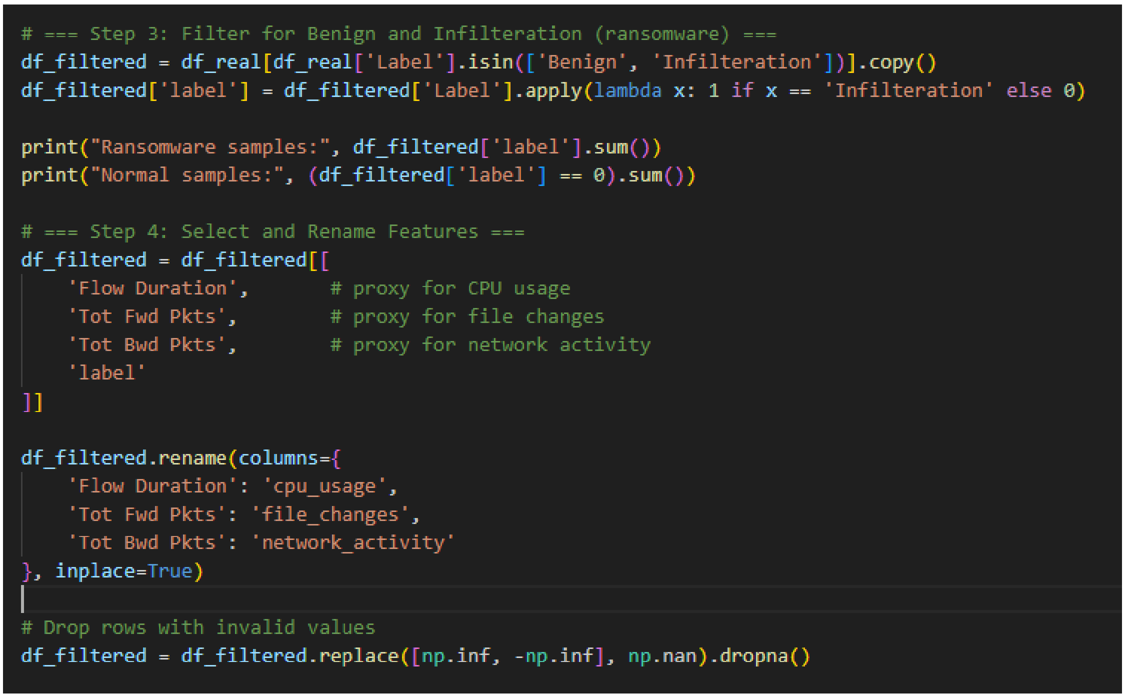
These features were than matched to the simulated ones

- Flow Duration → CPU usage proxy
- Tot Fwd Pkts → File changes proxy
- Tot Bwd Pkts → Network activity

These mappings are based on literature aligning traffic volume and duration with corresponding system-level behaviours (Hyuk et al., 2023).

The real-world dataset was filtered to only include benign and infiltration traffic and samples with missing or infinite values was dropped to improve model reliability.

While the CSE-CIC-IDS2018 dataset is widely recognised, evaluating models across multiple datasets or recent attack variants would further improve generalisability (Zhang et al., 2025).



```
# === Step 3: Filter for Benign and Infilteration (ransomware) ===
df_filtered = df_real[df_real['Label'].isin(['Benign', 'Infilteration'])].copy()
df_filtered['label'] = df_filtered['Label'].apply(lambda x: 1 if x == 'Infilteration' else 0)

print("Ransomware samples:", df_filtered['label'].sum())
print("Normal samples:", (df_filtered['label'] == 0).sum())

# === Step 4: Select and Rename Features ===
df_filtered = df_filtered[[
    'Flow Duration',       # proxy for CPU usage
    'Tot Fwd Pkts',        # proxy for file changes
    'Tot Bwd Pkts',        # proxy for network activity
    'label'
]]

df_filtered.rename(columns={
    'Flow Duration': 'cpu_usage',
    'Tot Fwd Pkts': 'file_changes',
    'Tot Bwd Pkts': 'network_activity'
}, inplace=True)

# Drop rows with invalid values
df_filtered = df_filtered.replace([np.inf, -np.inf], np.nan).dropna()
```

Figure 2. Code showing the filtering of the real-test data and conversion of features.

3.3 Feature Selection and Preprocessing

The selected features were:

- CPU usage proxy (Flow Duration)
- File change proxy (Tot Fwd Pkts)
- Network activity (Tot Bwd Pkts)

All features were standardised where necessary, and labels were binary encoded:

- 0 for normal activity
- 1 for ransomware

In both datasets, rows with missing or invalid values were removed. The real-world data was especially noisy, requiring careful cleaning.

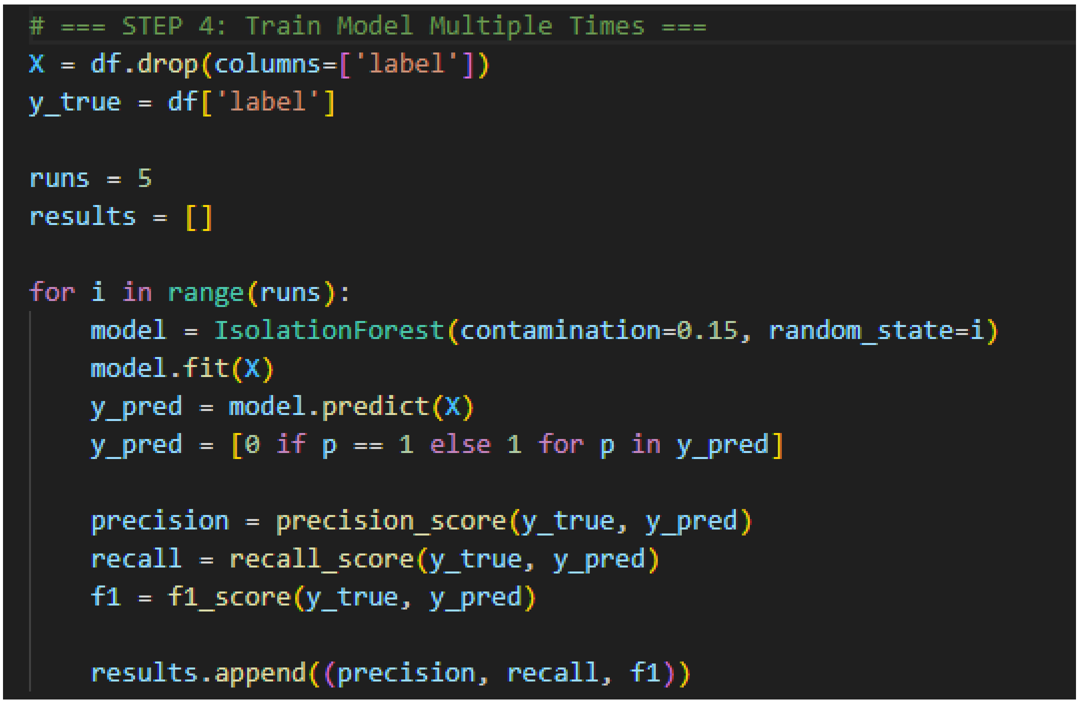
3.4 Model Selection

To evaluate numerous machine learning strategies, two different models were implemented:

Isolation Forest (unsupervised)

- Designed specifically for anomaly detection without relying on labelled training data
- Trained on benign data, the model learns normal behaviour and flags any significant deviations as anomalies
- Output predictions are converted to binary:
    - 1 for inliers
    - 0 for outliers

This model was chosen due to its ability to catch novel, previously unseen threats. This is an extremely valuable trait in evolving ransomware scenarios.



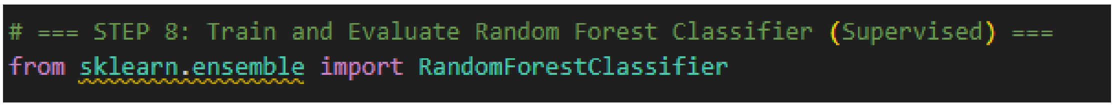
```python
# === STEP 4: Train Model Multiple Times ===
X = df.drop(columns=['label'])
y_true = df['label']

runs = 5
results = []

for i in range(runs):
    model = IsolationForest(contamination=0.15, random_state=i)
    model.fit(X)
    y_pred = model.predict(X)
    y_pred = [0 if p == 1 else 1 for p in y_pred]

    precision = precision_score(y_true, y_pred)
    recall = recall_score(y_true, y_pred)
    f1 = f1_score(y_true, y_pred)

    results.append((precision, recall, f1))
```

Figure 3. Importing and looping example of Isolation Forest (unsupervised).

Random Forest (Supervised)

- Widely used ensemble learning classifier based on decision trees
- Trained on the labelled data to classify samples directly as normal or ransomware
- Offers a robust interpretable feature importance, is generally less sensitive to noise and class imbalance

Isolation Forest and Random Forest were chosen for their robustness, interpretability, and ability to handle noise and outliers, which is especially important in intrusion detection contexts. Both have been consistently favoured in recent literature due to their strong performance in ransomware detection scenarios (Alqahtani & Sheldon, 2022; Fernando, 2023).



```
# === STEP 8: Train and Evaluate Random Forest Classifier (Supervised) ===
from sklearn.ensemble import RandomForestClassifier
```

Figure 4. Importing random forest machine learning model.

3.5 Model Evaluation and Metrics

Models were evaluated using standard classification metrics:

- Precision: How many predicted ransomware samples were ransomware
- Recall: How many actual ransomware samples were detected
- F1 Score: Harmonic mean of precision and recall
- Accuracy: Overall correct classifications
- AUC (Area Under Curve): Evaluates model performance at various threshold levels

To test robustness, the Random Forest model is run 5 times to rest robustness, with different seeds run each time. The isolation forest in comparison was only tested once, however due to its deterministic behaviour and fixed input data producing the same results every run. Visualisations included:

- Confusion matrix heatmaps
- ROC curves
- Scatter plots showing data separability

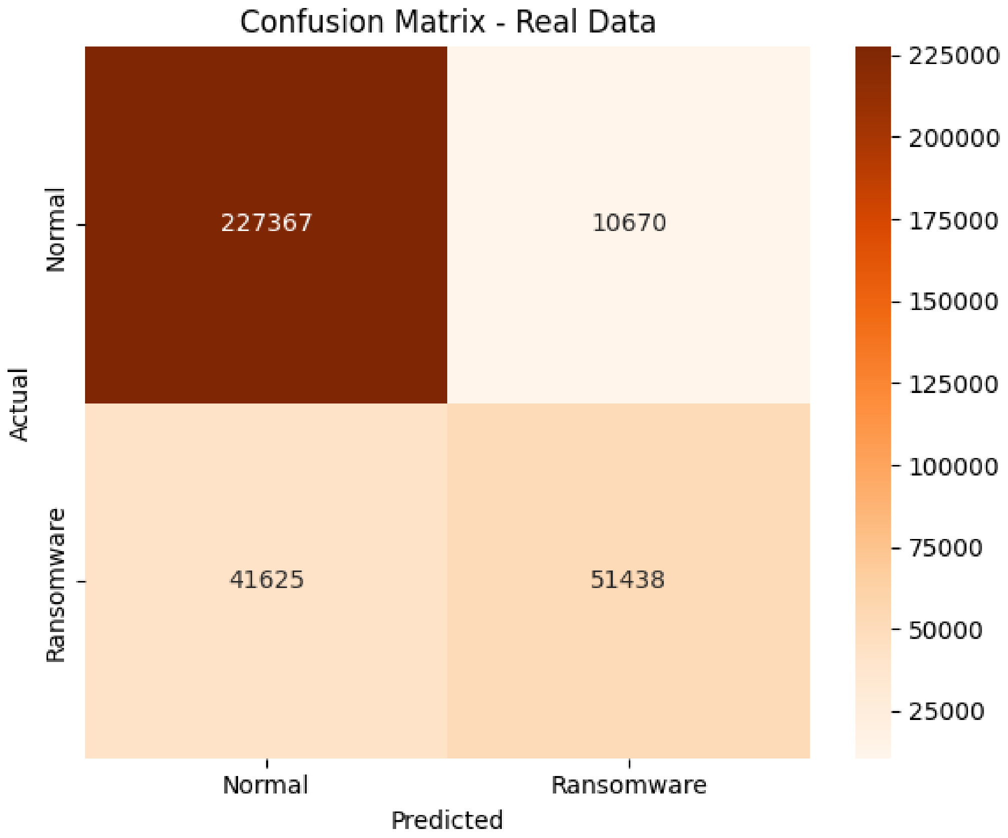


Figure 5. confusion matrix example from results.

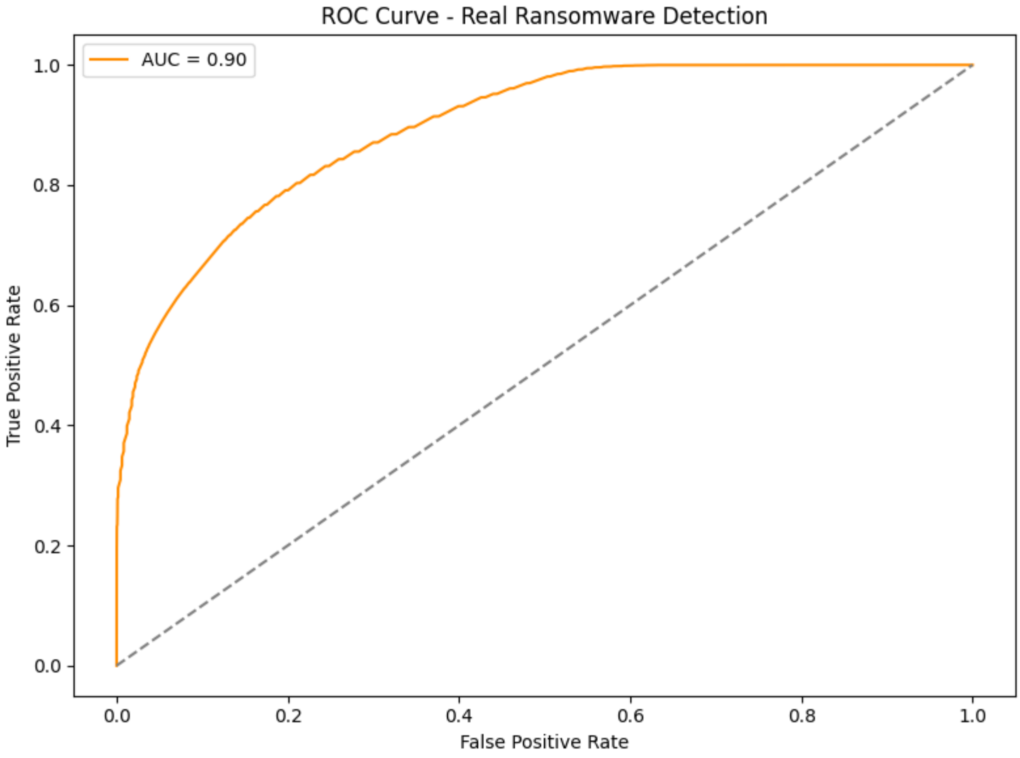


Figure 6. ROC curve example from results.

