**IOT BLOCKCHAIN SECURITY ANOMALY DETECTION SYSTEM: A COMPARATIVE MACHINE LEARNING APPROACH**

**BY**

**STUDENT NAME**

**BHU/22/00/00/0000**

**B.Sc. CYBER SECURITY**

**JUNE 2026**

**IOT BLOCKCHAIN SECURITY ANOMALY DETECTION SYSTEM: A COMPARATIVE MACHINE LEARNING APPROACH**

**BY**

**STUDENT NAME**

**BHU/22/00/00/0000**

**A PROJECT SUBMITTED TO THE DEPARTMENT OF CYBER SECURITY, FACULTY OF COMPUTING, BINGHAM UNIVERSITY, KARU, NASARAWA STATE.**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF BACHELOR OF SCIENCE DEGREE (B.Sc.) CYBER SECURITY.**

# **DECLARATION**

I hereby declare that the project work entitled **“**IOT BLOCKCHAIN SECURITY ANOMALY DETECTION SYSTEM: A COMPARATIVE MACHINE LEARNING APPROACH.” submitted to the Department of Cybersecurity, Faculty of Computing, Bingham University Karu, Nasarawa State. It is a record of an original work done by me under the supervision of Mr Musa Yusuf. This project is submitted in partial fulfillment of the requirement for the award of a Bachelor of Science degree in Cybersecurity. The results embodied in this project have not been submitted to any other University or Institute for the award of any degree or diploma. All Sources of information have been duly attributed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**STUDENT NAME DATE**

**BHU/22/00/00/0000**

# **CERTIFICATION**

This is to certify that this Project Report titled **“IOT BLOCKCHAIN SECURITY ANOMALY DETECTION SYSTEM: A COMPARATIVE MACHINE LEARNING APPROACH.”** was carried out by **Student Name** with the Matriculation Number BHU/22/00/00/0000in fulfillment for the award of Bachelor of Science Degree in Cyber Security, Bingham University, Nasarawa State, Nigeria.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Mr. Musa Yusuf** **DATE**

**SUPERVISOR**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Dr. Oluwasegun I. Adelaiye** **DATE**

**HEAD OF DEPARTMENT**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**EXTERNAL EXAMINER DATE**

# **ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to God Almighty for the gift of life. I wish to express my sincere appreciation to my project supervisor Mr Musa Yusuffor the guidance, inspiration and continuous support during the project period.

I would like to thank the Dean of the Faculty of Computing Dr. Adamu S. Usman. My profound gratitude goes to the Departmental SIWES Coordinator Dr. Yakubu Aliyu Ibrahim and all the lecturers in the Department of Computer Science which include Mrs. Oluwatoyin Adelakun Adeyemo, Mr. Maikori Ezekiel Jenom and Mr. Barka T. Fori. A special thanks to all the lecturers in Faculty of Computing which include: Dr. Anne Hassan Bijik, Dr. Onu Egena. Dr. Oluwasegun Ishaya Adelaiye, Dr. Victor Kulugh, Mr. Musa Yusuf, Mr. Ibrahim Lawal and Mrs. Ipole Nancy. I also thank the Technologists in the Faculty which include: Mr. Joseph Oladele Aremu, Mr. Ngale Lautong and Mr. Shadrach Akoh. I also appreciate the Departmental Secretary Mrs. Talatu Bowman. A big thank you to all the teaching and non-teaching staff of the Faculty of Computing, Bingham University.

I owe a debt of gratitude to my loving parents, Mr. and Mrs. SURNAME, for their unwavering support in all aspects. I also thank my siblings for their constant presence and encouragement.

Lastly, I appreciate my fellow students in the Department of Cyber Security for their valuable contributions, collaboration, and support throughout this project.

# **DEDICATION**

Firstly, this project is dedicated to God, for providing me adequate strength, guidance and wisdom throughout this project, and also; to my parents Mr and Mrs …..

**TABLE OF CONTENTS**

[DECLARATION i](#_Toc216555749)

[CERTIFICATION ii](#_Toc216555750)

[ACKNOWLEDGEMENT iii](#_Toc216555751)

[DEDICATION iv](#_Toc216555752)

[CHAPTER ONE 1](#_Toc216555753)

[INTRODUCTION 1](#_Toc216555754)

[1.1 Background of Study 1](#_Toc216555755)

[1.1.1 IoT Security Challenges 2](#_Toc216555756)

[1.1.2 Blockchain Integration in IoT 2](#_Toc216555757)

[1.1.3 Machine Learning for Anomaly Detection 3](#_Toc216555758)

[1.2 Statement of Problem 3](#_Toc216555759)

[1.3 Aim and Objectives 4](#_Toc216555760)

[1.4 Significance of Study 5](#_Toc216555761)

[1.5 Scope of Study 5](#_Toc216555762)

[1.6 Organization of the Study 5](#_Toc216555763)

[CHAPTER TWO 7](#_Toc216555764)

[LITERATURE REVIEW 7](#_Toc216555765)

[2.1 Introduction to the Project Topic 7](#_Toc216555766)

[2.2 Historical Context of the Research Topic 7](#_Toc216555767)

[2.3 Theoretical Framework in Cybersecurity 8](#_Toc216555768)

[2.4 Key Concepts and Definitions 9](#_Toc216555769)

[2.5 Review of Relevant Literature 10](#_Toc216555770)

[2.7 Methodological Contributions 14](#_Toc216555771)

[2.8 Practical Implications 15](#_Toc216555772)

[2.8 Related Works 16](#_Toc216555773)

[CHAPTER THREE 19](#_Toc216555774)

[SYSTEM DESIGN AND METHODOLOGY 19](#_Toc216555775)

[3.1 Research Approach 19](#_Toc216555776)

[3.2 Research Design 19](#_Toc216555777)

[3.3 Software Development Methodology 19](#_Toc216555778)

[3.4 Data Collection Methods 20](#_Toc216555779)

[3.5 Ethical Considerations 21](#_Toc216555780)

[3.6 Tools and Technologies 22](#_Toc216555781)

[3.7 Data Analysis / Evaluation Plan 22](#_Toc216555782)

[3.8 Validity and Reliability 23](#_Toc216555783)

[3.9 System Requirements 24](#_Toc216555784)

[3.10 System Architecture / Workflow 24](#_Toc216555785)

[3.10.1 Model Training Pipeline 24](#_Toc216555786)

[3.10.2 System Components and Workllow 25](#_Toc216555787)

[3.11 System Modelling 27](#_Toc216555788)

[3.11.1 Use Case Diagram 27](#_Toc216555789)

[3.11.2 Activity Diagram 28](#_Toc216555790)

[3.11.3 Sequence Diagram 29](#_Toc216555791)

[CHAPTER FOUR 30](#_Toc216555792)

[IMPLEMENTATION AND TESTING 30](#_Toc216555793)

[4.1 System Requirements 30](#_Toc216555794)

[4.1.1 Hardware Requirements 30](#_Toc216555795)

[4.1.2 Software Requirements 30](#_Toc216555796)

[4.2 Implementation 31](#_Toc216555797)

[4.2.1 Data Acquisition and Preprocessing 31](#_Toc216555798)

[4.2.2 Model Training and Comparative Evaluation 33](#_Toc216555799)

[4.2.3 Streamlit Application Development 39](#_Toc216555800)

[4.3 Testing 45](#_Toc216555801)

[4.4 Results 45](#_Toc216555802)

[4.4.1 Data Collection and Preparation 45](#_Toc216555803)

[4.4.2 Comparative Model Analysis 48](#_Toc216555804)

[4.4.3 Significance and Implications 49](#_Toc216555805)

[4.4.4 Limitations and Considerations 49](#_Toc216555806)

[CHAPTER FIVE 52](#_Toc216555807)

[SUMMARY, CONCLUSION AND RECOMMENDATION 52](#_Toc216555808)

[5.1 Summary 52](#_Toc216555809)

[5.2 Conclusion 53](#_Toc216555810)

[5.3 Recommendation 54](#_Toc216555811)

[REFERENCE 55](#_Toc216555812)

**LIST OF TABLES**

[Table 1: Related Works 16](#_Toc216555972)

[Table 2: Comparative Performance of All Models 46](#_Toc216555973)

**LIST OF FIGURES**

[Figure 1: Sample of the IoT Blockchain Security Dataset 20](#_Toc216555889)

[Figure 2: Use Case Diagram 27](#_Toc216555890)

[Figure 3: Activity Diagram 28](#_Toc216555891)

[Figure 4: Sequence Diagram 29](#_Toc216555892)

[Figure 5: Data Preprocessing and Model Training Pipeline for IoT Blockchain Security Anomaly Detection 35](#_Toc216555893)

[Figure 6: Python Script for Training and Evaluation of Multiple Machine Learning Models for Anomaly Detection 36](#_Toc216555894)

[Figure 7: Model Performance Summary and Final Selection for IoT Blockchain Security Anomaly Detection 37](#_Toc216555895)

[Figure 8: Confusion Matrices for Five Machine Learning Models 38](#_Toc216555896)

[Figure 9: IoT Blockchain Security Anomaly Detection System: System Overview Dashboard 40](#_Toc216555897)

[Figure 10: IoT Blockchain Security Anomaly Detection System: Single Prediction - Threat Mitigated 41](#_Toc216555898)

[Figure 11: IoT Blockchain Security Anomaly Detection System: Single Prediction - Threat Active 42](#_Toc216555899)

[Figure 12: Batch Transaction Analysis Interface and Report for IoT Blockchain Security System 43](#_Toc216555900)

[Figure 13:Simulated Real-time Monitoring Dashboard for IoT Blockchain Security Anomaly Detection 44](#_Toc216555901)

[Figure 14: Model Evaluation Results on Test Set 46](#_Toc216555902)

## ABSTRACT

*Security vulnerabilities in Internet of Things (IoT) networks integrated with blockchain technology pose significant challenges for protecting distributed systems. This project develops a machine learning based anomaly detection system comparing five algorithms to identify the most effective approach for threat detection in resource constrained environments. The study evaluates Random Forest, Logistic Regression, Decision Tree, Gradient Boosting, and Isolation Forest for detecting DDoS attacks, data tampering, and unauthorized access. Using Python with Scikit-learn and Streamlit, models were trained on 1000 simulated IoT blockchain transactions. Isolation Forest achieved the best performance with 70% accuracy, 81.25% F1 score, and 90.91% recall, correctly identifying 130 out of 143 mitigated threats. Random Forest reached 67% accuracy with 93.01% recall, while other algorithms showed lower performance. A Streamlit application was developed with modules for single transaction analysis, batch processing, and real time monitoring. Despite dataset size limitations and class imbalance affecting active threat detection, the system demonstrates practical applicability for IoT blockchain security, providing insights for strengthening protection in critical infrastructure through machine learning approaches***.**

# **CHAPTER ONE**

# **INTRODUCTION**

The exponential growth of Internet of Things (IoT) devices has fundamentally transformed how we interact with technology in our daily lives. From smart homes to industrial automation, IoT ecosystems have become deeply embedded in critical infrastructure. However, this rapid expansion has introduced significant security vulnerabilities that threaten both individual privacy and organizational integrity. The integration of blockchain technology with IoT networks presents a promising solution to address these security concerns, yet it also introduces new complexities in threat detection and system monitoring.

Traditional security mechanisms often struggle to keep pace with the sophisticated and evolving nature of cyber threats targeting IoT networks. Conventional rule-based approaches lack the adaptability needed to identify novel attack patterns, while the distributed architecture of blockchain-integrated IoT systems creates additional challenges for centralized monitoring. This research explores how machine learning techniques can be leveraged to develop an intelligent anomaly detection system specifically designed for IoT-blockchain environments.

The heterogeneous nature of IoT devices, combined with the immutable characteristics of blockchain transactions, creates unique opportunities for applying comparative machine learning approaches. By analyzing network behavior patterns, transaction anomalies, and device-level activities, it becomes possible to construct a robust security framework capable of identifying threats that traditional methods might miss.

## 1.1 Background of Study

The convergence of IoT and blockchain technology represents one of the most significant developments in modern computing infrastructure. Developing and emerging economies have witnessed remarkable growth in IoT adoption across various sectors including healthcare, agriculture, transportation, and financial services. According to recent industry reports, the global IoT market is projected to grow substantially over the next decade, driven by increasing smartphone penetration and improved internet connectivity across diverse regions.

However, this growth trajectory brings with it serious security implications. IoT devices typically have limited computational resources, making it difficult to implement robust security protocols at the device level. Traditional security mechanisms in IIoT environments often struggle with issues related to scalability, efficiency, and vulnerability to various cyber threats (Okfie & Mishra, 2024). Additionally, the constant exchange of data between devices and centralized servers creates multiple potential entry points for malicious actors. When blockchain technology is introduced into this ecosystem to provide decentralization and data integrity, the complexity of security monitoring increases exponentially.

### **IoT Security Challenges**

IoT networks face distinct security challenges that differentiate them from traditional computer networks. The sheer volume of connected devices, estimated to reach billions globally, creates an enormous attack surface. Many IoT devices operate with default credentials, inadequate encryption, and infrequent security updates. The limited processing power of these devices also restricts the implementation of sophisticated security measures.

Common attack vectors in IoT environments include Distributed Denial of Service (DDoS) attacks, man-in-the-middle attacks, device hijacking, and data tampering. Each of these threats can have devastating consequences, particularly when IoT systems control critical infrastructure or handle sensitive personal information. The 2016 Mirai botnet attack, which compromised hundreds of thousands of IoT devices to launch massive DDoS attacks, demonstrated the catastrophic potential of IoT security vulnerabilities.

### **Blockchain Integration in IoT**

Blockchain technology has emerged as a potential solution to many IoT security concerns. By providing a decentralized, immutable ledger for recording transactions and device interactions, blockchain can enhance data integrity and establish trust in IoT networks without requiring central authority. Smart contracts enable automated enforcement of security policies, while cryptographic techniques ensure data confidentiality and authentication.

Despite these advantages, blockchain-integrated IoT systems face their own set of challenges. The consensus mechanisms required for blockchain operation can be computationally intensive, potentially overwhelming resource-constrained IoT devices. Transaction latency and energy consumption become critical concerns, particularly in applications requiring real-time responses. Furthermore, the transparency of blockchain transactions must be balanced against privacy requirements in sensitive applications.

### **Machine Learning for Anomaly Detection**

Machine learning has proven highly effective in identifying complex patterns and anomalies in large datasets. Unlike traditional rule-based systems that rely on predefined signatures of known threats, machine learning models can detect previously unseen attack patterns by learning the characteristics of normal system behavior. This adaptive capability makes machine learning particularly suitable for IoT-blockchain security, where threat landscapes evolve rapidly.

Several machine learning paradigms offer distinct advantages for anomaly detection. The distinctive quality of ML models is their ability to learn from large datasets and then detect deviations that signal potential malicious doings through autonomous pattern recognition (Sarma et al., 2023). Supervised learning algorithms can be trained on labeled datasets containing examples of both normal and malicious activities, enabling them to classify new observations with high accuracy. Unsupervised learning techniques excel at discovering hidden patterns in unlabeled data, making them valuable for detecting novel threats. Hybrid approaches that combine multiple algorithms can leverage the strengths of different methodologies to achieve superior detection performance.

## 1.2 Statement of Problem

The proliferation of IoT devices globally has outpaced the development of adequate security measures, leaving networks vulnerable to increasingly sophisticated cyber-attacks. While blockchain technology offers promising security enhancements, its integration with IoT systems creates new monitoring challenges that existing security solutions struggle to address effectively.

Current security approaches suffer from several critical limitations. Rule-based intrusion detection systems generate excessive false positives when applied to the dynamic behavior patterns typical of IoT networks. These systems also fail to detect zero-day attacks and novel threat variants that don't match predefined signatures. Meanwhile, the computational overhead of traditional security mechanisms often exceeds the processing capabilities of resource-constrained IoT devices.

The situation becomes more complex in blockchain-integrated environments. Transaction validation delays can mask suspicious activities, while the distributed nature of blockchain networks complicates centralized monitoring efforts. Energy consumption constraints further limit the deployment of computationally intensive security measures. Additionally, the lack of standardized security frameworks for IoT-blockchain systems means organizations must develop ad-hoc solutions that may not address all vulnerabilities.

Financial institutions, healthcare providers, and critical infrastructure operators face particular risks from these security gaps. A successful attack on IoT devices controlling payment systems, medical equipment, or power distribution networks could result in substantial financial losses, compromised patient safety, or disrupted essential services. The reputational damage from such incidents could also undermine public trust in digital transformation initiatives.

There is an urgent need for an intelligent security solution that can effectively monitor IoT-blockchain networks, accurately detect anomalies in real-time, and adapt to evolving threat patterns without overwhelming system resources. Such a solution must balance detection accuracy with computational efficiency while maintaining compatibility with diverse IoT devices and blockchain protocols.

## 1.3 Aim and Objectives

The aim of this research is to develop an IoT blockchain security anomaly detection system using a comparative machine learning approach to identify the most effective algorithm for threat detection in resource-constrained environments.

1. To identify and extract relevant features from IoT-blockchain network data for anomaly detection.
2. To train and evaluate five machine learning algorithms (Random Forest, Logistic Regression, Decision Tree, Gradient Boosting, and Isolation Forest) for threat detection.
3. To conduct comparative analysis of the trained models based on performance metrics including accuracy, precision, recall, and F1-score.
4. To select the best performing model and develop a functional anomaly detection system with real-time monitoring capabilities.
5. To evaluate the performance and efficiency of the developed system.