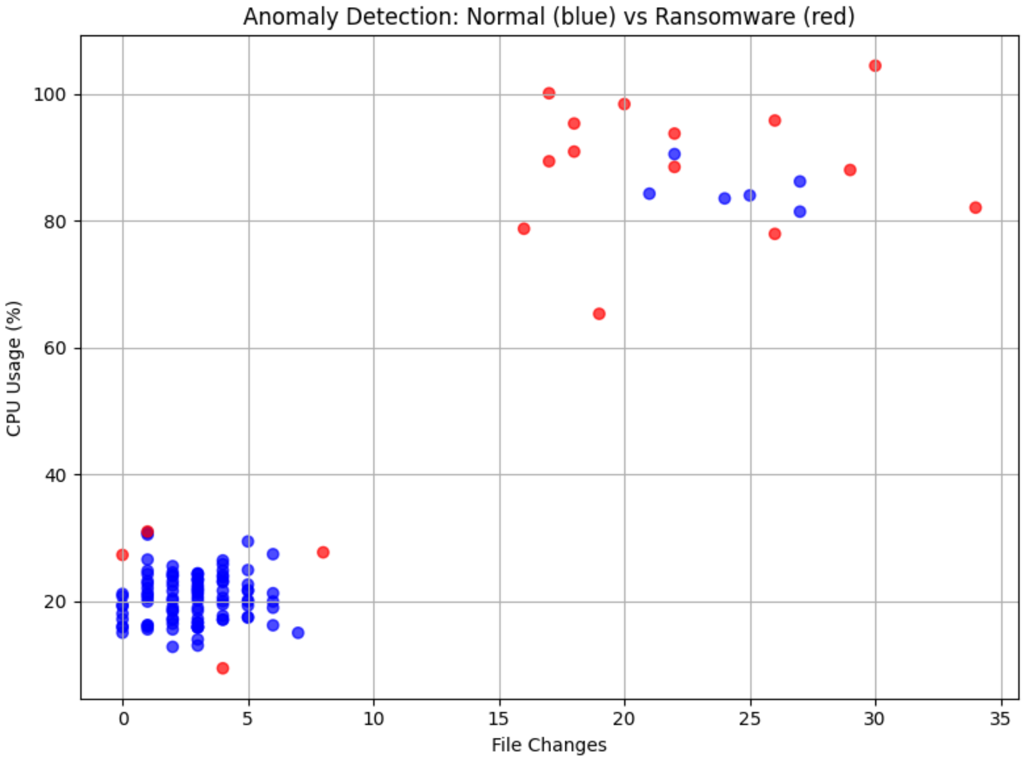


Figure 7. scatter graph showing data separability.

3.6 Blockchain Based Logging System Design

To work alongside the machine learning detection system, this project aims to benefit from the tamper resistant forensic logging that a blockchain based logging system can provide. The aim is to ensure that once suspicious activity is detected, relevant metadata (e.g., timestamps, detection summaries) is recorded immutably, preserving critical information, even if an attacker later compromises or wipes the local system.

This aligns with Zero Trust principles, which assume no system component is inherently secure and all interactions must be verified (Kayıkçı & Khoshgoftaar, 2024) and addresses one of the major gaps in traditional ransomware defences: log tampering.

3.7 Design rationale

Blockchain was integrated due to its key security properties:

- Immutability: once written, logs cannot but altered without detection
- Decentralised Trust: doesn't rely on a single machine, or administrator to preserve critical logs
- Transparency and Traceability: the ability to provide verifiable and timestamped records, pivotal in forensic and compliance contexts

This aligns with Zero Trust Principles, where the integrity of forensic records is not assumed, but enforced by distributed consensus.

3.8 Development Environment

Due to this System being a PoC the system was deployed on a local hardhat blockchain, which eliminated concerns like gas fees and latency. Hardhat offers:

- Fast block times, transactions confirmed almost instantly on the local network
- no-cost transactions
- Multiple test accounts for simulating real users/systems
- Seamless integration with solidity and web3.py

While Hardhat is not a production-grade blockchain, it provides a realistic simulation of Ethereum's cryptographic structure, enabling safe and cost-free prototyping of on-chain functionality. Also, it is being widely used in recent Ethereum development due to its support for local testnets, account simulation and fast compilation, (Austin & Di Troia, 2023) making it ideal for us to test on.

This simulated environment served as a controlled setting for validating functionality, future work could extend the deployment to public testnets (e.g. Goerli) or private blockchains (e.g. Quorum) for saleability testing.

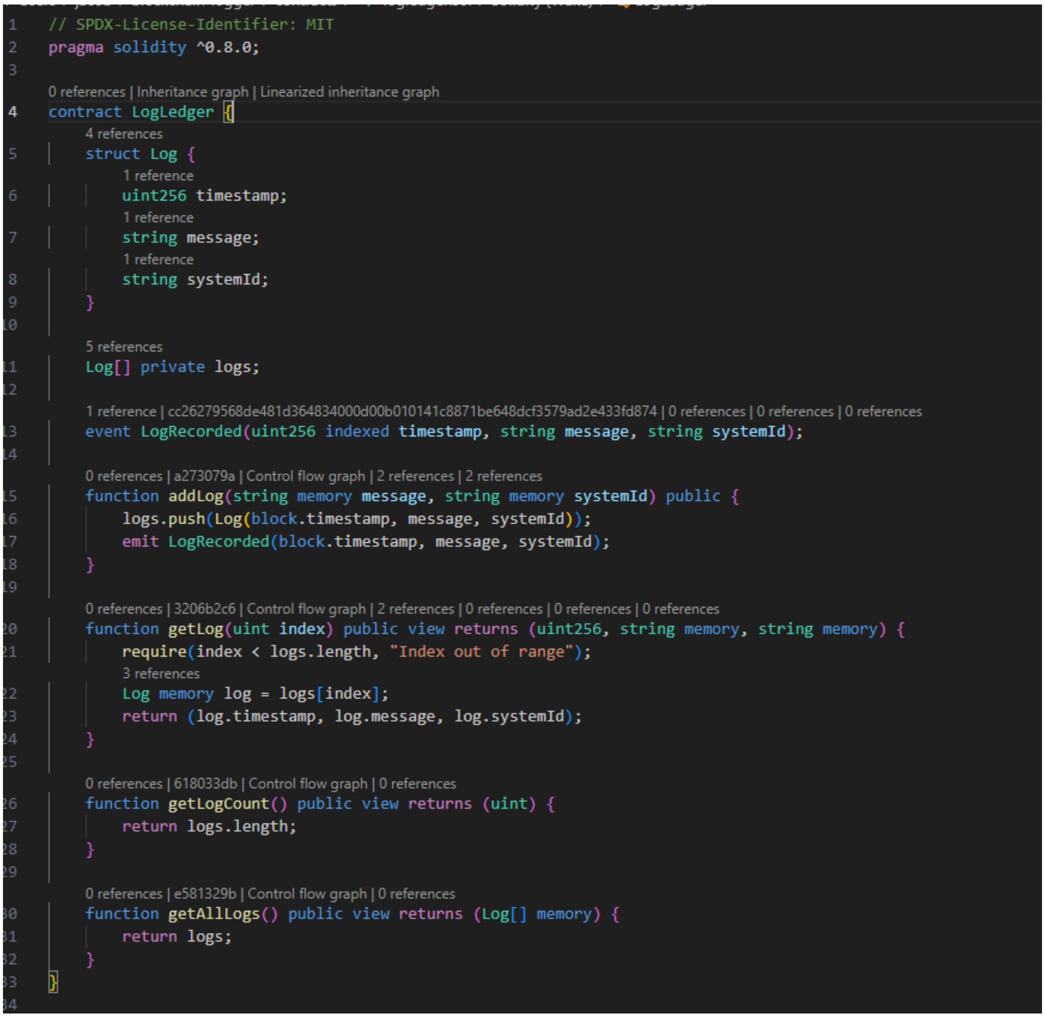
3.9 Smart Contract Design

A smart contract was written in Solidity and deployed using Hardhat. Its primary responsibilities include:

- addLog(message, systemId): Records a log entry with the timestamp, detection message, and system identifier
- getLog(index) and getLogCount(): Enables log retrieval and auditing

Each log entry consists of:

- timestamp (when the detection occurred)
- message (e.g., "Ransomware detected - file spike")
- systemId (identifier of the detection system)



```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract LogLedger {
    struct Log {
        uint256 timestamp;
        string message;
        string systemId;
    }

    Log[] private logs;

    event LogRecorded(uint256 indexed timestamp, string message, string systemId);

    function addLog(string memory message, string memory systemId) public {
        logs.push(Log(block.timestamp, message, systemId));
        emit LogRecorded(block.timestamp, message, systemId);
    }

    function getLog(uint index) public view returns (uint256, string memory, string memory) {
        require(index < logs.length, "Index out of range");
        Log memory log = logs[index];
        return (log.timestamp, log.message, log.systemId);
    }

    function getLogCount() public view returns (uint) {
        return logs.length;
    }

    function getAllLogs() public view returns (Log[] memory) {
        return logs;
    }
}
```

0 references | Inheritance graph | Linearized inheritance graph
4 references
1 reference
1 reference
1 reference
5 references
1 reference | cc26279568de481d364834000d00b010141c8871be648dcf3579ad2e433fd874 | 0 references | 0 references | 0 references
0 references | a273079a | Control flow graph | 2 references | 2 references
0 references | 3206b2c6 | Control flow graph | 2 references | 0 references | 0 references | 0 references
3 references
0 references | 618033db | Control flow graph | 0 references
0 references | e581329b | Control flow graph | 0 references

Figure 8. LogLedger, shows smart contract design.

Logs are stored in an on-chain array, making them immutable and queryable.

For simplicity, the smart contract currently allows unrestricted logging from any sender. Future versions could implement role-based access control or authentication mechanisms to restrict log submission to authorised systems. Also, to maintain on-chain efficiency and avoid exposing sensitive data, only high-level metadata was stored. Full logs or files were excluded due to privacy concerns and the high cost of storing large data on-chain.

3.10 Integration and Detection System

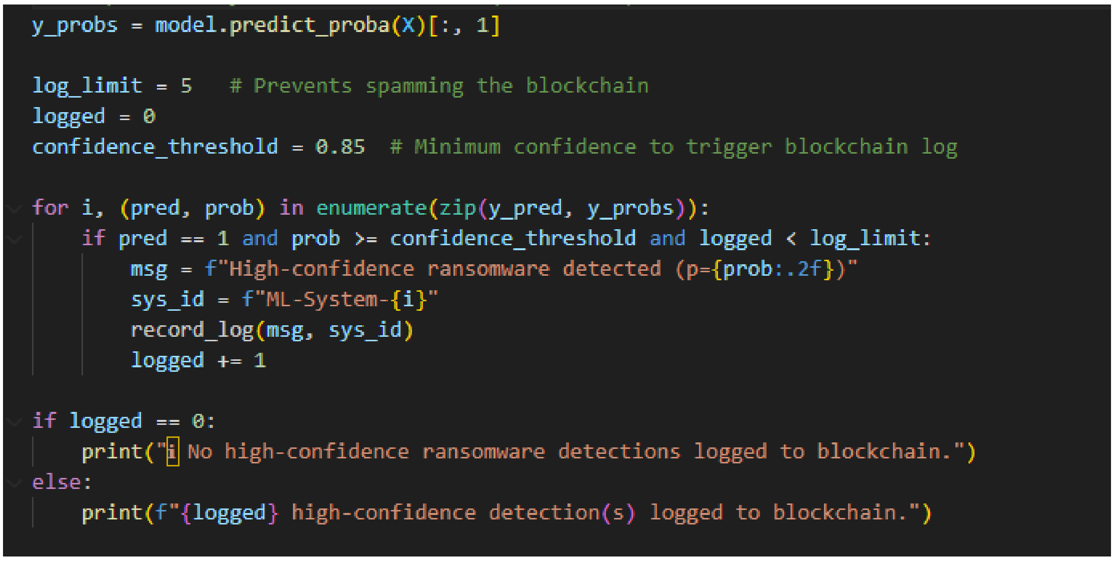
A Python interface was developed using web3.py, allowing real-time communication between the ML system and the deployed smart contract. When the ML model classifies an event as ransomware, it:

1. Constructs a metadata message
2. Sends a transaction to the blockchain smart contract
3. Waits for confirmation (tx receipt)

Real time logging was chosen over batching to simulate immediate forensic capture upon detection. Whilst this is perfectly suitable for this demonstration, real-world deployments could implement event batching or buffered writes to reduce transaction volume and improve throughput.

The decision to log only the first 5 detections per run was made for:

- Performance control: Each transaction increases latency; limiting the number guarantees tests run quickly and don’t flood the chain.
- Resource efficiency: Each transaction costs money in a real-world system, limiting the number of entries helps to retain the realistic functionality without overwhelming the system or incurring necessary overhead
- Demonstration clarity: A smaller number of logs allows for cleaner output, making it easier to demonstrate functionality



```python
y_probs = model.predict_proba(X)[:, 1]

log_limit = 5   # Prevents spamming the blockchain
logged = 0
confidence_threshold = 0.85  # Minimum confidence to trigger blockchain log

for i, (pred, prob) in enumerate(zip(y_pred, y_probs)):
    if pred == 1 and prob >= confidence_threshold and logged < log_limit:
        msg = f"High-confidence ransomware detected (p={prob:.2f})"
        sys_id = f"ML-System-{i}"
        record_log(msg, sys_id)
        logged += 1

if logged == 0:
    print("ℹ No high-confidence ransomware detections logged to blockchain.")
else:
    print(f"{logged} high-confidence detection(s) logged to blockchain.")
```

Figure 9. python code sending logs when high confidence ransomware is detected limited to 5.

This limit reflects a practical trade-off between demonstration clarity and simulated performance, under constraints similar to real-world transaction limits.

In a production environment, dynamic thresholds, rate-limiting, or an event batching strategy would replace this static limit.

3.11 Limitations and Security Assumptions

- The system only stores metadata, not raw logs or large files, to maintain efficiency.
- Logs are public on the local chain; in a real-world deployment, privacy-preserving techniques (e.g., hashing sensitive fields) would be needed.
- The private key for sending transactions is stored locally; secure key management was out of scope but is critical in production.
- The prototype assumes the blockchain itself is secure and available, and that transactions complete successfully.

3.12 Summary of Proof of Concept

This Dual-layered Proof of Concept presented combines machine learning (ML) with blockchain technology, delivering both proactive detection and tamper-proof logging of ransomware threats. The ML aspect leverages behavioural features (CPU usage, file modification, and network activity) to identify ransomware in both synthetic and real-world data, using both supervised and unsupervised models. When an attack is detected, the blockchain component is triggered to immutably record a forensic log, ensuring critical alerts are preserved even when the local system is compromised. This integration demonstrates how combining intelligent threat detection with immutable evidence storage can enhance cyber resilience, address real-world gaps such as log deletion, and support post-incident analysis. The system was tested end-to-end, proving technically feasible within a controlled environment and offering a foundation for scalable real-world deployment. While prior research has explored ML or blockchain individually for ransomware detection, this proof of concept demonstrates a working integration of both, contributing a practical approach to intelligent and forensically robust defences (Oz et al., 2022).