## 1.4 Significance of Study

This study is significant as it highlights the importance of strengthening security in IoT-blockchain systems and offers practical insights for organizations adopting these technologies. It compares different machine learning models to identify the most effective approach for detecting anomalies in integrated networks and introduces techniques for preparing diverse IoT and blockchain data for analysis. Traditional security controls involving intrusion detection systems with signature-based detection and rule-based firewalls have become less effective against the advancing threats of modern times, which makes advanced intelligent security systems essential for contemporary use" (Sarma et al., 2023). The findings may provide organizations with guidance on choosing suitable detection methods that work even with limited resources. Sectors such as finance, healthcare, and industry can apply these results to protect their systems and reduce cyber risks. The study also supports ongoing digital transformation and can help shape better security policies and standards.

## 1.5 Scope of Study

This study develops and evaluates machine learning methods for detecting anomalies in IoT systems that operate alongside blockchain technology. It compares five models, examines their performance, and considers how well they adapt to different types of threats. The work also reviews key blockchain consensus mechanisms and looks at major IoT network layers where attacks commonly occur. It focuses on threats such as DDoS, data tampering, and unauthorized access. The research does not cover physical attacks, social engineering, or large-scale real-world deployment. It relies on simulated datasets designed to reflect real IoT-blockchain environments while recognizing the limits of synthetic data.

## 1.6 Organization of the Study

Chapter One is the introduction of this study, comprising its background, statement of problem, research questions, aim and objectives, significance, scope, and empirical review of related works.

Chapter Two presents a review of existing literature on IoT security, blockchain technology, and machine learning-based anomaly detection systems. It examines key concepts, security threats, detection methodologies, and theoretical contributions related to the research topic.

Chapter Three outlines the methodology and system design used in the study. It details the research approach, dataset construction, feature engineering techniques, the five machine learning algorithms employed, preprocessing procedures, and evaluation metrics for comparative analysis.

Chapter Four covers the implementation, training, and evaluation of the IoT-blockchain anomaly detection system. It explains how the models were developed, tested, compared, and integrated into a functional system with simulated real-time monitoring.

Chapter Five summarizes the study's findings, draws conclusions on the comparative performance of the algorithms, discusses the implications of the developed system, and offers recommendations for future research and practical deployment. It also includes all references cited in the project, concluding with key insights and suggestions for further work.

**CHAPTER TWO**

## LITERATURE REVIEW

## 2.1 Introduction to the Project Topic

The intersection of Internet of Things (IoT) technology and blockchain systems has created a new frontier in cybersecurity research. As connected devices continue to proliferate across industries, the security challenges associated with protecting these networks have become increasingly complex. The integration of blockchain technology offers potential solutions through its decentralized and tamper-resistant characteristics, yet this combination also introduces unique monitoring and detection challenges that traditional security measures struggle to address.

Machine learning has emerged as a powerful tool for identifying anomalies in complex network environments. Unlike conventional rule-based systems that depend on predefined attack signatures, machine learning algorithms can adapt to evolving threat patterns by learning from network behavior. This adaptive capability makes machine learning particularly valuable for securing IoT-blockchain ecosystems where threats constantly evolve and traditional detection methods often fall short.

This literature review examines existing research on IoT security vulnerabilities, blockchain integration approaches, and machine learning-based anomaly detection systems. It explores how researchers have attempted to address security challenges in distributed networks and identifies gaps in current knowledge that this study aims to fill. The review also considers the theoretical foundations that underpin cybersecurity research and evaluates the practical implications of integrating machine learning with blockchain technology for threat detection.

## 2.2 Historical Context of the Research Topic

The concept of IoT emerged in the late 1990s, but widespread adoption only began in the 2010s as sensor technology became more affordable and internet connectivity improved. Early IoT implementations focused primarily on functionality and convenience, with security often treated as an afterthought. This approach led to numerous high-profile security incidents that exposed fundamental vulnerabilities in connected device networks.

The 2016 Mirai botnet attack marked a turning point in IoT security awareness. This attack compromised hundreds of thousands of poorly secured IoT devices and used them to launch massive, distributed denial of service attacks that disrupted major internet services. The incident demonstrated how vulnerable IoT devices could be weaponized and highlighted the urgent need for improved security mechanisms in connected device networks.

Blockchain technology, originally developed for cryptocurrency applications, began to attract attention as a potential solution for IoT security challenges around 2015. Researchers recognized that blockchain's decentralized architecture and cryptographic foundations could address many of the trust and integrity issues plaguing IoT networks. "Blockchain technology, known for its decentralized and tamper-resistant nature, has emerged as a promising solution to fortify the security of data exchanges" (Okfie & Mishra, 2024).

The application of machine learning to cybersecurity has evolved significantly over the past decade. Early systems relied heavily on signature-based detection methods that could only identify known threats. As computing power increased and datasets grew larger, researchers began exploring how machine learning algorithms could detect novel attacks by identifying deviations from normal behavior patterns. This shift toward intelligent, adaptive security systems has become particularly relevant for IoT-blockchain environments where threat landscapes change rapidly.

## 2.3 Theoretical Framework in Cybersecurity

The CIA triad Confidentiality, Integrity, and Availability, provides the foundational theoretical framework for understanding cybersecurity principles. This model has guided security research and practice for decades and remains highly relevant for evaluating security measures in IoT-blockchain systems.

Confidentiality ensures that information remains accessible only to authorized parties. In IoT-blockchain contexts, confidentiality involves protecting sensitive data transmitted between devices and ensuring that transaction details on the blockchain are visible only to appropriate participants. Encryption mechanisms and access control policies serve as primary tools for maintaining confidentiality.

Integrity focuses on maintaining the accuracy and trustworthiness of data throughout its lifecycle. This principle is particularly significant in blockchain systems where the immutability of recorded transactions is a core feature. "Once recorded, data on a blockchain is immutable and tamper-proof... ensuring that any alteration to a block would require altering all subsequent blocks" (Bello et al., 2024). For IoT devices, integrity mechanisms must prevent unauthorized modifications to sensor data and device configurations.

Availability addresses the requirement that systems and data remain accessible to authorized users when needed. IoT networks supporting critical infrastructure must maintain high availability despite potential attacks or system failures. Distributed denial of service attacks specifically target availability by overwhelming system resources and preventing legitimate access.

Beyond the traditional CIA triad, modern cybersecurity frameworks also consider authentication, authorization, and non-repudiation as essential components. Authentication verifies the identity of devices and users, while authorization determines what actions authenticated entities can perform. Non-repudiation ensures that parties cannot deny their actions, a feature that blockchain technology inherently provides through its transparent and traceable transaction records.

## 2.4 Key Concepts and Definitions

1. Internet of Things (IoT): A network of physical devices embedded with sensors, software, and connectivity capabilities that enable them to collect and exchange data. These devices range from simple sensors to complex industrial equipment and operate with varying levels of computational resources.
2. Blockchain: A distributed ledger technology that maintains a continuously growing list of records, called blocks, which are linked and secured using cryptography. Each block contains transaction data, a timestamp, and a cryptographic hash of the previous block, creating an immutable chain of information.
3. Anomaly Detection: The process of identifying patterns in data that do not conform to expected behavior. In cybersecurity contexts, anomalies often indicate potential security threats, system malfunctions, or policy violations that require investigation.
4. Machine Learning: A subset of artificial intelligence that enables systems to learn from data and improve their performance over time without being explicitly programmed. Machine learning algorithms identify patterns in training data and use these patterns to make predictions or decisions about new data.
5. Supervised Learning: A machine learning approach where algorithms learn from labeled training data containing examples of both normal and abnormal behavior. The algorithm uses these examples to develop a model that can classify new observations.
6. Unsupervised Learning: Machine learning techniques that work with unlabeled data to discover hidden patterns or structures. These methods are particularly useful for detecting novel threats that do not match known attack signatures.

## 2.5 Review of Relevant Literature

The reviewed literature on IoT security, blockchain integration, and machine learning-based detection systems reveals both significant progress and persistent challenges in securing connected device networks.

Research consistently identifies fundamental vulnerabilities that make IoT networks attractive targets for cyber attacks. "Massive device networks stemming from the rapid growth of Internet of Things devices became a security threat because they expanded exposure to cyberattacks" (Sarma et al., 2023). The scale of this challenge continues to grow as deployment accelerates across industries.

Resource constraints present a particularly difficult obstacle for implementing robust security measures. "Traditional security mechanisms in IIoT environments often struggle with issues related to scalability, efficiency, and vulnerability to various cyber threats" (Okfie & Mishra, 2024). Many IoT devices lack the processing power and memory needed to run sophisticated encryption or intrusion detection software, forcing security architects to balance protection with performance.

The decentralized architecture of IoT networks compounds these difficulties. "IoT networks' decentralized nature, resource-limited devices, and the vast amount of real-time data they generate, thus rendering them highly susceptible to cybersecurity issues" (Sathyabama & Katiravan, 2025). This distributed structure creates numerous potential attack vectors and makes centralized monitoring approaches impractical.

Detection capabilities have struggled to keep pace with evolving threats. "Detection of attacks in IoT and detecting malicious traffic in the early stages is a very challenging problem due to the increase in the size of network traffic" (Anwer et al., 2021). The volume and velocity of data generated by IoT networks overwhelm traditional security tools designed for more manageable data flows.