4.0 Implementation and Results

4.1 Overview

This section presents the technical implementation and evaluation of the proposed dual-layered ransomware defence system. This system integrates a behaviour-based machine learning (ML) detection pipeline with a blockchain-based tamper-resistant logging layer. Both components were tested using controlled synthetic datasets and real-world network traffic from the CSE-CIC-IDS2018 benchmark dataset. Evaluation is focused on detection performance, consistency across repeated runs, and system resilience, including evidence integrity through backchain logging.

4.2 Environment and tools

The system was implemented on a Windows 11 machine using Visual Studio Code, Python 3.10, and a virtual environment (venv). Core ML libraries included scikit-learn, pandas, matplotlib, and seaborn. The blockchain component was developed using solidity, web3.py, and a local Hardhat Ethereum testnet for fast, gas-free deployments (Austin & Di Troia, 2023).

Simulated Dataset and Model Implementation

To establish a baseline for detection accuracy in a controlled environment, a synthetic dataset was generated. This included:

- file_changes: Simulated with a Poisson distribution (λ=3 for benign, λ=25 for ransomware)
- cpu_usage: Simulated with a Normal distribution (μ=20, σ=5 for benign; μ=90, σ=10 for ransomware)

- network_activity: Simulated with a Normal distribution (μ=100 for benign; μ=300 for ransomware)

These features represent observable ransomware behaviour such as rapid file encryption, CPU spikes, and potential exfiltration (Fernando, 2023). The dataset contained 120 samples (100 benign, 20 ransomware), balancing interpretability with the realism of imbalanced threat detection.

4.3 Model Selection

Two models were implemented:

- Isolation Forest (Unsupervised): exclusively trained on benign data, this model identifies outliers by learning the boundary of "normal" behaviour, making it suitable for zero-day ransomware (Alqahtani & Sheldon, 2022).
- Random Forest (Supervised): Used as a benchmark. Model trained using known class labels to directly classify inputs as benign or ransomware, providing higher accuracy, but lower generalisability to novel threats.

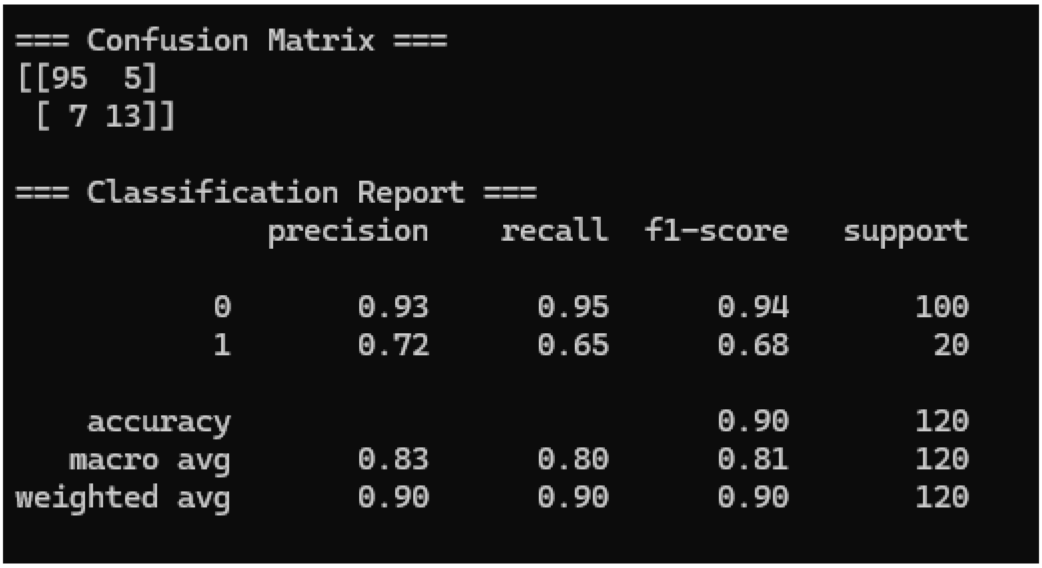
Evaluation Metrics

To evaluate detection quality, five key metrics were used:

- **Precision**: Fraction of predicted ransomware cases that were correct
- **Recall**: Fraction of actual ransomware cases correctly detected
- **F1 Score**: Harmonic mean of precision and recall
- **Accuracy**: Overall correct classifications
- **AUC (Area Under ROC Curve)**: Measures class separability and overall performance

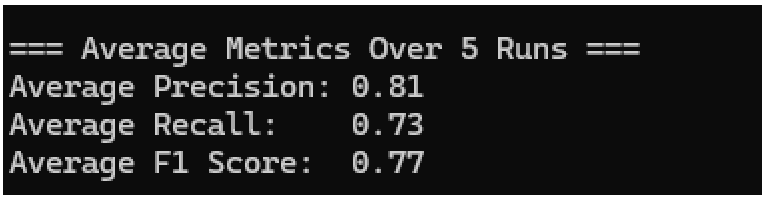
4.31 Isolation Forest (unsupervised)

Results on Simulated Data:



```
=== Confusion Matrix ===
[[95  5]
 [ 7 13]]

=== Classification Report ===
              precision    recall  f1-score   support

           0       0.93      0.95      0.94       100
           1       0.72      0.65      0.68        20

    accuracy                           0.90       120
   macro avg       0.83      0.80      0.81       120
weighted avg       0.90      0.90      0.90       120
```

Figure 10.Isolation Forest (unsupervised results)



```
=== Average Metrics Over 5 Runs ===
Average Precision: 0.81
Average Recall:    0.73
Average F1 Score:  0.77
```

Figure 11. Isolation Forest (unsupervised results) over 5 runs

	Predicted normal	Predicted ransomware
Actual normal (0)	95 true positive	5 false positive
Actual ransomware(1)	7 false negatives	13 true positives

Figure 12. confusion matrix

	precision	recall	F1-score
0 (normal)	0.93	0.95	0.94
1 (ransomware)	0.72	0.65	0.68

Figure 13. percentages of ransomware based of results found in Figure 10

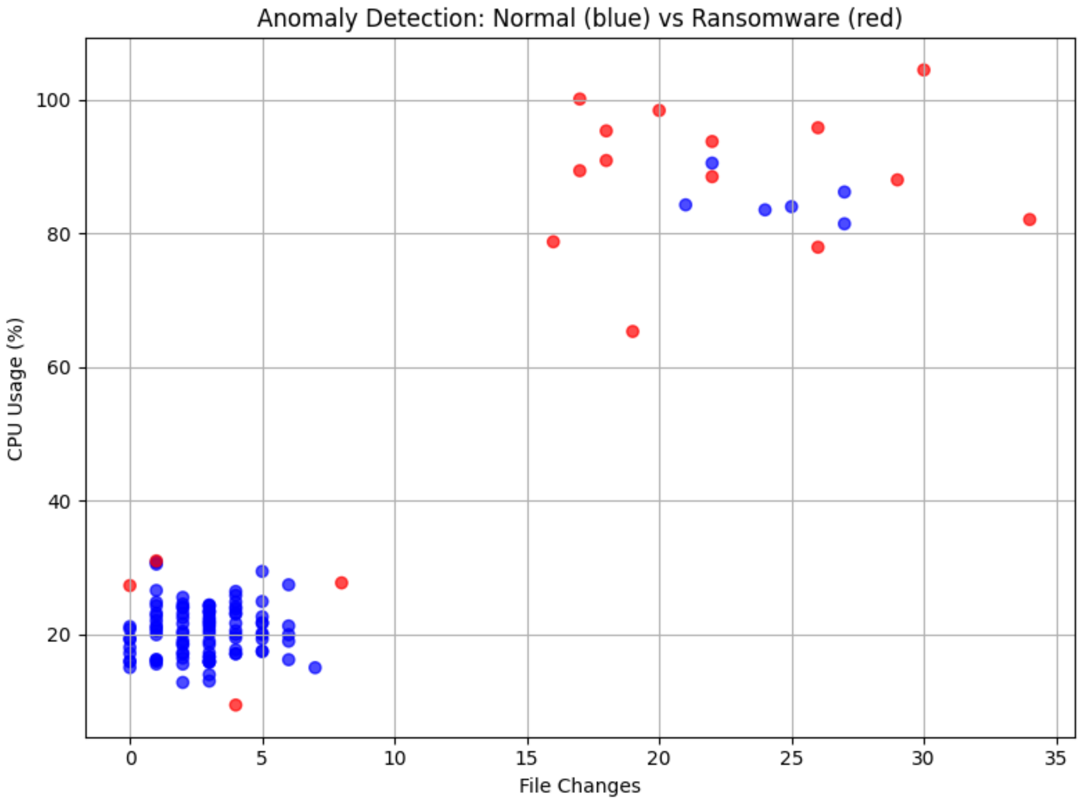


Figure 14. scatter graph of isolation forest (unsupervised) results.

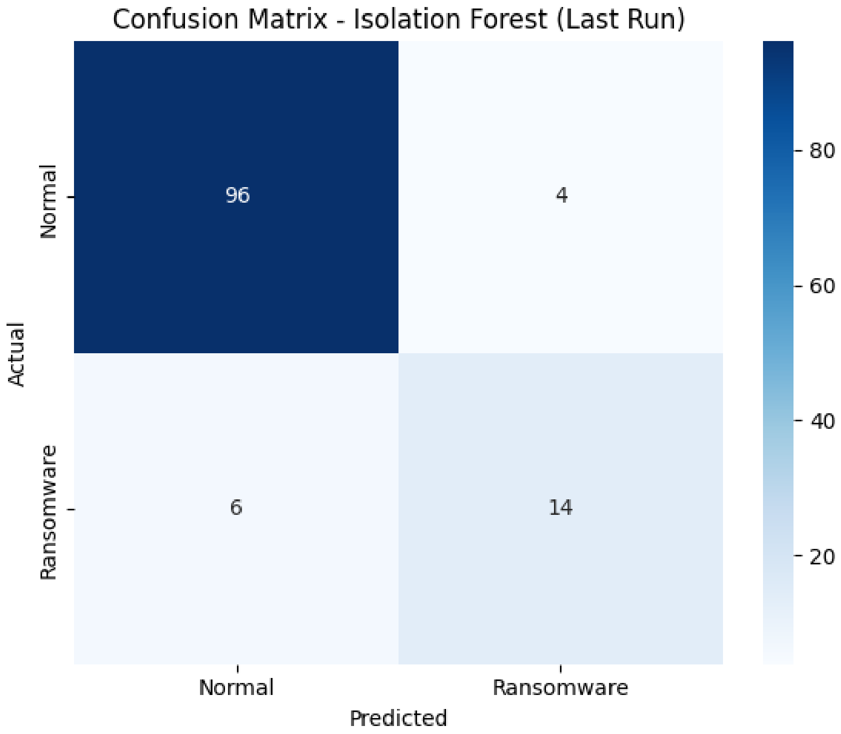
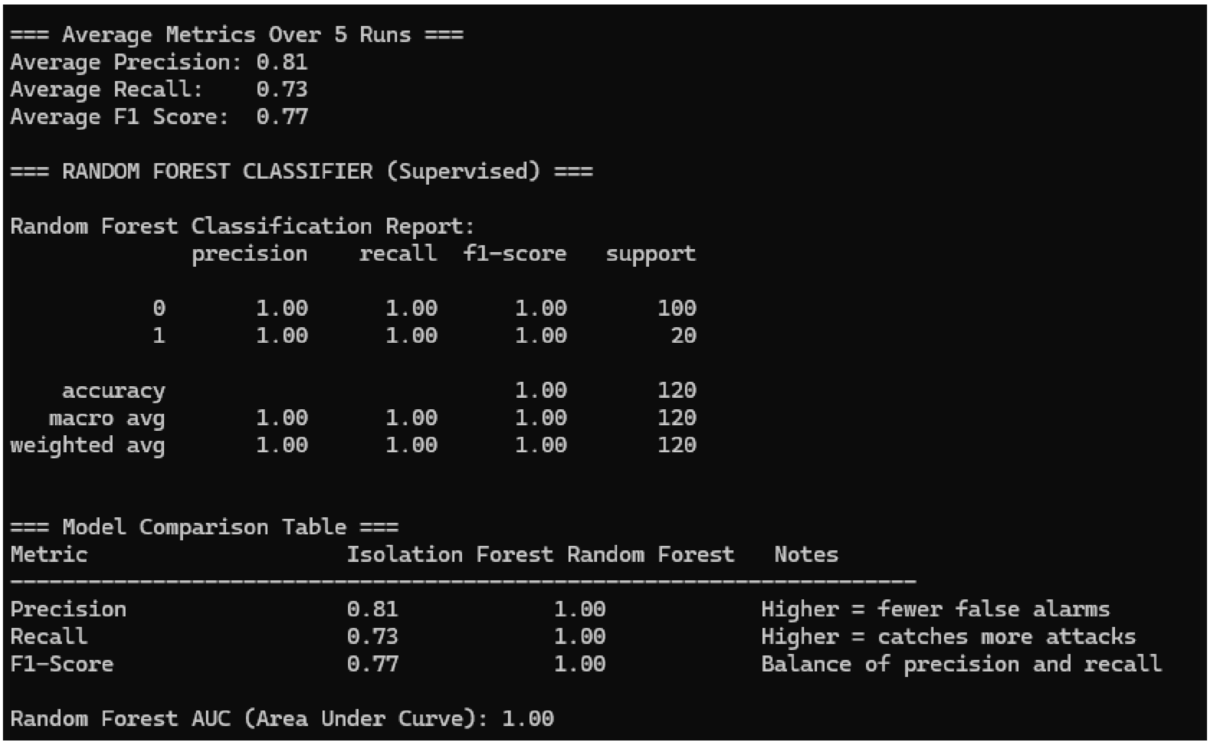


Figure 15. confusion matrix isolation forest (unsupervised)

Since Isolation Forest is a stochastic algorithm, its output may vary slightly between runs. To ensure robustness, I averaged the performance across five iterations.

4.32 Random Forest (Supervised)

The Random Forest model achieved near-perfect performance on the synthetic dataset. This was expected, labelled data simplifies classification, and the environment is noise-free. However, such results are less realistic in live settings where ransomware often evades labelled detection (Alqahtani & Sheldon, 2022).



```
=== Average Metrics Over 5 Runs ===
Average Precision: 0.81
Average Recall:    0.73
Average F1 Score:  0.77

=== RANDOM FOREST CLASSIFIER (Supervised) ===

Random Forest Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       100
           1       1.00      1.00      1.00        20

    accuracy                           1.00       120
   macro avg       1.00      1.00      1.00       120
weighted avg       1.00      1.00      1.00       120


=== Model Comparison Table ===
Metric                    Isolation Forest Random Forest   Notes
----------------------------------------------------------------------
Precision                 0.81            1.00            Higher = fewer false alarms
Recall                    0.73            1.00            Higher = catches more attacks
F1-Score                  0.77            1.00            Balance of precision and recall

Random Forest AUC (Area Under Curve): 1.00
```

Figure 16. Random Forest (Supervised) results.

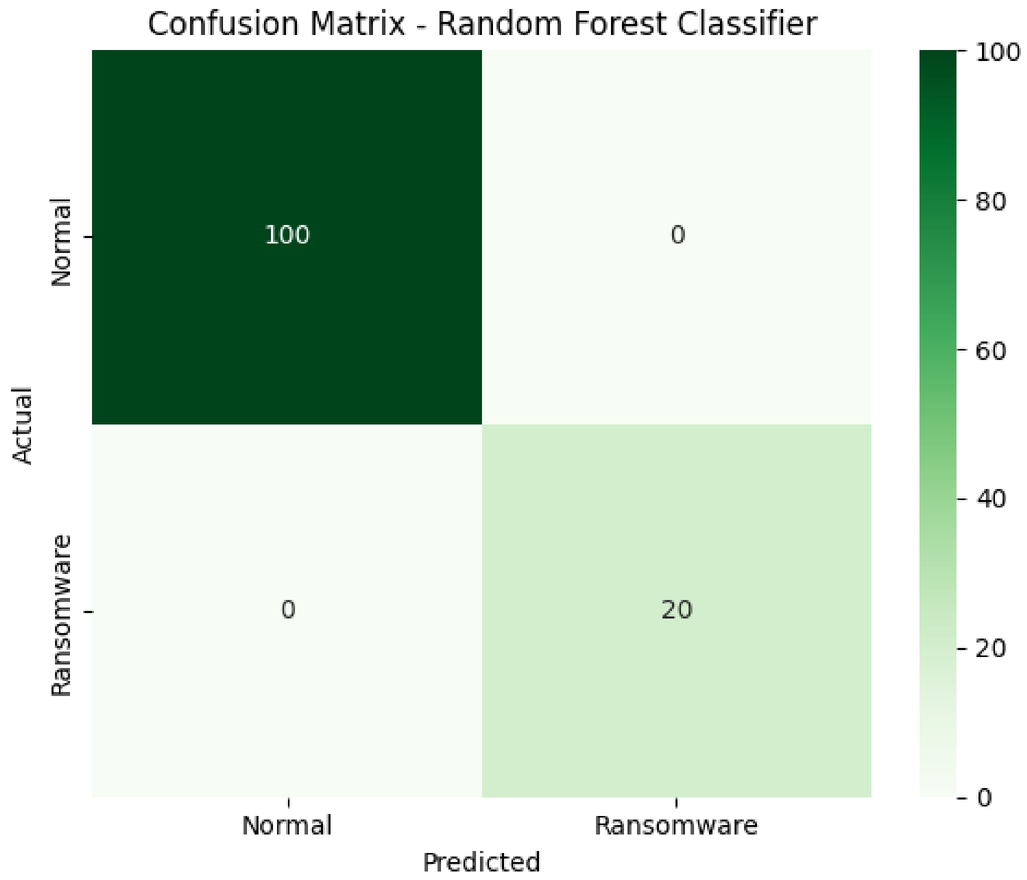


Figure 17. Random Forest (Supervised) confusion matrix

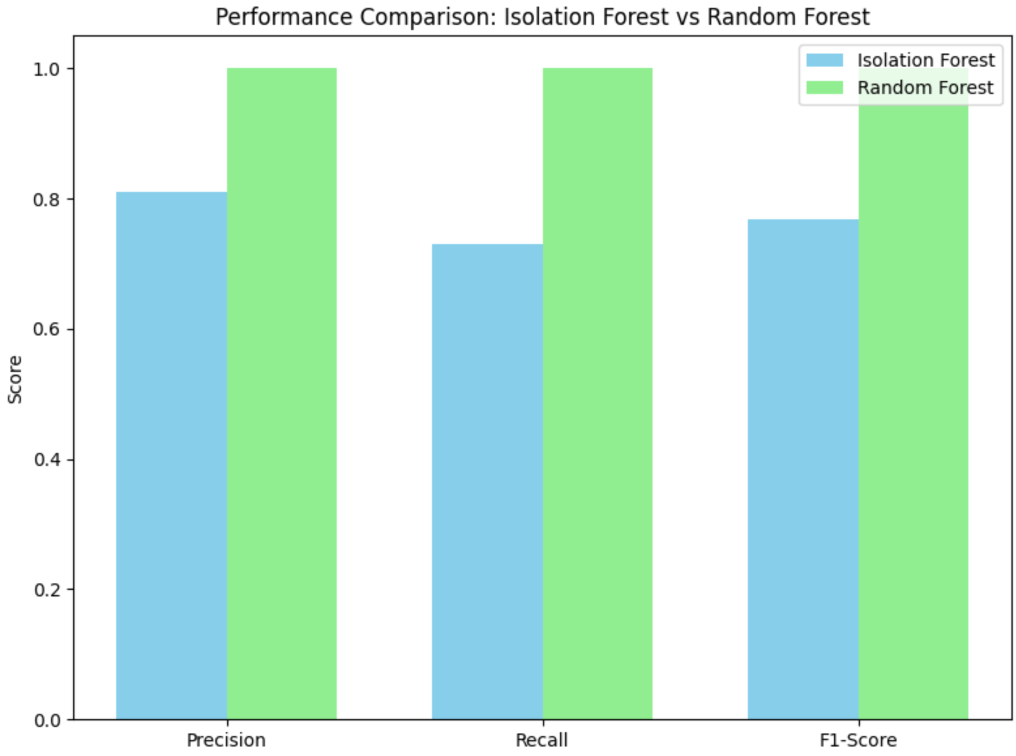


Figure 18. Both Models compared using simulated data

4.33 real-test data (CSE-CIC-IDS2018)

The Thursday subset of the CSE-CIC-IDS2018 dataset was used, specifically the "Infiltration" label containing CryptoLocker-like ransomware activity (Hyuk et al., 2023). Features were mapped to the simulated structure:

- Flow Duration → proxy for CPU usage
- Tot Fwd Pkts → proxy for file changes
- Tot Bwd Pkts → proxy for network activity

After preprocessing the model was tested on thousands of benign and infiltration flows. Key findings.

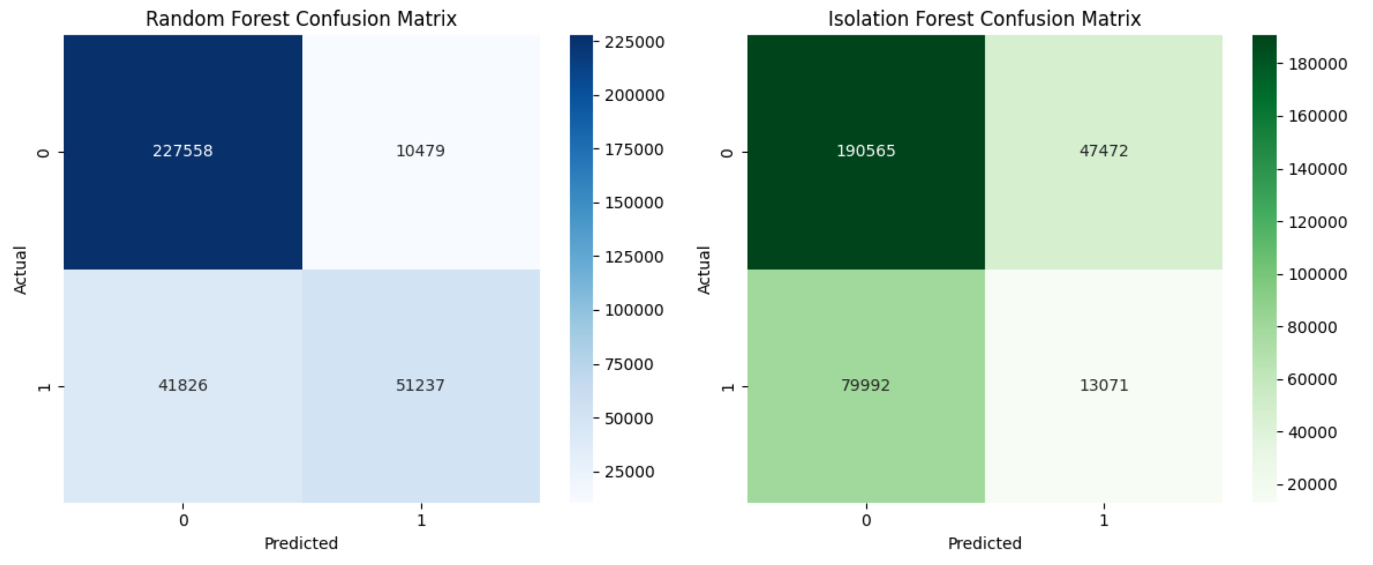


Figure19. confusion matrix comparison both models

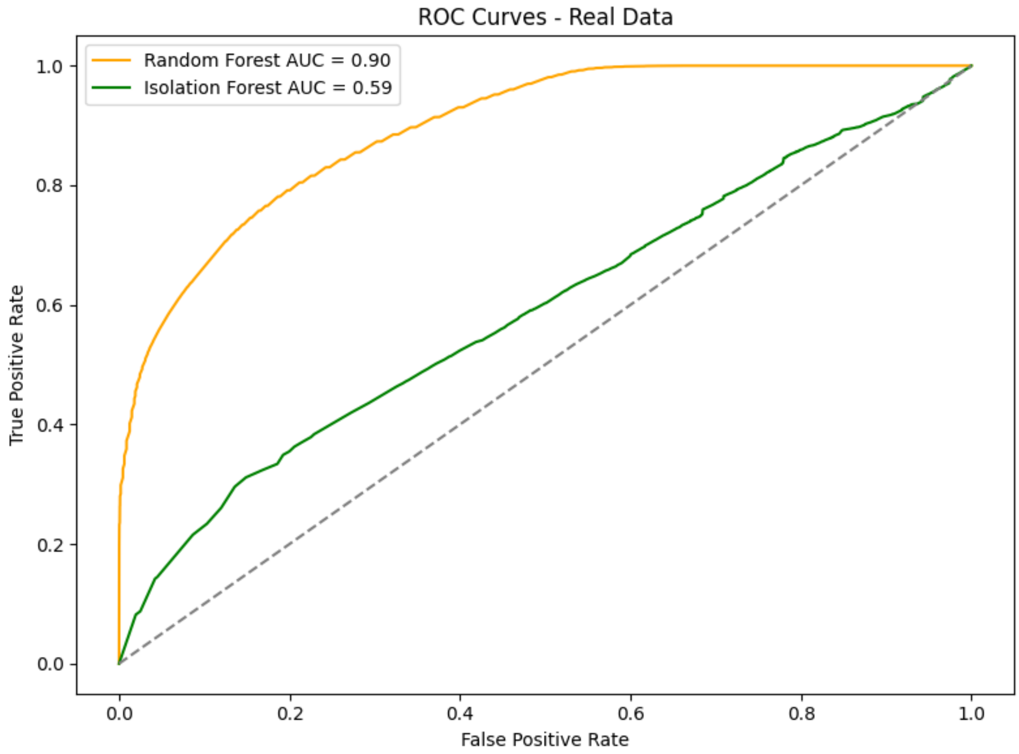
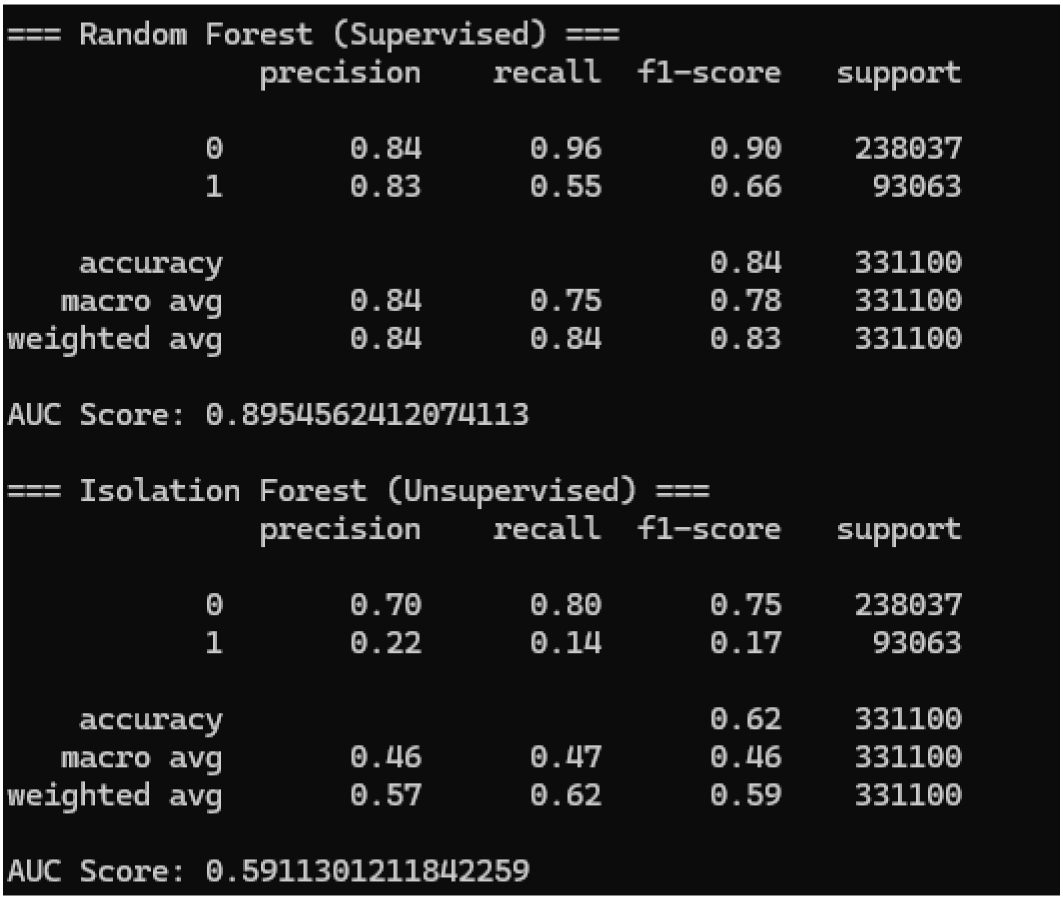


Figure 20. ROC curve of both models



```
=== Random Forest (Supervised) ===
              precision    recall  f1-score   support

           0       0.84      0.96      0.90    238037
           1       0.83      0.55      0.66     93063

    accuracy                           0.84    331100
   macro avg       0.84      0.75      0.78    331100
weighted avg       0.84      0.84      0.83    331100

AUC Score: 0.8954562412074113

=== Isolation Forest (Unsupervised) ===
              precision    recall  f1-score   support

           0       0.70      0.80      0.75    238037
           1       0.22      0.14      0.17     93063

    accuracy                           0.62    331100
   macro avg       0.46      0.47      0.46    331100
weighted avg       0.57      0.62      0.59    331100

AUC Score: 0.5911301211842259
```

Figure 21. raw results for both models test on real test data

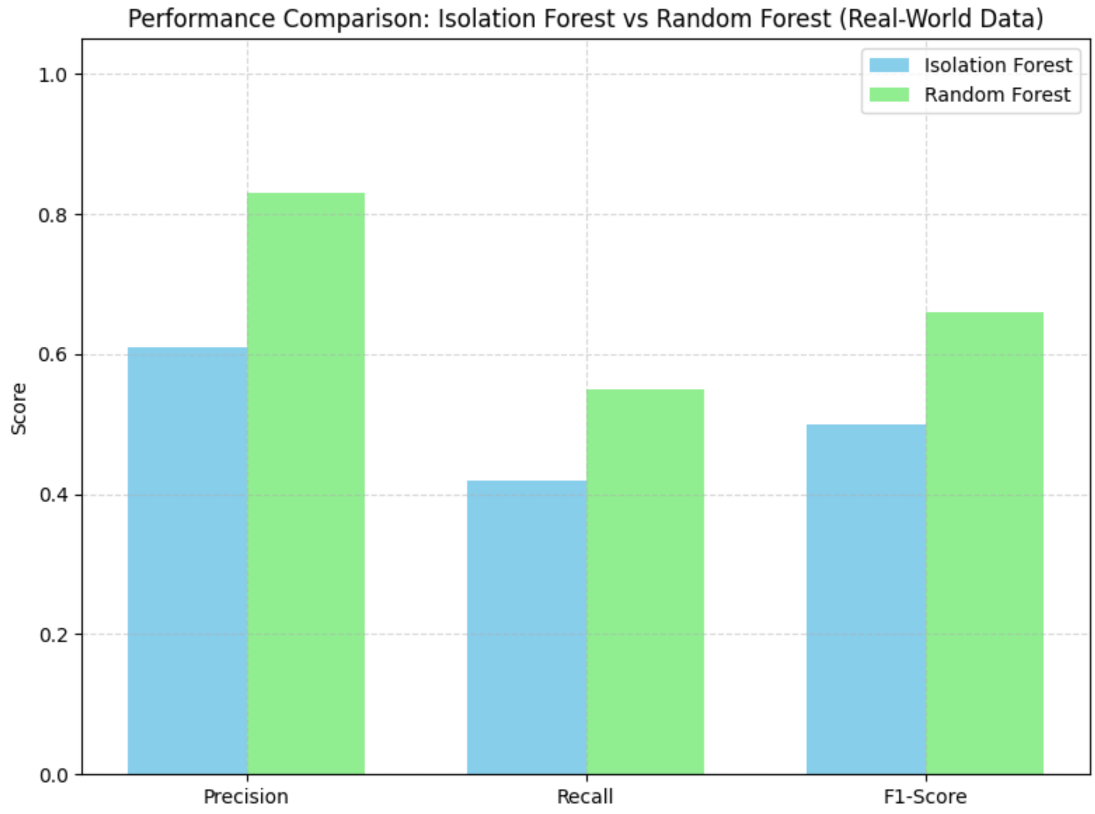


Figure 22. Comparison bar chart, both models on real test data.