Critical infrastructure deployments raise the stakes considerably. "IoT networks have been integrated into industrial infrastructure schemes, positioning themselves as devices that communicate highly classified information for the most critical companies" (Vargas et al., 2021). When connected devices control essential services like power distribution or healthcare systems, security failures can have severe real-world consequences beyond data breaches.

Researchers have explored blockchain technology as a mechanism to address IoT security weaknesses. The technology's core characteristics align well with security requirements for distributed networks. "This distributed ledger approach enhances trust, authentication, and secure communication across IoT devices" (Sathyabama & Katiravan, 2025).

Blockchain's immutability provides particular value for maintaining data integrity. "By leveraging blockchain's immutable ledger, detected anomalies are recorded securely, preventing post-hoc alterations" (Lindgren, 2024). This feature ensures that security events cannot be erased or modified after detection, creating a reliable audit trail.

Transparency represents another important benefit. "Blockchain provides transparency by allowing all participants in the network to view the entire transaction history... promoting accountability and trust among users" (Bello et al., 2024). This visibility enables security teams to trace suspicious activities across the network and identify patterns that might indicate coordinated attacks.

However, blockchain integration introduces its own complications. "The immutable and distributed nature of blockchain also presents unique challenges for detecting anomalies and suspicious activities within the network" (Siddamsetti et al., 2024). The very characteristics that make blockchain attractive for security can also make threat detection more complex.

Resource requirements present practical deployment challenges. "Integrating blockchain directly in low-resource IoT systems has certain issues like added latency, increased bandwidth usage and additional computational overhead" (Jumma et al., 2025). These constraints force researchers to develop lightweight protocols that can operate within the limitations of typical IoT devices.

Machine learning has demonstrated considerable promise for identifying security threats in complex network environments. "Machine learning algorithms under supervised methods serve crucial functions in IoT security anomaly detection by using tagged data to train models that determine network patterns as normal or abnormal" (Sarma et al., 2023). This capability to learn from examples and generalize to new situations makes machine learning well-suited for evolving threat landscapes.

Different algorithms offer distinct advantages for security applications. "Advanced machine learning models, particularly Random Forest and XGBoost, are anticipated to achieve high accuracy in distinguishing between normal and malicious traffic" (Alwan et al., 2024). Ensemble methods like Random Forest combine multiple decision trees to improve prediction accuracy and reduce overfitting risks.

Deep learning architectures have shown particular effectiveness with large datasets. "Using multi-layer feature extraction, adaptive learning methods, and high computing capabilities, DNNs can evaluate large volumes of IoT network data" (Sathyabama & Katiravan, 2025). These sophisticated models can identify subtle patterns that simpler algorithms might miss.

Unsupervised approaches excel at detecting novel threats. "Isolation Forest focuses on isolating anomalies by constructing binary trees, while K-means clustering aims to identify abnormal instances based on their distance from cluster centroids" (Siddamsetti et al., 2024). These techniques do not require labeled training data, making them valuable when examples of specific attack types are unavailable.

Feature engineering significantly impacts model performance. "Feature selection techniques such as Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA) could significantly improve model efficiency and accuracy" (Alwan et al., 2024). Selecting the most relevant features reduces computational requirements while often improving detection capabilities.

Performance comparisons reveal important tradeoffs between different approaches. "Decision trees and rule-based classifiers comprise supervised models with superior interpretability compared to deep learning-based unsupervised models" (Sarma et al., 2023). While complex neural networks may achieve higher accuracy, simpler models offer transparency that helps security teams understand why particular activities were flagged as suspicious.

Recent research increasingly explores how machine learning and blockchain can work together to enhance security. "Combining machine learning and blockchain strategies allowed us to establish a strategy for identifying and mitigating attackers in real time in an IIoT network" (Vargas et al., 2021). This integration leverages the strengths of both technologies to create more robust security frameworks.

The synergy between these technologies addresses complementary weaknesses. "When combined with a blockchain, DL systems would have access to immutable, trustable datasets, while the blockchain would enjoy the intelligent, train-able filtering and classification of the DL models" (Jumma et al., 2025). Machine learning provides adaptive threat detection while blockchain ensures the integrity of detection records and system responses.

Several researchers have proposed specific architectures for integrated systems. "The study suggests a mechanism called Deep Blockchain-Enabled Collaborative Anomaly Detection (DBC-CAD) for security-focused distributed Anomaly Detection" (Ravuri et al., 2024). These frameworks typically distribute detection capabilities across multiple nodes while using blockchain to coordinate responses and maintain consistent security policies.

Logging security events to blockchain provides valuable forensic capabilities. "The logging of anomalous activities onto a blockchain prevents tampering, enhances transparency, and enables a verifiable trail of security events" (Lindgren, 2024). This audit trail supports incident investigation and can provide evidence for legal proceedings when necessary.

Performance evaluations of integrated systems show encouraging results. "With a low false-positive rate of 15.42% and a strong detection accuracy of 99.18%, the proposed model successfully identifies malicious activity, including malware injections" (Sathyabama & Katiravan, 2025). Such metrics suggest that combined approaches can achieve both high accuracy and low false alarm rates.

Comparative studies help identify optimal algorithm selections. "The GBDT model outperforms the other algorithms in terms of the recall, F1-score, and accuracy" (Wu et al., 2022). Understanding which algorithms perform best for specific threat types and network configurations guides practical implementation decisions.

**2.6 Theoretical Contributions**

The body of literature makes several important theoretical contributions to understanding security in IoT-blockchain systems. First, researchers have extended traditional cybersecurity frameworks to account for the unique characteristics of distributed, resource-constrained networks. The CIA triad remains relevant but requires adaptation to address blockchain's transparency requirements and IoT's operational constraints.

Second, the literature advances understanding of how machine learning models can be systematically evaluated for security applications. Researchers have developed metrics and methodologies for comparing different algorithms based not only on detection accuracy but also on computational efficiency, false positive rates, and adaptability to new threats. This theoretical foundation supports more rigorous evaluation of security solutions.

Third, studies have contributed to anomaly detection theory by identifying which types of deviations indicate genuine security threats versus benign system variations. This work helps refine the conceptual boundaries between normal network behavior and suspicious activities in IoT-blockchain contexts.

Finally, the literature has begun developing theoretical models for how blockchain-based logging and smart contract enforcement can complement algorithmic detection methods. These models explore how immutable records and automated responses create defense-in-depth strategies that go beyond simple threat identification.

## 2.7 Methodological Contributions

Research in this field has advanced detection methodologies in several ways. Studies have demonstrated effective approaches for feature extraction from heterogeneous IoT data streams and blockchain transaction records. These techniques enable machine learning models to process diverse data types while maintaining computational efficiency.

Researchers have also contributed methods for creating representative datasets that capture realistic attack scenarios in IoT-blockchain environments. Since real-world attack data can be difficult to obtain, these synthetic data generation approaches support model training and evaluation while protecting sensitive information. The literature includes valuable methodologies for comparative evaluation of multiple algorithms under consistent conditions. These frameworks enable fair assessment of different approaches and help identify which algorithms perform best for specific threat types or network configurations.

Additionally, studies have developed techniques for adapting machine learning models to resource-constrained IoT devices. These lightweight implementations make sophisticated detection capabilities practical even on devices with limited processing power and memory.

## 2.8 Practical Implications

The research reviewed has significant implications for organizations implementing IoT-blockchain systems. Studies demonstrate that machine learning-based detection can significantly improve security compared to traditional rule-based approaches, particularly for identifying novel attacks. "The algorithm proved to be superior in some aspects to traditional solutions such as IDS in the identification of more sophisticated attacks" (Vargas et al., 2021).

However, implementation requires careful consideration of tradeoffs. "Although the use of blockchain introduces a modest increase in latency, the trade-off is justified by the substantial gains in data transparency and system resilience" (Lindgren, 2024). Organizations must balance security benefits against performance impacts and resource requirements.

The literature suggests that supervised learning approaches generally provide better accuracy for known attack types, while unsupervised methods excel at detecting completely novel threats. Organizations should consider deploying hybrid systems that leverage both approaches depending on their specific threat landscapes and risk tolerances.

Resource constraints remain a practical challenge that affects deployment decisions. "Many ML algorithms impose additional processing and communication costs with the increase of data that is imminent for most IoT networks" (Waheed et al., 2020). Organizations must select algorithms that can operate within their devices' computational budgets while still providing adequate detection capabilities.

Finally, the research indicates that integrated systems combining machine learning and blockchain offer the most comprehensive security posture. "The integration of ML and blockchain offers a robust framework... paving the way for more advanced and automated compliance systems" (Bello et al., 2024). Organizations adopting both technologies can benefit from synergies that enhance overall system security and regulatory compliance.