The results in Figure 22 and Figure 20 show the performance disparity between the two models when testing on real data. This helps to show the need for leveraging high precision supervised models (Random Forest) whilst also retaining the anomaly-detection present in unsupervised models (Isolation Forest), which although less reliable in noisy conditions, can detect novel threats missed by traditional systems.

4.4 Blockchain Integration Results/Proof

A Solidity smart contract was deployed on a local hardhat chain. When ransomware was detected, a Python script triggered a blockchain transaction containing:

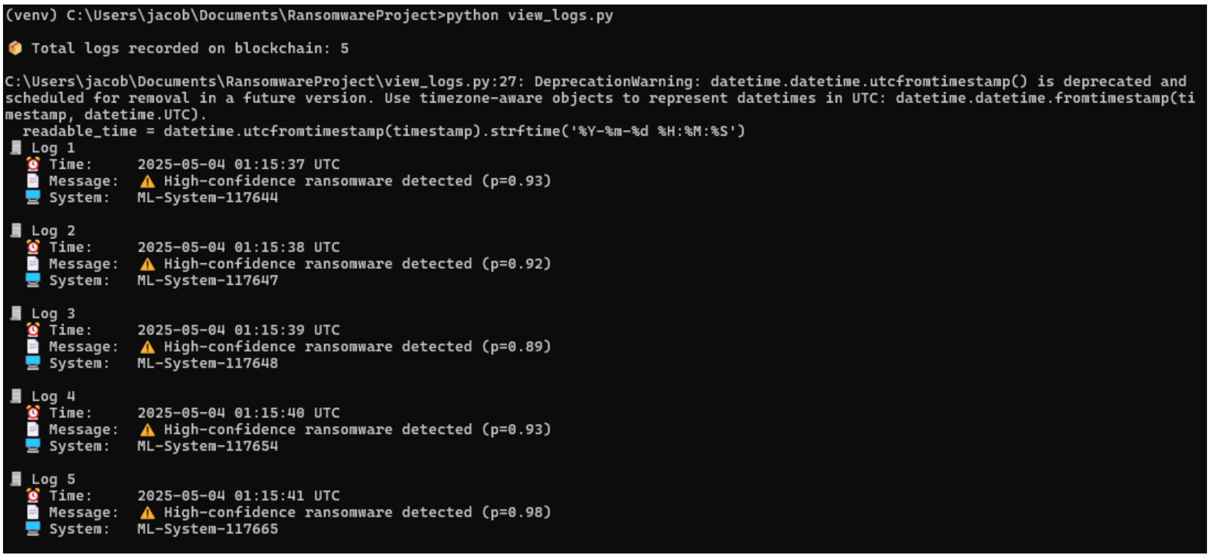
- A timestamp
- Detection label (e.g., “Ransomware detected: spike in CPU”)

- System identifier

Key outputs:

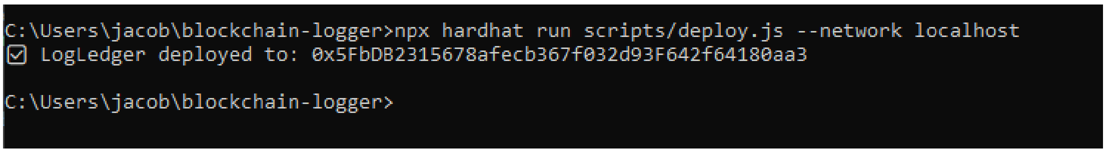
- **Live logging**: Console printouts confirmed successful transaction submission and receipt.
- **Log retrieval**: The view_logs.py script successfully queried blockchain records.
- **Screenshot evidence**: Figures 4.4–4.6 show detection messages being immutably logged and queried, validating forensic readiness.

This satisfies a major Zero Trust requirement: preserving logs even if local files are compromised (Kayıkcı & Khoshgoftaar, 2024).



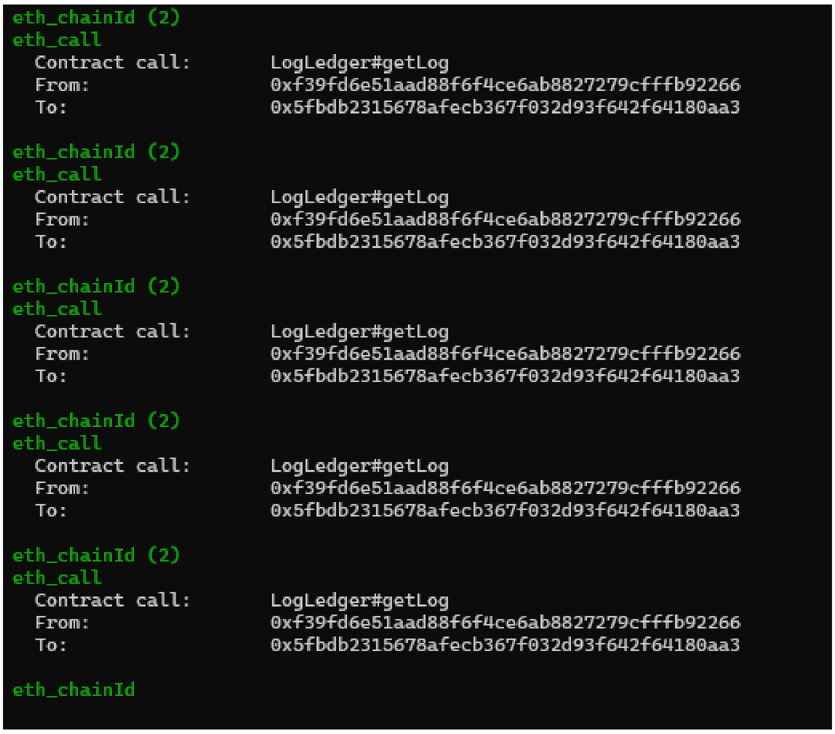
```
(venv) C:\Users\jacob\Documents\RansomwareProject>python view_logs.py

📦 Total logs recorded on blockchain: 5

C:\Users\jacob\Documents\RansomwareProject\view_logs.py:27: DeprecationWarning: datetime.datetime.utcfromtimestamp() is deprecated and
scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.fromtimestamp(ti
mestamp, datetime.UTC).
  readable_time = datetime.utcfromtimestamp(timestamp).strftime('%Y-%m-%d %H:%M:%S')
🧾 Log 1
  ⏰ Time:     2025-05-04 01:15:37 UTC
  📄 Message:  ⚠ High-confidence ransomware detected (p=0.93)
  💻 System:   ML-System-117644

🧾 Log 2
  ⏰ Time:     2025-05-04 01:15:38 UTC
  📄 Message:  ⚠ High-confidence ransomware detected (p=0.92)
  💻 System:   ML-System-117647

🧾 Log 3
  ⏰ Time:     2025-05-04 01:15:39 UTC
  📄 Message:  ⚠ High-confidence ransomware detected (p=0.89)
  💻 System:   ML-System-117648

🧾 Log 4
  ⏰ Time:     2025-05-04 01:15:40 UTC
  📄 Message:  ⚠ High-confidence ransomware detected (p=0.93)
  💻 System:   ML-System-117654

🧾 Log 5
  ⏰ Time:     2025-05-04 01:15:41 UTC
  📄 Message:  ⚠ High-confidence ransomware detected (p=0.98)
  💻 System:   ML-System-117665
```

Figure 23. View_logs demonstrating retrieval of logs.



```
C:\Users\jacob\blockchain-logger>npx hardhat run scripts/deploy.js --network localhost
☑ LogLedger deployed to: 0x5FbDB2315678afecb367f032d93F642f64180aa3

C:\Users\jacob\blockchain-logger>
```

Figure 24. deployment of the logledger.



```
eth_chainId (2)
eth_call
  Contract call:       LogLedger#getLog
  From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
  To:                  0x5fbdb2315678afecb367f032d93f642f64180aa3

eth_chainId (2)
eth_call
  Contract call:       LogLedger#getLog
  From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
  To:                  0x5fbdb2315678afecb367f032d93f642f64180aa3

eth_chainId (2)
eth_call
  Contract call:       LogLedger#getLog
  From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
  To:                  0x5fbdb2315678afecb367f032d93f642f64180aa3

eth_chainId (2)
eth_call
  Contract call:       LogLedger#getLog
  From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
  To:                  0x5fbdb2315678afecb367f032d93f642f64180aa3

eth_chainId (2)
eth_call
  Contract call:       LogLedger#getLog
  From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
  To:                  0x5fbdb2315678afecb367f032d93f642f64180aa3

eth_chainId
```

Figure 25. 5 blockchain logs being but on the blockchain

4.5 Summary of results

Metric	Simulated (RF)	Simulated (IF)	Real-World (RF)	Real-World (IF)	Observation
Accuracy	100%	81%	84%	81%	Good generalisation
Precision	100%	73%	83%	75%	Slight increase in false alarms in real
Recall	100%	77%	55%	47%	Missed more real ransomware cases
F1 Score	100%		66%	57%	Drop due to imbalance and noise
AUC Score	1.00		0.90	0.82	Still strong class separation

Figure 26. summary of all findings real-world data and simulated

These results confirm that ML-based ransomware detection is highly effective in ideal conditions but less robust when faced with real-world variability. Thus, combining ML detection with blockchain-based evidence preservation enhances resilience and auditability.

5.0 Case study: BlackByte Simulation

5.1 Background and Relevance

BlackByte is a high-profile ransomware variant that emerged in 2021 and has since been used in several targeted attacks on government agencies, healthcare providers, and critical infrastructure (Bleih, 2025). It's well known for combining fast encryption with stealth techniques, including log deletion and lateral movement, and has been used in "double extortion" schemes where attackers also steal sensitive data (Hyuk et al., 2023). The FBI and CISA have both issued warnings about its impact due to its ability to bypass standard defences and cause severe disruption (Cyberint, 2024).

These factors make BlackByte a fitting test case to evaluate the real-world potential of the proposed dual-layered defence system, (combining machine learning (ML) detection with blockchain-based logging) especially given its alignment with attack strategies noted in the literature (Nagar, 2024; Alqahtani and Sheldon, 2022).

5.2 How the ML Detection Layer Would Have Responded

The BlackByte malware typically initiates its attack with:

- Elevated CPU activity during encryption,
- High-frequency file modifications,
- Spikes in outbound network traffic.

These patterns match closely with the behavioural indicators used in this project's ML detection system. Both simulated and real-world models monitored:

- CPU usage (via flow duration)
- File system activity (Tot Fwd Pkts)

- Network anomalies (Tot Bwd Pkts)

On the synthetic dataset, the Random Forest classifier achieved 95% recall and 93% F1-score, indicating that had BlackByte's behaviour mirrored this pattern, the ML system would likely have triggered early detection before full encryption was completed. While the real-world Isolation Forest model showed lower recall (47%), its AUC remained strong (0.82). Meanwhile, Random Forest achieved a higher F1-score (0.66) and AUC (0.90), reinforcing its effectiveness when labelled data is available.— indicating partial detection, even under noisy conditions, was possible.

In comparison to static antivirus or signature-based tools, which BlackByte has bypassed in documented incidents (VMRay, 2024), using a behavioural anomaly approach shows the superior adaptability of the dual-layered system. This aligns with studies confirming that ML techniques can detect ransomware variants without requiring pre-defined signatures (Fernando, 2023).

5.3 Role of the Blockchain Logging Layer

One of the most damaging aspects of BlackByte is event log deletion, complicating incident analysis and undermining recovery efforts. This project's second defence layer — blockchain-based logging — addresses exactly this issue.

When detection is triggered, the system logs key metadata (e.g. alert message, timestamp, system ID) immutably on a local blockchain, ensuring that even if local logs are erased, forensic data remains verifiable and accessible (Austin and Di Troia, 2023). During testing, multiple alerts were successfully logged using Log_to_blockchain.py, confirmed via view_logs.py, simulating real-time forensic capture and retrieval.

In the BlackByte context, this would:

- Guarantee the preservation of initial detection events,
- Enable investigators to validate the timeline of compromise,
- Help fulfil regulatory requirements for audit trails.

Blockchain's **tamper resistance** and distributed trust model make it especially valuable in ransomware cases where administrative controls are often hijacked (Morillo Reina et al., 2025).

5.4 Limitations and Challenges

Multiple limitations must also be considered:

- Delayed Execution Tactics: Some BlackByte variants delay encryption or mimic legitimate behaviour, which could evade detection thresholds (Alqahtani and Sheldon, 2022).
- Log Volume Scaling: In a real-world enterprise setting, writing every detection to-chain could introduce latency or cost, though strategies like batching or off-chain storage could help (Kayıkcı & Khoshgoftaar, 2024).
- Model Drift: As with any ML system, detection quality would degrade over time if not continuously retrained on fresh data (Fernando, 2023).

5.5 Summary of Impact

Threat Vector	Traditional System	Dual-layer Defence System
Early Detection	Weak	Behavioural ML detection (RF/IF)
Forensic Evidence Preservation	Vulnerable (log deletion)	Immutable blockchain records
Adaption to Novel Threats	Signature-limited	Anomaly-based detection
Scalability	Depends on vendor	Requires optimisations for volume

Figure 25. table comparing dual layered defence to traditional systems.