## 2.8 Related Works

Table 1: Related Works

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S/N | Author(s) | Title | Year | Method | Findings | Limitations |
| 1 | Okfie & Mishra | Anomaly Detection in IIoT Transactions using Machine Learning... | 2024 | Lightweight Blockchain + Isolation Forest & K-Means | Isolation Forest achieved 92.90% accuracy. The system detected 429,713 anomalies in Bitcoin transaction data. | Trade-off between precision and recall leads to false positives; lack of standardized protocols. |
| 2 | Jumma et al. | A review paper: Blockchain security with IoT devices and deep-learning methods | 2025 | Review of Deep Learning (CNN, GAN, DRL) & Blockchain types | Blockchain secures data pipelines for learning models; decentralized storage reduces single points of failure. | High energy consumption, latency, and scalability issues when integrating blockchain with IoT. |
| 3 | Anwer et al. | Attack Detection in IoT using Machine Learning | 2021 | RF, SVM, GBDT on NSL-KDD | RF achieved the highest accuracy (85.34%). | Feature selection decreased accuracy; false positive/negative rates remain a concern. |
| 4 | Bello et al. | Integrating machine learning and blockchain... | 2024 | Conceptual Framework + Smart Contracts | Integration enhances detection speed/accuracy and reduces false positives. | Scalability issues with high transaction volumes. |
| 5 | Siddamsetti et al. | Anomaly Detection in Blockchain Using Machine Learning | 2024 | Isolation Forest & K-Means | Identified 10% of data as anomalies. | Challenges with real-time data processing. |
| 6 | Waheed et al. | Security and Privacy in IoT Using Machine Learning and Blockchain... | 2020 | Survey of ML + Blockchain | Integrated approaches offer better security/privacy. | Latency issues in blockchain; ML vulnerable to adversarial attacks. |
| 7 | Sunanda et al. | Enhancing IoT Network Security: ML and Blockchain for Intrusion Detection | 2024 | Red Fox Optimization + Bi-LSTM + Blockchain | Achieved ~98.9% accuracy. | Computational overhead; unsuitable for low-power IoT devices. |
| 8 | Lindgren | Anomaly Detection in IoT Sensor Networks Using Blockchain... | 2024 | Isolation Forest, LOF + Hyperledger Fabric | Blockchain-based model achieved 92.5% accuracy. | Increase in logging latency (1.1s avg). |
| 9 | Ahmad et al. | Cyber Threat Detection for IoT Using Machine Learning | N.D. | RF, SVM, GBDT | RF provided the best accuracy (85.34%). | Low accuracy detecting DoS, R2L, U2R. |
| 10 | Sarma et al. | Machine learning-based anomaly detection in IoT Security... | 2023 | Supervised vs Unsupervised models | RF 95.2%. Isolation Forest better for zero-day attacks. | Supervised models fail on unknown threats. |
| 11 | Sathyabama & Katiravan | Enhancing anomaly detection using deep neural networks and blockchain... | 2025 | DNN + Private Blockchain | 99.18% detection accuracy. | Vulnerable to adversarial attacks; blockchain latency. |
| 12 | Ravuri et al. | Blockchain-enabled collaborative anomaly detection for IoT security | 2024 | Modified-LSTM + Ethereum | M-LSTM achieved 99.1%. | High computational burden; complex deployment. |
| 13 | Wu et al. | Blockchain-Based IoT: Machine Learning Tea Sensing... | 2022 | AdaBoost, XGBoost, GBDT, RF | GBDT and RF highest accuracy. | One-hot encoding limitations; privacy concerns. |
| 14 | Ahmed et al. | Detection of DDoS attacks in IoT Networks using ML | 2024 | RF, SVM, NB, XGBoost, KNN + RFE/PCA | RF & XGBoost high accuracy. | IoT variability reduces generalizability. |
| 15 | Vargas et al. | Detection of Security Attacks in Industrial IoT Networks... | 2021 | KNN + Lightweight Blockchain | Detected 100% of DoS attacks. | Implementation challenges in real IIoT. |

**CHAPTER THREE**

**SYSTEM DESIGN AND METHODOLOGY**

## 3.1 Research Approach

This study will adopt a quantitative research approach to develop and evaluate a machine learning-based anomaly detection system for IoT-blockchain security. The quantitative methodology is appropriate because the research involves numerical data analysis, statistical comparisons, and measurable performance metrics. The approach will focus on collecting structured data, training multiple machine learning algorithms, and comparing their effectiveness using established performance indicators.

## 3.2 Research Design

The research design will follow a systematic process that begins with problem identification and ends with a deployed anomaly detection system. The design incorporates both theoretical and practical elements to ensure the research contributes to academic knowledge while also producing a usable security tool.

## 3.3 Software Development Methodology

The development of the anomaly detection system will follow the Agile software development methodology. Agile is well-suited for this project because it emphasizes iterative development, continuous testing, and flexibility to adapt to new findings as the research progresses. Rather than trying to build the entire system at once, development will occur in short cycles called sprints, each lasting about two weeks.

Each sprint will focus on a specific component of the system. For example, one sprint might concentrate on data preprocessing, while another focuses on training a particular machine learning algorithm. At the end of each sprint, the work will be reviewed and tested to ensure it meets quality standards. This iterative approach allows problems to be identified and corrected early, before they can affect later stages of development.

Agile methodology also encourages regular reflection on the development process. After completing each sprint, time will be allocated to review what went well and what could be improved. This reflection helps optimize the workflow and ensures that lessons learned in one sprint can be applied to subsequent sprints. The flexibility of Agile means that if initial results suggest a different approach would be more effective, the research plan can be adjusted accordingly.

## 3.4 Data Collection Methods

The primary data for this research will be obtained from a publicly available dataset hosted on Kaggle. The dataset titled IoT Blockchain Security Dataset contains 1000 simulated IoT network transactions with security-related attributes, blockchain parameters, and attack mitigation outcomes. This dataset was chosen because it provides a comprehensive collection of features relevant to IoT-blockchain security and includes labeled examples of both successful and unsuccessful threat mitigation.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 1: Sample of the IoT Blockchain Security Dataset**

The dataset includes several categories of information that are essential for training anomaly detection models. IoT device activity data covers various types of network operations including data transmission, authentication requests, encrypted data transfers, and smart contract execution. These different activity types represent the normal operations that occur in IoT networks and provide examples of baseline behavior against which anomalies can be detected.

Security threat information in the dataset includes multiple attack types such as Distributed Denial of Service attacks, data tampering, unauthorized access attempts, man-in-the-middle attacks, and eavesdropping. Each transaction is labeled with the type of security threat present and a severity rating from zero to ten. This information allows the machine learning models to learn the characteristics of different threat types and understand how severity levels relate to mitigation success.

Blockchain-related metrics in the dataset include transaction confirmation times, consensus mechanism types, and energy consumption measurements. These features are important because blockchain integration affects how IoT networks operate and how security measures perform. Transaction times can indicate network congestion or processing delays, while consensus mechanisms influence the computational resources required for validation. Energy consumption is particularly relevant for resource-constrained IoT devices where excessive energy use can indicate malicious activity.

The target variable in the dataset indicates whether each security threat was successfully mitigated, with zero representing unsuccessful mitigation and one representing successful mitigation. This binary classification allows the research to frame the problem as a supervised learning task where models learn to predict mitigation outcomes based on the characteristics of each transaction. The presence of both positive and negative examples enables models to learn the patterns that distinguish successfully mitigated threats from active threats.

While the dataset consists of simulated rather than real-world data, simulation is a common and accepted practice in cybersecurity research. Real attack data is difficult to obtain due to privacy concerns and the sensitive nature of security incidents. Simulated data allows researchers to create diverse scenarios covering multiple attack types and network conditions without exposing actual systems to risk. The dataset will be examined carefully to ensure it contains realistic values and appropriate distributions that reflect actual IoT-blockchain environments.

## 3.5 Ethical Considerations

This research avoids providing information that could be misused to harm IoT systems and limits technical details to defensive purposes. It maintains data privacy awareness, even though simulated data is used, and considers compliance requirements for real deployments. Results are reported honestly, including limitations and weak model performance. Sharing of models or code is balanced against the risk of misuse. The study also acknowledges the limits of synthetic data to prevent unrealistic expectations during real-world deployment.

## 3.6 Tools and Technologies

The implementation of this project will utilize several key tools and technologies selected for their capabilities in machine learning, data processing, and web application development.

1. Python: Primary programming language for model development and system implementation
2. Scikit-learn: Provides machine learning algorithms, preprocessing tools, and evaluation metrics
3. Pandas: Used for data loading, cleaning, transformation, and structured data handling
4. NumPy: Supports numerical operations and efficient array-based computations
5. Matplotlib & Seaborn: Generate visualizations for analysis and model performance
6. Streamlit: Builds the interactive web interface for the anomaly detection system
7. Joblib: Saves and loads trained models for deployment without retraining
8. Plotly: Creates interactive charts for an engaging and detailed user experience
9. Jupyter Notebook, VS Code & Git: Support development, experimentation, and version control

## 3.7 Data Analysis / Evaluation Plan

The performance of the model is evaluated using the following metrics:

**Accuracy:** The proportion of correctly classified instances among the total instances.

**Formula:** Accuracy=(TP+TN) / (TP+TN+FP+FN)

​**Precision**: The proportion of true positive transactions (fraudulent transactions correctly identified) among the instances classified as fraudulent by the model, indicating the model’s ability to avoid false positives.

**Formula:** Precision=TP/(TP+FP)

**Recall**: The proportion of true positive instances (fraudulent transactions correctly identified) among all the actual fraudulent transactions, measuring the model’s sensitivity and its ability to detect anomalies.

**Formula:** Recall=TP/(TP+FN)

**F1-Score:** The harmonic mean of precision and recall, which gives a balanced measure regarding the performance of the model. It is especially useful when there is an imbalance between the number of normal and anomalous transactions (Sokolova & Lapalme, 2009).

**Formula:** F1=2\*(precision\*recall)/(precision + recall)

**Confusion Matrix:** This will visualize the types of errors each model makes. A confusion matrix shows true positives, true negatives, false positives, and false negatives in a tabular format. This breakdown helps identify whether a model has systematic biases, such as consistently misclassifying one particular type of threat.

**ROC curves and AUC scores**: This will assess model performance across different classification thresholds. Receiver Operating Characteristic curves plot true positive rate against false positive rate at various threshold settings. The Area Under the Curve provides a single number summarizing performance across all thresholds, with values closer to one indicating better performance.

**Precision-Recall curves**: This will provide additional insight, particularly for imbalanced datasets. These curves show how precision and recall trade off as the classification threshold changes. They are often more informative than ROC curves when dealing with imbalanced classes because they focus on the performance for the positive class.

**Key Components**

1. True Positives (TP): Fraud cases correctly detected as fraud.
2. True Negatives (TN): Genuine transactions correctly identified as genuine.
3. False Positives (FP): Genuine transactions wrongly flagged as fraud (Type I error).
4. False Negatives (FN): Fraud cases wrongly classified as genuine (Type II error).

## 3.8 Validity and Reliability

1. Internal Validity: Ensured through proper data preprocessing and consistent model evaluation techniques.
2. Reliability: The system was tested on a separate test set and results were reproducible through saved models (joblib).
3. External Validity: Though the dataset is synthetic, it reflects real-world financial behavior, providing reasonable external applicability.

## 3.9 System Requirements

System requirements categorized using MoSCoW.

Functional Requirements:

1. MUST HAVE: Single device prediction, batch prediction, simulated real-time monitoring, and model loading with confidence display.
2. SHOULD HAVE: Prediction visualizations, summary metrics for batch analysis, and CSV report export.
3. COULD HAVE: Detailed dashboards, advanced batch filtering, and adjustable simulation parameters.
4. WON'T HAVE: Database integration, real-time streaming from devices, user authentication, and automated model retraining.

Non-Functional Requirements:

1. MUST HAVE: Intuitive Streamlit UI and fast prediction performance.
2. SHOULD HAVE: Reliability and Maintainability.
3. COULD HAVE: Scalability and basic security.
4. WON'T HAVE: High availability and advanced security.

## 3.10 System Architecture / Workflow

This section provides a detailed explanation of the system's architecture and the flow of data and control within the IoT-blockchain security threat detection system. The system is designed to process IoT-blockchain transaction data, utilize a pre-trained machine learning model to identify potential security threats, and present the results to the user in an accessible manner.

### **3.10.1 Model Training Pipeline**

The model training process will follow a systematic pipeline designed to ensure fair comparison across all five algorithms. First, the preprocessed and scaled training data will be fed into each algorithm with carefully configured hyperparameters. For supervised algorithms (Random Forest, Logistic Regression, Decision Tree, and Gradient Boosting), class weights will be balanced to address the moderate class imbalance in the dataset, preventing the models from being biased toward the majority class. Random Forest will be configured with 150 trees and constrained depth to prevent overfitting, while Logistic Regression will use L2 regularization with the SAGA solver for efficient convergence. Decision Tree will be limited in depth and minimum samples per leaf to maintain generalizability, and Gradient Boosting will employ a low learning rate with shallow trees to build the ensemble gradually. Isolation Forest, being unsupervised, will be trained differently, it will learn patterns from the feature space without labels, with its contamination parameter set based on the proportion of threats in the training data to guide how many instances should be considered anomalies. All models will use the same random seed (42) to ensure reproducibility. After training, each model will be evaluated on the test set using accuracy, precision, recall, F1-score, and confusion matrices. The model with the highest F1-score will be selected as the best performer, as this metric provides the most balanced assessment of classification quality, particularly important in security applications where both false positives and false negatives carry significant costs. All trained models will be serialized using Joblib and saved to disk, allowing them to be loaded quickly during deployment without requiring retraining. This comprehensive training approach ensures that the comparative analysis is rigorous and that the selected model represents the most effective solution for IoT-blockchain threat detection.

### **3.10.2 System Components and Workllow**

The system architecture is structured to handle both batch processing of transaction data and individual, real-time transaction analysis. The primary components and their interactions are described below:

The system handles both batch and individual transaction analysis.

1. Data Input: The system accepts data in three ways: First, via CSV upload for historical or new transactions. Second, via manual input of IoT transaction details through the Streamlit application. Third, via simulated real-time transaction streams for monitoring purposes.
2. Data Preprocessing (Implicit): Categorical variables such as IoT layer type, request type, threat classification, and consensus mechanism are encoded numerically. Numerical features are scaled using the pre-fitted StandardScaler. Batch data is assumed to be pre-processed, while manual or simulated inputs are processed in real-time.
3. Threat Detection Model (Trained ML Model): A pre-trained machine learning model (best performing model from comparative analysis) is used to classify transactions as threats mitigated or active. The model is loaded along with its preprocessing objects to ensure accurate predictions.
4. Prediction Engine: The prediction engine operates in three modes. For batch predictions, Pandas reads CSV files, features are extracted, preprocessed, and passed to the model for prediction. For single transaction predictions, input values are gathered, encoded, scaled, and passed to the model for immediate assessment. For real-time monitoring, simulated transactions are continuously processed through the same pipeline, updating live metrics.
5. Output and Visualization: Streamlit presents predictions and optional analyses. For single transaction inputs, the interface displays whether the threat was mitigated or remains active, along with a confidence score. For batch analysis, summary statistics, threat type breakdowns, and visualizations are provided. Real-time monitoring includes live metrics, timeline visualizations, and activity feeds showing recent detections.
6. User Interface (Streamlit Application): The Streamlit application provides navigation between modules, including dashboard, single prediction, batch analysis, real-time monitoring, and documentation. Users can input data, view predictions, analyze results, download reports, and access guidance for operating the system effectively.

Workflow:

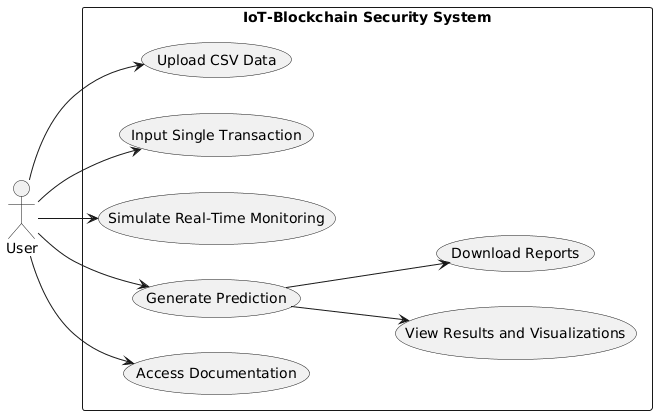
1. The user interacts with the system via Streamlit.
2. The user uploads a CSV, enters a single transaction manually, or initiates real-time monitoring.
3. CSV or manual input data is read and preprocessed; simulated transactions are generated and processed.
4. Categorical variables are encoded and numerical features scaled.
5. Data is passed to the trained machine learning model.
6. The model generates predictions (threat mitigated or active) along with confidence scores.
7. Streamlit displays predictions, metrics, and visualizations appropriate to the module.
8. For batch analysis, summary statistics and CSV reports are generated. For real-time monitoring, live dashboards and activity feeds are continuously updated.

## 3.11 System Modelling

This section presents the system modeling using three UML diagrams to describe the behavior and structure of the anomaly detection system.

### **3.11.1 Use Case Diagram**

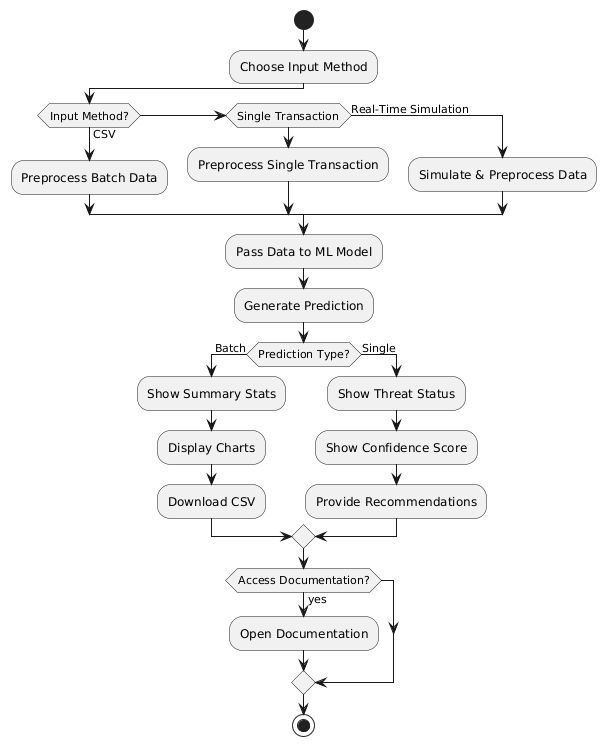
A Use Case Diagram illustrates the interactions between actors (users or external systems) and the system itself. It describes the goals that actors have when using the system.