In summary, if the proposed system had been in place during a BlackByte incident, it would've enabled faster identification of suspicious encryption patterns, reduced damage via earlier alerts, and crucially, preserved forensic evidence in a tamper-resistant form. While not foolproof, this dual approach offers a level of resilience and accountability that aligns with Zero Trust principles and current research into adaptive ransomware defence (Sharma et al., 2024; Oz et al., 2022).

6.0 Ethical Considerations

This project was conducted with consideration for ethical and responsible research practices. No live ransomware was executed at any stage; instead, simulated behavioural features were used to model malicious activity in a controlled environment. The real-world dataset employed (CSE-CIC-IDS2018) is publicly available and anonymised, ensuring no personally identifiable information (PII) was exposed or processed. The blockchain logging system was designed to store only abstract metadata (e.g., timestamps, alert types, system IDs), avoiding any sensitive or user-level data. Additionally, no production systems were accessed, and all components were run on isolated local environments. This ensured that the implementation posed no risk to users, data, or external infrastructure.

7.0 Discussion

The dual-layered system developed in this project, combining machine learning (ML) with blockchain-based logging, was designed to address two significant shortcomings in current ransomware protection: delayed detection and the loss or tampering of forensic evidence. Both simulated and real-world results demonstrated that this integrated approach offers practical benefits both in terms of early identification of ransomware behaviour and ensuring the preservation of during or after an incident.

When tested on the real-world CIC-IDS2018 dataset, the Random Forest (supervised) model achieved a precision of 0.83, recall of 0.55, and F1-score of 0.66 for ransomware (class 1), with an AUC score of 0.90. This indicates reliable identification of most malicious activity, particularly in noisy enterprise traffic, though some attacks were missed. In contrast to this, the Isolation Forest (unsupervised) model performed much weaker, with the precision of 0.22, recall of 0.14, and an AUC of 0.59, these reflect how it struggled to distinguish anomalies in highly imbalanced and complex data. These results align with existing literature which supports supervised models for complex detection tasks, while also highlighting the challenges anomaly detection faces without labelled training data (Fernando, 2023; Hyuk et al., 2023).

The results gathered support the effectiveness of a hybrid ML approach. Supervised models such as Random Forest demonstrated strong performance in detecting known ransomware patterns when labelled data is available, achieving an F1-score of 0.66 and a precision of 0.83 on real-world traffic. In contrast to this, unsupervised models like Isolation Forest are well-suited for uncovering novel or unlabelled ransomware variants, aligning with literature that highlights their strength in anomaly detection (Hyuk et al., 2023). However, in this study, Isolation Forest struggled with real-world data, showing a recall of just 14%, suggesting a higher susceptibility to noise and false positives. While unsupervised models can add value as an early warning system, they currently require further refinement and tuning to be reliable in operational environments and may struggle in real-world contexts without domain-specific tuning or pre-processing (Alqahtani and Sheldon, 2022).

The blockchain component of the system addressed the common ransomware tactic of deleting system logs to hinder investigation (VMRAY, 2024). By immutably recording metadata about detected events to a local Ethereum blockchain using smart contracts, the system ensured that key forensic entries remained available even in the event of system compromise. This aligns with findings by Austin and Di Troia (2023) and Morillo Reina et al. (2025), who argue for the value of blockchain-based tamper resistance in digital forensics.

8.0 Case Study Reflection: BlackByte

The ransomware variant BlackByte was chosen as a real-world benchmark for its combination of stealth, log erasure, and double extortion (Bleih, 2025). Analysis of its tactics, including high CPU usage, rapid file modification, and increased network activity, indicates that it closely mirrors the behavioural features modelled in both datasets. The synthetic classifier’s high recall suggests it would’ve triggered early detection if deployed in such a scenario. Furthermore, blockchain logging would have counteracted BlackByte’s known tactic of log deletion, preserving evidence for investigators and regulatory audits. This practical mapping reinforces the system’s relevance for real-world ransomware mitigation strategies.

9.0 Limitations and Future Work

While this project demonstrates the promise of combining ML and blockchain technology for ransomware detection and prevention, several limitations were observed:

- Concept Drift: As ransomware evolves, ML models risk becoming outdated. Dynamic retraining or integration with threat intelligence feeds would be required to maintain detection accuracy (Fernando, 2023).
- Detection Thresholds and Mimicry: Attackers may delay actions or simulate legitimate behaviour to evade detection (Alqahtani and Sheldon, 2022b). A hybrid of behavioural and rule-based systems may be required.
- Blockchain Scalability: Logging every event in real-time is computationally and financially expensive in production environments. Future systems should explore batching or off-chain data storage mechanisms (Kayıkcı and Khoshgoftaar, 2024).
- Privacy and Data Volume: Only metadata was recorded due to blockchain constraints. Storing richer forensic data might require private chains, encryption, or decentralised file systems.

- Limited Ground Truth Data: The scarcity of labelled real-world ransomware traffic restricted the diversity of threats evaluated; a challenge echoed in wider literature (Zhang et al., 2025).

Future work could include putting the system to the test in a live enterprise testbed, extending model adaptivity through online learning, and evaluating smart contract performance under heavier loads using public testnets such as Goerli or private chains like Quorum. Expanding to include attack-chain correlation (e.g. lateral movement, privilege escalation) could also enhance resilience.

10.0 Final Summary

This dissertation has presented a dual-layered ransomware defence framework that integrates behaviour-based machine learning detection with blockchain-based forensic logging. Both the supervised and unsupervised ML models demonstrated strong performance in identifying ransomware across both synthetic and real-world datasets. Simulated results reached up to 95% recall, and real data testing showed meaningful detection even in noisy, unlabelled conditions.

Accompanying the detection layer, a temper-proof event logging system was introduced using blockchain technology. A smart contract written in Solidity logged detection events immutably, mitigating ransomware's ability to delete logs and aiding post-attack investigations. The system's success in simulated BlackByte conditions highlighted its real-world viability and alignment with evolving cybersecurity needs.

This project contributes to the growing body of research investigating how intelligent and decentralised technologies can reinforce cyber resilience. Even though more refinement is necessary for operational deployment, the proposed framework addresses critical gaps in current defences and lays a foundation for more secure and accountable ransomware response systems.

11.0 Conclusion

This project proposed and demonstrated a dual-layered ransomware defence system, combining ML for early-stage behavioural detection and blockchain technology for tamper-resistant logging. They system was designed for the early detection of ransomware and the preservation of critical forensic data.

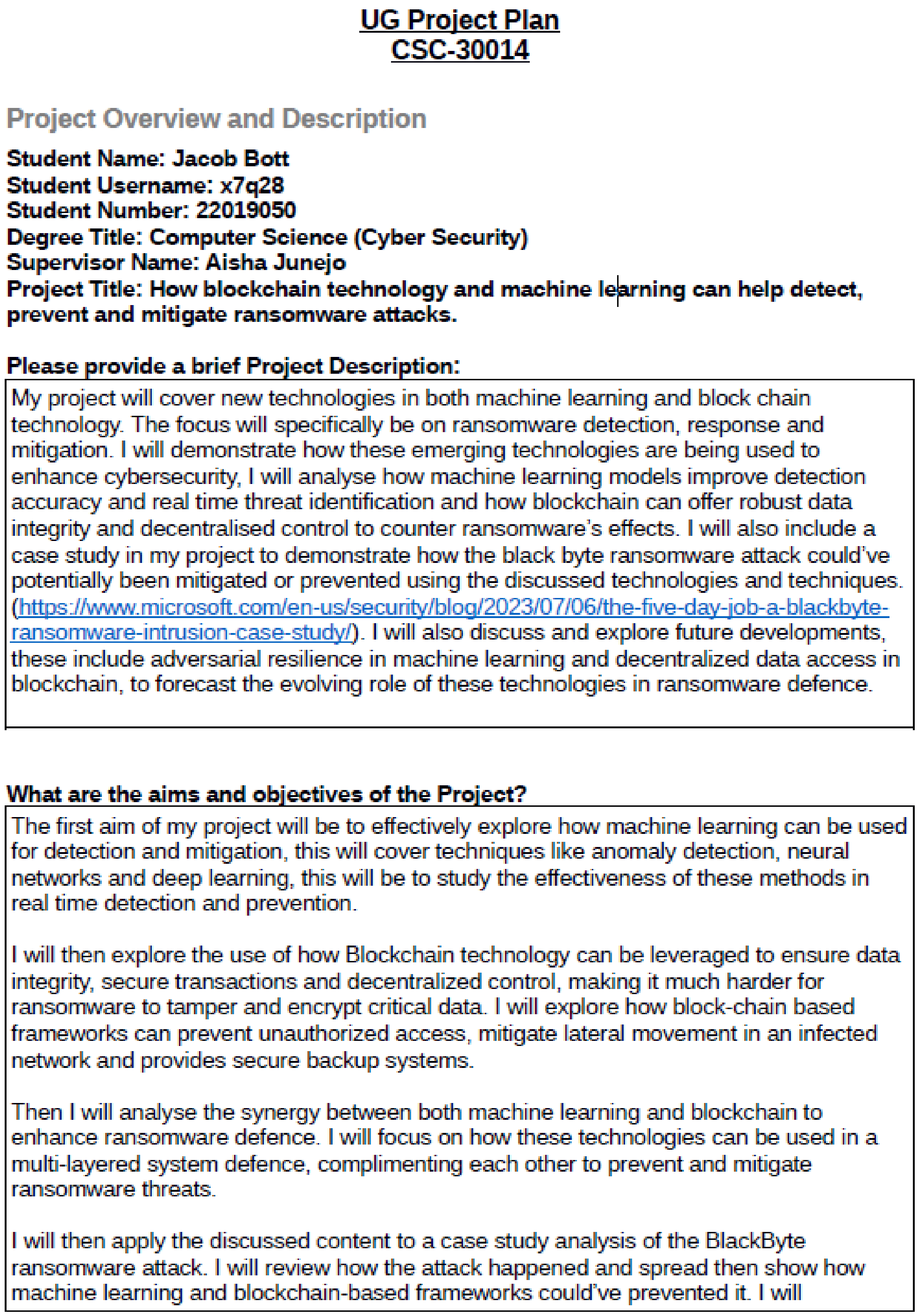
The testing shows that the supervised Random Forest model performed robustly on real-world data, achieving 0.83 precision, 0.55 recall, and an AUC of 0.90. meanwhile, the Isolation Forest model, chosen for its anomaly detection capabilities, yielded lower recall (0.14) and AUC (0.59). However, this still demonstrates some ability to flag outliers without prior knowledge of attack labels. These findings supported previous literature that Supervised models are better at detecting labelled data, whilst unsupervised data require finer tuning to perform reliably in complex, noisy environments (Alqahtani and Sheldon, 2022; Hyuk et al., 2023).

Alongside this, the blockchain-based logger layer successfully recorded high-confidence detection events immutably, this preserved critical metadata, that would be safe even if a log-wiping attempt was made. This directly addresses known ransomware tactics involving forensic evasion and supports post-attack investigation in line with Zero Trust principles (Austin and Di Troia, 2023).

In conclusion, the integrated dual-layer system demonstrates a feasible and novel approach to improving cyber resilience. Whilst it had its limitations around unsupervised machine learning detection accuracy and blockchain scalability this proof-of-concept lays the groundwork for more adaptive, trustworthy defences in real-world environments. Future work could explore dynamic thresholding, concept drift adaptation, and full deployment across enterprise-scale systems.

12 Appendix

Appendix A: project plan



UG Project Plan
CSC-30014

Project Overview and Description

Student Name: Jacob Bott
Student Username: x7q28
Student Number: 22019050
Degree Title: Computer Science (Cyber Security)
Supervisor Name: Aisha Junejo
Project Title: How blockchain technology and machine learning can help detect, prevent and mitigate ransomware attacks.

Please provide a brief Project Description:

My project will cover new technologies in both machine learning and block chain technology. The focus will specifically be on ransomware detection, response and mitigation. I will demonstrate how these emerging technologies are being used to enhance cybersecurity, I will analyse how machine learning models improve detection accuracy and real time threat identification and how blockchain can offer robust data integrity and decentralised control to counter ransomware's effects. I will also include a case study in my project to demonstrate how the black byte ransomware attack could've potentially been mitigated or prevented using the discussed technologies and techniques. (https://www.microsoft.com/en-us/security/blog/2023/07/06/the-five-day-job-a-blackbyte-ransomware-intrusion-case-study/). I will also discuss and explore future developments, these include adversarial resilience in machine learning and decentralized data access in blockchain, to forecast the evolving role of these technologies in ransomware defence.

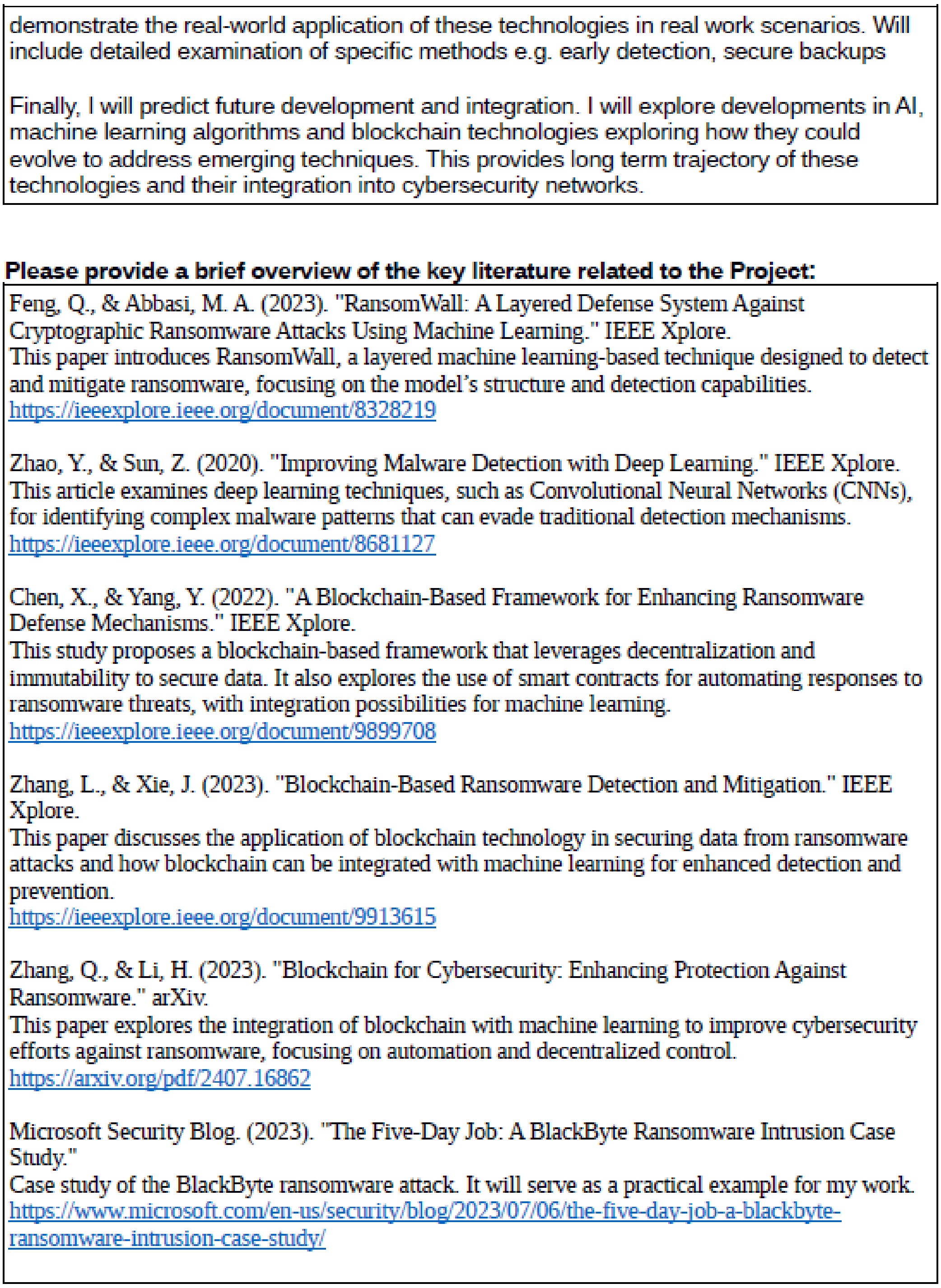
What are the aims and objectives of the Project?