**Figure 2: Use Case Diagram**

### **3.11.2 Activity Diagram**

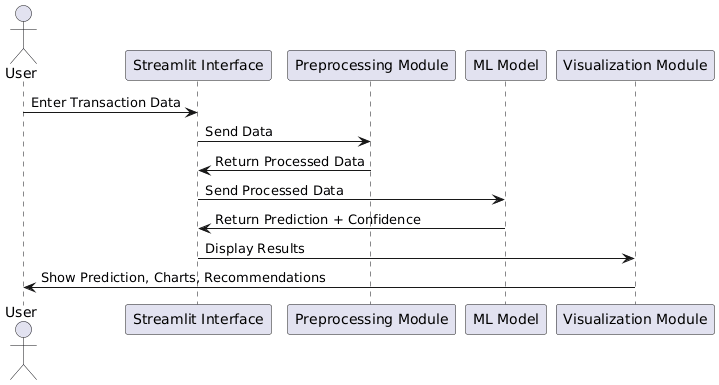
An Activity Diagram shows the flow of activities within a system. It's similar to a flowchart but can also represent parallel activities.



**Figure 3: Activity Diagram**

### **3.11.3 Sequence Diagram**

A Sequence Diagram illustrates how objects interact, and the sequence of messages exchanged between them to achieve a particular use case.



**Figure 4: Sequence Diagram**

# **CHAPTER FOUR**

# **IMPLEMENTATION AND TESTING**

This chapter presents the practical implementation of the IoT-blockchain security anomaly detection system. It covers the system requirements necessary for development and deployment, the detailed implementation process including data preprocessing and model training, and the testing procedures used to validate system functionality and performance.

## 4.1 System Requirements

This section outlines the minimum system requirements for the development and operation of the IoT-blockchain security anomaly detection system. These specifications ensure that the system functions correctly and can be utilized effectively for threat detection in IoT environments.

### **4.1.1 Hardware Requirements**

The hardware requirements for this project are as follows. Note that these are not the *absolute* minimum but represent a reasonable configuration for development and deployment.

1. Processor: Intel Core i3 or equivalent. Sufficient processing power for training the machine learning model and running the Streamlit application.
2. RAM: 4 GB. Allows for efficient handling of the dataset and smooth execution of the Python code.
3. Storage: 20 GB of free disk space. Necessary to store the dataset, the trained machine learning model, and the application files.

### **4.1.2 Software Requirements**

The software requirements for this project are as follows:

1. Operating System: Windows 10/11, macOS, or a Linux distribution.
2. Python: Version 3.8 or higher.
3. Required Python Libraries: Pandas – For data manipulation, CSV handling, and IoT transaction analysis. NumPy – For numerical computations and array operations. Scikit-learn – For machine learning algorithms (Random Forest, Logistic Regression, Decision Tree, Gradient Boosting, Isolation Forest), data preprocessing, model evaluation metrics, and cross-validation. Joblib – For saving and loading trained models and preprocessing objects. Streamlit – For building the interactive web application with multiple modules. Matplotlib – For generating static visualizations including confusion matrices and performance charts. Seaborn – For enhanced statistical plots and heatmaps. Plotly – For creating interactive visualizations in the web interface
4. Web Browser: A modern web browser (e.g., Chrome, Firefox, Safari) for accessing the Streamlit application.

## 4.2 Implementation

This section details the implementation of the IoT-blockchain security anomaly detection system, explaining how the system was built and how its components work together. The system was implemented using Python and several key libraries for machine learning, data processing, and web application development. The core of the system consists of five machine learning models trained comparatively to identify security threats in IoT-blockchain networks, with the best performing model (Isolation Forest) selected for deployment. A Streamlit application was developed to provide a user-friendly interface for threat analysis, batch processing, and real-time monitoring.

The system implementation can be broken down into the following stages:

1. Data Acquisition and Preprocessing
2. Model Training and Comparative Evaluation
3. Application Development

### **4.2.1 Data Acquisition and Preprocessing**

The first stage of implementation involved obtaining and preparing the IoT-blockchain security dataset for machine learning analysis. The dataset, sourced from Kaggle, contains 1000 simulated IoT network transactions with comprehensive security attributes, blockchain parameters, and threat mitigation outcomes.

**Data Loading**

The dataset was imported using Pandas for efficient CSV handling and initial exploration. This revealed the structure, feature types, and overall distribution of the target classes.

**Feature Engineering**

The dataset includes categorical and numerical variables. Categorical features such as IoT Layer, Request Type, Threat Type, and Consensus Mechanism were transformed using Label Encoding, converting each unique category into numeric form suitable for machine learning.

**Feature Selection**

Nine key features were chosen for training based on relevance to IoT-blockchain security behavior:

1. IoT Layer (encoded)
2. Request Type (encoded)
3. Data Size (KB)
4. Processing Time (ms)
5. Security Threat Type (encoded)
6. Attack Severity
7. Blockchain Transaction Time (ms)
8. Consensus Mechanism (encoded)
9. Energy Consumption (mJ)

These features collectively capture network activity, threat characteristics, and blockchain performance indicators.

**Data Splitting**

The data was split into 80% training and 20% testing using stratified sampling to preserve the original class balance.

**Feature Scaling**

Numerical features were standardized using StandardScaler, ensuring all values share a similar scale. This prevents larger-scaled features (e.g., Data Size) from overwhelming smaller ones (e.g., Processing Time) during model training. The scaler was fitted only on training data to avoid leakage.

**Saving Preprocessing Objects**

All encoders and the scaler were saved with Joblib to ensure consistent preprocessing during deployment, allowing new incoming data to be transformed in the same way as the training dataset.

### **4.2.2 Model Training and Comparative Evaluation**

The second stage involved training five different machine learning algorithms and conducting comprehensive comparative analysis to identify the most effective approach for IoT-blockchain anomaly detection.

**Algorithm Selection**

Five algorithms were selected based on their documented effectiveness for anomaly detection and classification tasks:

1. **Random Forest** – An ensemble method that combines multiple decision trees to improve prediction accuracy and reduce overfitting
2. **Logistic Regression** – A linear model suitable for binary classification that provides interpretable coefficients
3. **Decision Tree** – A tree-based model that makes decisions through a series of questions about feature values
4. **Gradient Boosting** – An ensemble technique that builds models sequentially, with each model correcting errors from previous ones
5. **Isolation Forest** – An unsupervised algorithm specifically designed for anomaly detection that isolates outliers using random partitioning

**Model Configuration and Training**

Each algorithm was configured with parameters optimized for the IoT-blockchain security task. Class weights were balanced for supervised models to address the moderate class imbalance in the dataset.

Random Forest with 150 trees was configured with depth and sample constraints to prevent overfitting. The 'sqrt' setting for max\_features means each tree considers the square root of total features when making splits, introducing randomness that improves ensemble diversity.

Logistic Regression was configured with L2 regularization (C=0.5) to prevent overfitting and the SAGA solver for efficient training on medium-sized datasets.

Decision Tree was constrained with maximum depth and minimum sample requirements to create a simpler tree that generalizes better to unseen data.

Gradient Boosting was configured with a low learning rate (0.08) and shallow trees (depth=4) to build the ensemble gradually and avoid overfitting.

Isolation Forest was trained using only feature data without labels, as it is an unsupervised algorithm. The contamination parameter was set based on the proportion of threats in the training data, telling the algorithm approximately what percentage of data points to consider as anomalies.