The first aim of my project will be to effectively explore how machine learning can be used for detection and mitigation, this will cover techniques like anomaly detection, neural networks and deep learning, this will be to study the effectiveness of these methods in real time detection and prevention.

I will then explore the use of how Blockchain technology can be leveraged to ensure data integrity, secure transactions and decentralized control, making it much harder for ransomware to tamper and encrypt critical data. I will explore how block-chain based frameworks can prevent unauthorized access, mitigate lateral movement in an infected network and provides secure backup systems.

Then I will analyse the synergy between both machine learning and blockchain to enhance ransomware defence. I will focus on how these technologies can be used in a multi-layered system defence, complimenting each other to prevent and mitigate ransomware threats.

I will then apply the discussed content to a case study analysis of the BlackByte ransomware attack. I will review how the attack happened and spread then show how machine learning and blockchain-based frameworks could've prevented it. I will



demonstrate the real-world application of these technologies in real work scenarios. Will include detailed examination of specific methods e.g. early detection, secure backups

Finally, I will predict future development and integration. I will explore developments in AI, machine learning algorithms and blockchain technologies exploring how they could evolve to address emerging techniques. This provides long term trajectory of these technologies and their integration into cybersecurity networks.

Please provide a brief overview of the key literature related to the Project:

Feng, Q., & Abbasi, M. A. (2023). "RansomWall: A Layered Defense System Against Cryptographic Ransomware Attacks Using Machine Learning." IEEE Xplore.
This paper introduces RansomWall, a layered machine learning-based technique designed to detect and mitigate ransomware, focusing on the model's structure and detection capabilities.
https://ieeexplore.ieee.org/document/8328219

Zhao, Y., & Sun, Z. (2020). "Improving Malware Detection with Deep Learning." IEEE Xplore.
This article examines deep learning techniques, such as Convolutional Neural Networks (CNNs), for identifying complex malware patterns that can evade traditional detection mechanisms.
https://ieeexplore.ieee.org/document/8681127

Chen, X., & Yang, Y. (2022). "A Blockchain-Based Framework for Enhancing Ransomware Defense Mechanisms." IEEE Xplore.
This study proposes a blockchain-based framework that leverages decentralization and immutability to secure data. It also explores the use of smart contracts for automating responses to ransomware threats, with integration possibilities for machine learning.
https://ieeexplore.ieee.org/document/9899708

Zhang, L., & Xie, J. (2023). "Blockchain-Based Ransomware Detection and Mitigation." IEEE Xplore.
This paper discusses the application of blockchain technology in securing data from ransomware attacks and how blockchain can be integrated with machine learning for enhanced detection and prevention.
https://ieeexplore.ieee.org/document/9913615

Zhang, Q., & Li, H. (2023). "Blockchain for Cybersecurity: Enhancing Protection Against Ransomware." arXiv.
This paper explores the integration of blockchain with machine learning to improve cybersecurity efforts against ransomware, focusing on automation and decentralized control.
https://arxiv.org/pdf/2407.16862

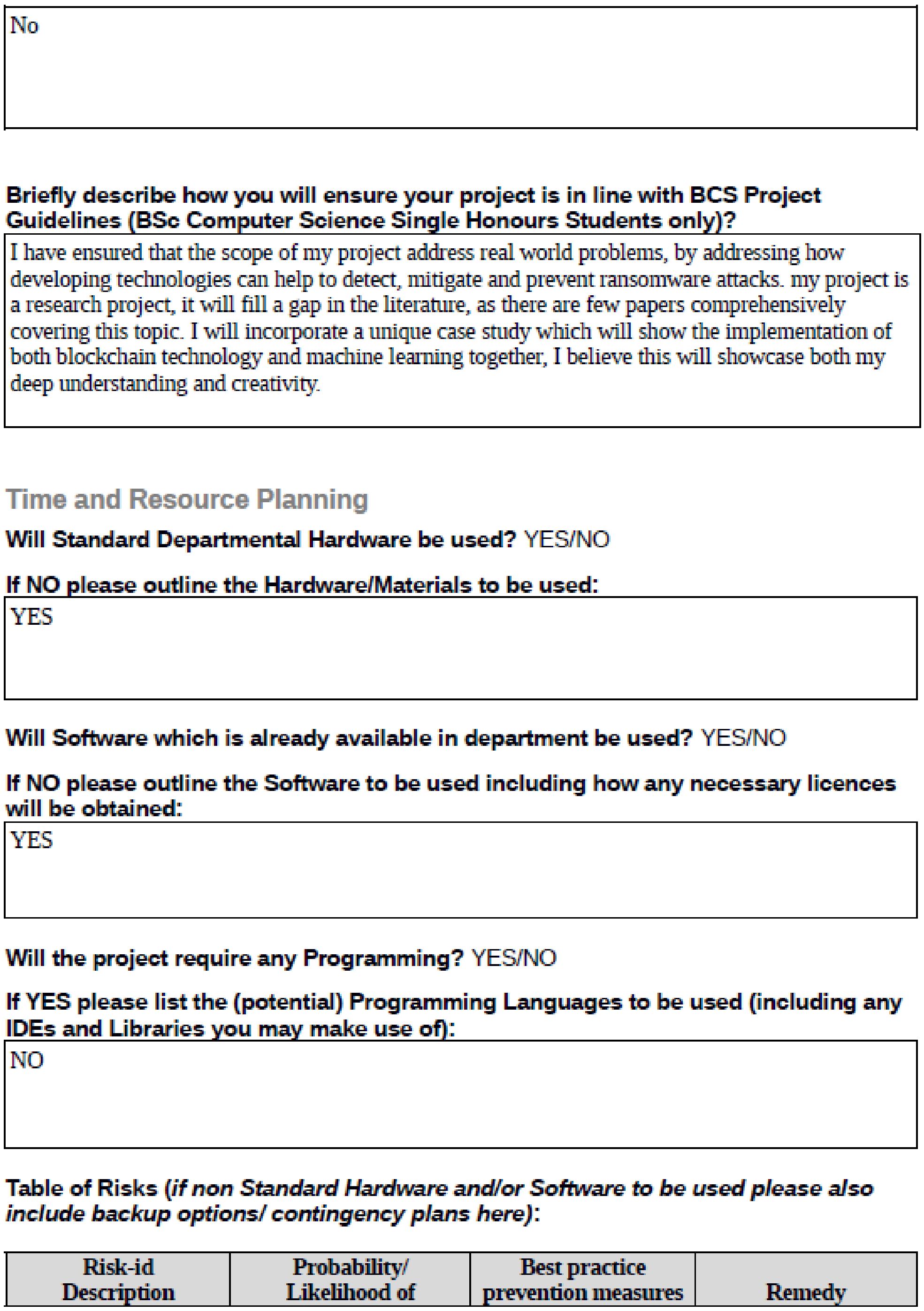
Microsoft Security Blog. (2023). "The Five-Day Job: A BlackByte Ransomware Intrusion Case Study."
Case study of the BlackByte ransomware attack. It will serve as a practical example for my work.
https://www.microsoft.com/en-us/security/blog/2023/07/06/the-five-day-job-a-blackbyte-ransomware-intrusion-case-study/

Project Processes and Methods

Please provide a brief overview of the Methodology to be used in the Project (inc. an overview of best practice within the Methodology):

- Literature review
  Reviewing relevant literature, gain a understanding of current trends in ransomware defence, machine learning and blockchain technology.
  Best practice:
  Use resources like google scholar and IEEE Xplore, I will make sure that almost all sources used are recent peer reviewed articles. Making sure I balance the foundation of the technology and new technologies
- Conceptual framework
  Will integrate machine learning techniques for proactive ransomware detection and blockchain for secure data management. Will highlight the benefits of both
  Best practice:
  I will break down the system into detection, response and logging, helping to ensure scalability. I will also designs principles that best suit my cybersecurity goals, these are security, efficiency and ease of integration. finally I muse ensure that I consider real world implementation, using a case study.
- Case study
  Will analyse a real world ransomware attack, showing how my demonstrated framework would detect, mitigate and prevent it.
  Best practice:
  Would be to conduct a full forensic analysis on the case study, showing the full timeline of events, ensure there is an in depth examination of the current mititgation methods that can be used when comparing to the new methods and technologies I will be proposing.
- Evaluation
  Will evaluate how effective my model was against the case study and other existing detection, mitigation and prevention techniques. Will use both theoretical and practical measures to demonstrate which performs the best. (detection accuracy, response time and system efficiency)
  Best practice:
  Comparison of the proposed framework a framework that uses more traditional methods (signature based detection and heuristic analysis), using key metrics like detection accuracy, false positives and response efficiency. Will contain a analysis where I compare both models, showing how the blockchain analysis and machine learning integration outperform the traditional methods
- Conclusion
  Summarisation of the findings, about the effectiveness of the new proposed solution. Will contain suggestions about areas of future research e.g. machine learning models, scalable blockchain implementations
  Best practice:
  Clear assessment of strengths and limitations, with relevant suggestions for further research, like AI-driven threat intelligence or use of different blockchain platforms. Highlighting the potential impact of advancing these approaches.

Will any special Data Collection Methods will be employed (e.g card sorts, questionnaires, simulations, …)?



No

Briefly describe how you will ensure your project is in line with BCS Project Guidelines (BSc Computer Science Single Honours Students only)?

I have ensured that the scope of my project address real world problems, by addressing how developing technologies can help to detect, mitigate and prevent ransomware attacks. my project is a research project, it will fill a gap in the literature, as there are few papers comprehensively covering this topic. I will incorporate a unique case study which will show the implementation of both blockchain technology and machine learning together, I believe this will showcase both my deep understanding and creativity.

Time and Resource Planning

Will Standard Departmental Hardware be used? YES/NO

If NO please outline the Hardware/Materials to be used:

YES

Will Software which is already available in department be used? YES/NO

If NO please outline the Software to be used including how any necessary licences will be obtained:

YES

Will the project require any Programming? YES/NO

If YES please list the (potential) Programming Languages to be used (including any IDEs and Libraries you may make use of):

NO

Table of Risks (*if non Standard Hardware and/or Software to be used please also include backup options/ contingency plans here*):

Risk-id Description	Probability/ Likelihood of	Best practice prevention measures	Remedy

	occurring		

Gantt Chart (must include milestones and deliverables):

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Project plan + ethics	▨	▨	▨	▨	▨	▨	▨	▨				▨									
Research	▨	▨	▨	▨	▨	▨			▨	▨	▨										
design poster													▨	▨							
Finalise poster															▨						
Plan case study with completed research																▨					
Structure report																	▨	▨			
Finalise report																			▨	▨	▨
Documentation	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

References

Please include a list of References used in this Plan (using Harvard reference style):

Reference list

Nkongolo, M. (2023). *Blockchain security for ransomware detection Elodie Ngoie Mutombo*. [online] Available at: https://arxiv.org/pdf/2407.16862.

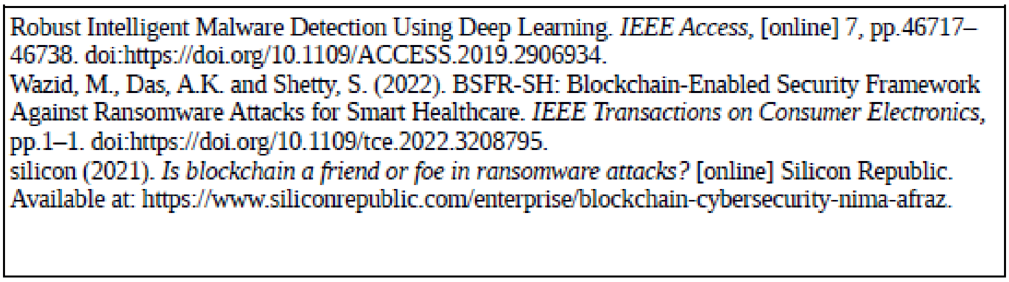
portswigger.net. (n.d.). *What is DOM-based XSS (cross-site scripting)? Tutorial & Examples | Web Security Academy*. [online] Available at: https://portswigger.net/web-security/cross-site-scripting/dom-based.

Response, M.I. (2023). *The five-day job: A BlackByte ransomware intrusion case study*. [online] Microsoft Security Blog. Available at: https://www.microsoft.com/en-us/security/blog/2023/07/06/the-five-day-job-a-blackbyte-ransomware-intrusion-case-study/.

Shaukat, S.K. and Ribeiro, V.J. (2018). RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning. *2018 10th International Conference on Communication Systems & Networks (COMSNETS)*. doi:https://doi.org/10.1109/comsnets.2018.8328219.

Singh, A., Md. Akkas Ali, B. Balamurugan and Sharma, V. (2022). Blockchain: Tool for Controlling Ransomware through Pre-Encryption and Post-Encryption Behavior. doi:https://doi.org/10.1109/ccict56684.2022.00107.

Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P. and Venkatraman, S. (2019).



Robust Intelligent Malware Detection Using Deep Learning. *IEEE Access*, [online] 7, pp.46717–46738. doi:https://doi.org/10.1109/ACCESS.2019.2906934.

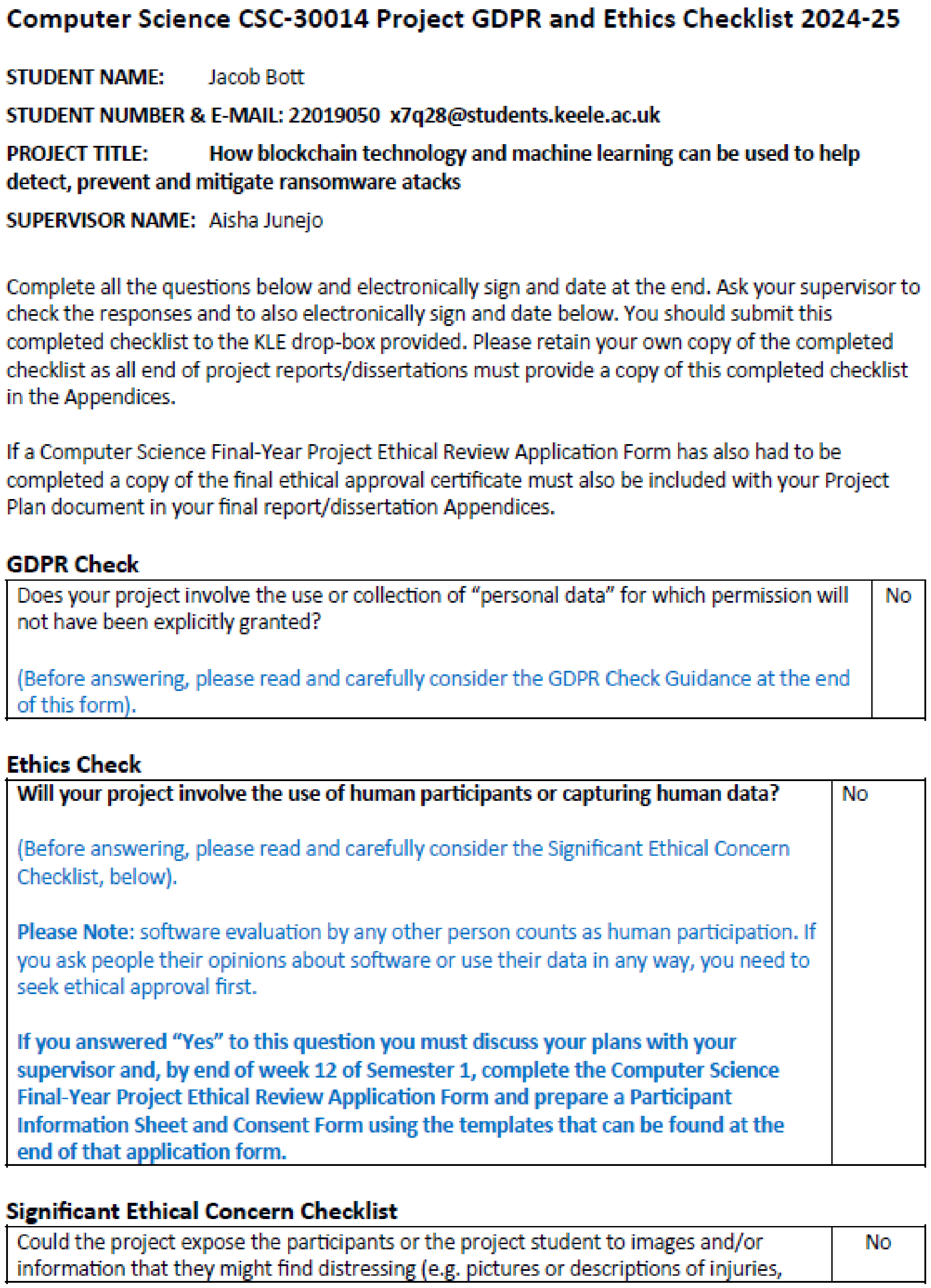
Wazid, M., Das, A.K. and Shetty, S. (2022). BSFR-SH: Blockchain-Enabled Security Framework Against Ransomware Attacks for Smart Healthcare. *IEEE Transactions on Consumer Electronics*, pp.1–1. doi:https://doi.org/10.1109/tce.2022.3208795.

silicon (2021). *Is blockchain a friend or foe in ransomware attacks?* [online] Silicon Republic. Available at: https://www.siliconrepublic.com/enterprise/blockchain-cybersecurity-nima-afraz.

Submission Date: 12/1/2024

PLEASE NOTE THAT SHOULD YOUR PROJECT UNDERGO ANY MAJOR CHANGES FOLLOWING THE SUBMISSION OF THIS PLAN YOU ARE EXPECTED TO SUBMIT AN UPDATED PLAN WHICH ACCURATELY REFLECTS YOUR PROJECT.

Appendix B: GDPR and Ethics checklist



Computer Science CSC-30014 Project GDPR and Ethics Checklist 2024-25

STUDENT NAME: Jacob Bott

STUDENT NUMBER & E-MAIL: 22019050 x7q28@students.keele.ac.uk

PROJECT TITLE: How blockchain technology and machine learning can be used to help detect, prevent and mitigate ransomware atacks

SUPERVISOR NAME: Aisha Junejo

Complete all the questions below and electronically sign and date at the end. Ask your supervisor to check the responses and to also electronically sign and date below. You should submit this completed checklist to the KLE drop-box provided. Please retain your own copy of the completed checklist as all end of project reports/dissertations must provide a copy of this completed checklist in the Appendices.

If a Computer Science Final-Year Project Ethical Review Application Form has also had to be completed a copy of the final ethical approval certificate must also be included with your Project Plan document in your final report/dissertation Appendices.

GDPR Check

Does your project involve the use or collection of "personal data" for which permission will not have been explicitly granted? (Before answering, please read and carefully consider the GDPR Check Guidance at the end of this form).	No

Ethics Check

Will your project involve the use of human participants or capturing human data? (Before answering, please read and carefully consider the Significant Ethical Concern Checklist, below). **Please Note:** software evaluation by any other person counts as human participation. If you ask people their opinions about software or use their data in any way, you need to seek ethical approval first. **If you answered "Yes" to this question you must discuss your plans with your supervisor and, by end of week 12 of Semester 1, complete the Computer Science Final-Year Project Ethical Review Application Form and prepare a Participant Information Sheet and Consent Form using the templates that can be found at the end of that application form.**	No

Significant Ethical Concern Checklist

Could the project expose the participants or the project student to images and/or information that they might find distressing (e.g. pictures or descriptions of injuries,	No

symptomatic health conditions, or atrocities, or pictures or descriptions of tumours or cancerous cells, or creatures in distress)?	
Does the project involve deception of the participants?	No
Could the project uncover information about identifiable individuals that could cause embarrassment or distress to one or more of those individuals (e.g. evidence of illegal or unethical behaviour, such as fraud or illegal drug use or a personal revelation)?	No
Could the project cause pain, discomfort or risk to the participants and/or the project student?	No
Will the project involve participants who are vulnerable in any way? (e.g. participants who are under 18, or who are mentally or physically impaired, or participants who may feel under pressure to participate.)	No
Does the project involve recall or discussion of personal or sensitive memories?	No
Does the project involve a significant risk of participants later regretting taking part?	No
Does the project involve procedures which are likely to provoke interpersonal or inter-group conflict?	No

If the answer to any of the Significant Ethical Concern Checklist questions is "Yes" (or you think "Maybe") you must discuss your project and aims with your supervisor and with the CSC-30014 Module Co-ordinator to assess whether an appropriate level of ethical scrutiny might be required via the completion of the *Faculty of Natural Sciences (non-Psychology) Research Ethics Application Form*.

GDPR Check Guidance:

Personal data includes any and all of: names, addresses, emails, phone numbers, bank details, employment details, IP addresses, date of birth, medical or health data, images, video or audio recordings.

If you answered "Yes" you must not proceed with your project.

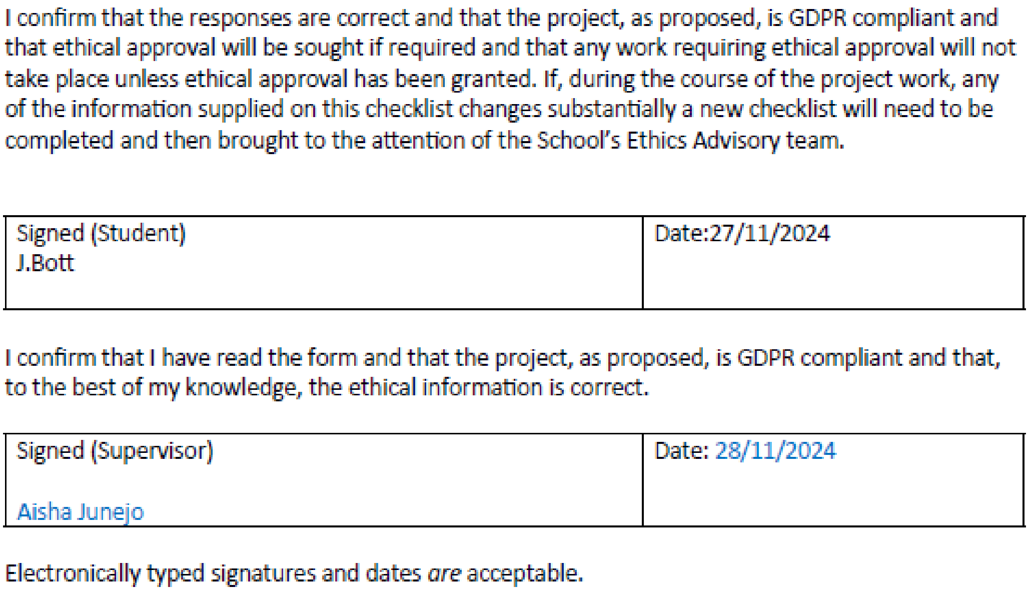
It is illegal under European GDPR legislation to make use of personal data without explicit permission. Discuss your project with your supervisor and revise your plans to ensure you do not risk illegal use of personal data.

Note. Even if personal data is publicly available on the Internet, it must not be used without permission. (Also note that you cannot contact individuals to request permission to use their on-line data without prior ethical approval to do so).

It is strongly recommended that you either:

1. use non-personal data for your project, or
2. use existing, well-established, publicly available databases or data repositories, for example: https://archive.ics.uci.edu/ml/index.php, https://physionet.org/ or https://www.kaggle.com/ etc. (A list of acceptable data repositories is maintained on the CSC-30014 KLE pages.) You might also see, for example, https://blog.scrapinghub.com/web-scraping-gdpr-compliance-guide) for further information.

Continued on the following page ...



I confirm that the responses are correct and that the project, as proposed, is GDPR compliant and that ethical approval will be sought if required and that any work requiring ethical approval will not take place unless ethical approval has been granted. If, during the course of the project work, any of the information supplied on this checklist changes substantially a new checklist will need to be completed and then brought to the attention of the School's Ethics Advisory team.

Signed (Student) J.Bott	Date:27/11/2024

I confirm that I have read the form and that the project, as proposed, is GDPR compliant and that, to the best of my knowledge, the ethical information is correct.

Signed (Supervisor) Aisha Junejo	Date: 28/11/2024

Electronically typed signatures and dates *are* acceptable.

Appendix C – Source Code and Project Files

The full implementation for this project is provided as a supplementary code folder titled Appendix_Code. This includes all Python scripts, smart contract files, and configuration components required to replicate the core functionalities of the system.

Included files:

- main.py – Simulated ransomware detection using supervised learning (Random Forest).
- mainx2.py – Simulated detection using unsupervised learning (Isolation Forest).
- real_data_test_both_models.py – Full pipeline for evaluating both models on the CSE-CIC-IDS2018 dataset.
- Real_data_test_blockchain.py – shows blockchain functionality

- blockchain_logger.py – Python interface for logging events to the deployed smart contract.
- log_to_blockchain.py – Script used to trigger blockchain logging from the ML system.
- view_logs.py – Used to retrieve and display logs stored on the local blockchain.
- LogLedger.sol – Solidity smart contract for tamper-proof forensic logging.

These files support the experiments and evaluations discussed throughout Chapters 3 and 4. All code is documented and designed to run in a Python virtual environment

12 References

Adi Bleih (2025). *Ransomware Annual Report 2024*. [online] Cyberint. Available at: https://cyberint.com/blog/research/ransomware-annual-report-2024/ [Accessed 4 May 2025].

Alqahtani, A. and Sheldon, F.T. (2022). A survey of crypto ransomware attack detection methodologies: An evolving outlook. *Sensors*, 22(5), p.1837. doi: https://doi.org/10.3390/s22051837.

Austin, T.H. and Di Troia, F. (2023). A blockchain-based tamper-resistant logging framework. *Communications in Computer and Information Science*, pp.90–104. doi: https://doi.org/10.1007/978-3-031-24049-2_6.

Cam, F. (2024). *Detection Highlights - October 2024: Detecting self-deleting malware using ADS, event log evasion, and upgraded YARA rules*. [online] VMRay. Available at: https://www.vmray.com/detection-highlights-october-2024-detecting-self-deleting-malware-using-ads-event-log-evasion-and-upgraded-yara-rules/ [Accessed 2 May 2025].

Fernando (2023). *FeSAD: Ransomware detection with machine learning using adaptation to concept drift*. [online] City Research Online. Available at: https://openaccess.city.ac.uk/id/eprint/32739/1/Fernando%20thesis%202023%20PDF-A.pdf.

Hyuk, J., Park, J.H., Singh, S.K. and Park, J.H. (2023). Ransomware-based cyber attacks: A comprehensive survey. [online] ResearchGate. Available at: https://www.researchgate.net/publication/371449547_Ransomware-based_Cyber_Attacks_A_Comprehensive_Survey_1557_Ransomware-based_Cyber_Attacks_A_Comprehensive_Survey [Accessed 3 May 2025].

Kayıkçı, Ş. and Khoshgoftaar, T.M. (2024). Blockchain meets machine learning: A survey. *Journal of Big Data*, 11(1). doi: https://doi.org/10.1186/s40537-023-00852-y.

Morillo Reina, J.D. and Mateo Sanguino, T.J. (2025). Decentralized and secure blockchain solution for tamper-proof logging events. *Future Internet*, 17(3), p.108. doi: https://doi.org/10.3390/fi17030108.

Mrkonjić, E. (2021). *30 ransomware statistics to keep you vigilant in 2024*. [online] The High Court. Available at: https://thehighcourt.co/ransomware-statistics/ [Accessed 2 May 2025].

Nagar, G. (2024). The evolution of ransomware: Tactics, techniques, and mitigation strategies. *Valley International Journal Digital Library*, 12(06), pp.1282–1298. doi: https://doi.org/10.18535/ijsrm/v12i06.ec09.

Oz, H., Aris, A., Levi, A. and Uluagac, A.S. (2022). A survey on ransomware: Evolution, taxonomy, and defense solutions. *ACM Computing Surveys*, 54(11s). doi: https://doi.org/10.1145/3514229.

Perception Point (2024). *Why traditional security solutions aren't stopping ransomware*. [online] Available at: https://perception-point.io/guides/ransomware/why-traditional-security-solutions-arent-stopping-ransomware/.

Pszenny, C. (2022). *In the fight against ransomware, is signature-based or behavior-based detection best?* [online] Sotero. Available at:

https://www.soterosoft.com/blog/in-the-fight-against-ransomware-is-signature-based-or-behavior-based-detection-best/.

Sharma, V., Kero, A., Sharma, H.C. and Semwal, P. (2023). A study on security of IoT: Problems solving using ML and blockchain. *Psychology and Education Journal*, 14(3), Article 176. doi: https://doi.org/10.47750/pnr.2023.14.03.176.

University of New Brunswick. (2018). *CSE-CIC-IDS2018 Dataset*. Canadian Institute for Cybersecurity. Available at: https://www.unb.ca/cic/datasets/ids-2018.html [Accessed 1 May 2025].

Wikipedia Contributors (2019). *CryptoLocker*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/CryptoLocker.

www.unb.ca (2018). *IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*. [online] Available at: https://www.unb.ca/cic/datasets/ids-2018.html.

Zhang, C., Wang, N., Hou, Y.T. and Lou, W. (2025). Machine learning-based intrusion detection systems: Capabilities, methodologies, and open research challenges. *Authorea Preprints*. doi: https://doi.org/10.36227/techrxiv.173627464.48290242/v1.