**Model Evaluation**

All models were evaluated on the test set using multiple performance metrics:

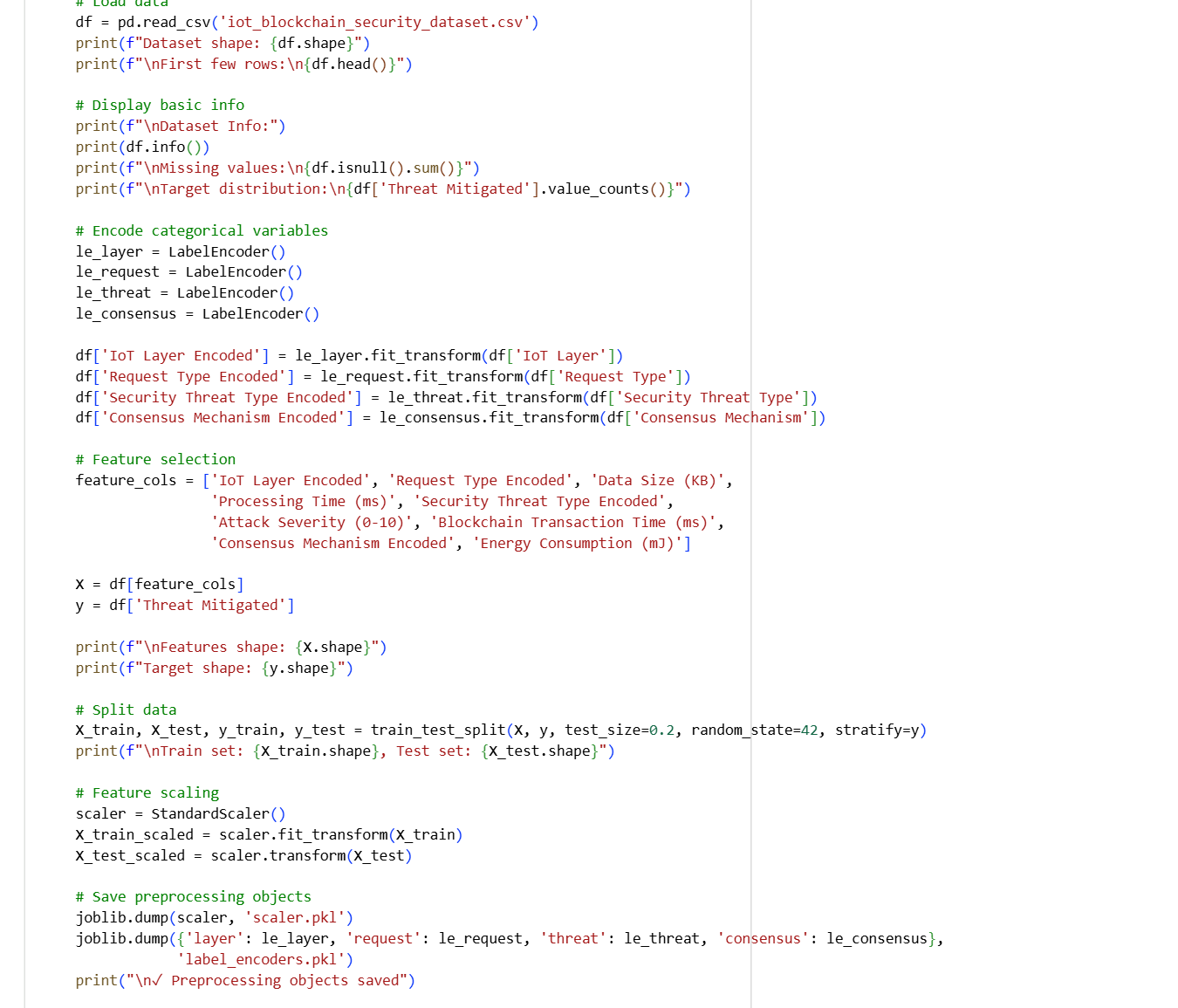
**Model Selection Justification**

Isolation Forest was selected as the best model based on its superior F1-score, which provides a balanced measure of precision and recall. The model's high recall (90.91%) is particularly valuable for security applications where detecting threats is critical. Additionally, Isolation Forest's unsupervised nature makes it well-suited for detecting novel attack patterns that were not present in the training data, addressing one of the key challenges identified in the literature review regarding zero-day threats.

The model's ability to identify anomalies without requiring labeled examples of every attack type provides flexibility and adaptability that supervised models lack. This characteristic aligns with the research objective of developing a system that can adapt to evolving threat patterns in IoT-blockchain environments.

**Model Persistence**

All trained models were saved for future use. Saving models allows them to be loaded quickly during deployment without requiring retraining, which is especially important for production environments where response time matters.



**Figure 5: Data Preprocessing and Model Training Pipeline for IoT Blockchain Security Anomaly Detection**

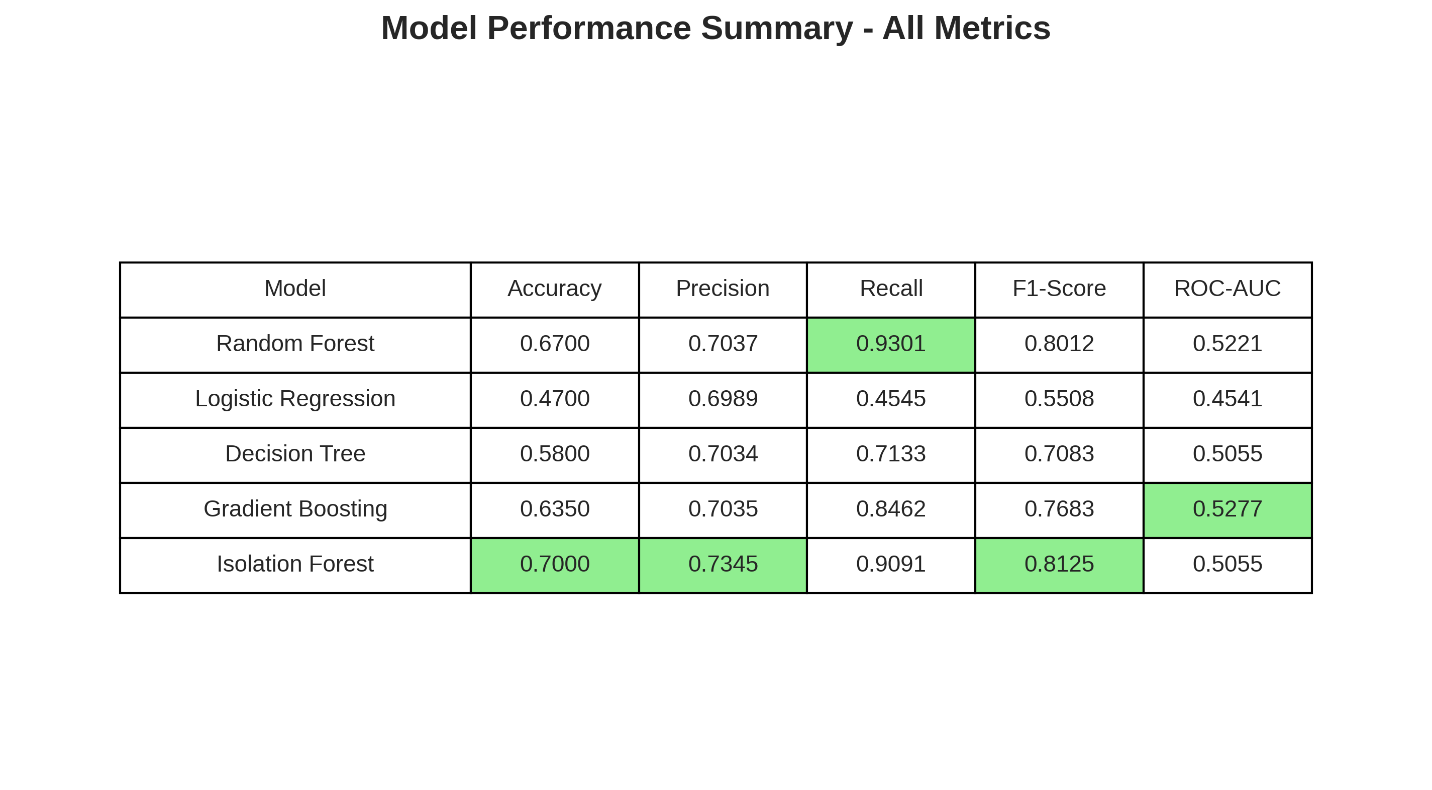
This script details the essential first steps in building the security system's threat detection model, combining both data preparation and initial training into a seamless workflow. The process begins with Data Preparation, where the raw IoT security dataset is loaded, cleaned up (categorical features are converted to numbers using encoding, and numerical values are standardized using scaling), and split into training and testing sets. Immediately following this, the Model Training phase kicks off, utilizing the prepared data to train five different machine learning algorithms including supervised classifiers like Random Forest and the unsupervised Isolation Forest providing a set of robust and diverse models ready for real-time anomaly detection.

A screenshot of a computer program

AI-generated content may be incorrect.

**Figure 6: Python Script for Training and Evaluation of Multiple Machine Learning Models for Anomaly Detection**

Following data preprocessing and feature scaling, this script executes the crucial Model Training and Evaluation phase (STEP 2) for the IoT Blockchain Security Anomaly Detection System. The code initializes, trains, and stores five distinct machine learning models on the prepared dataset (X\_train\_scaled and Y\_train ).



**Figure 7: Model Performance Summary and Final Selection for IoT Blockchain Security Anomaly Detection**

This figure displays the Comprehensive Model Evaluation Results and the final Model Selection stage of the machine learning pipeline. The Summary Table compares the performance of five algorithms (Random Forest, Logistic Regression, Decision Tree, Gradient Boosting, and Isolation Forest) on the test dataset using five key metrics, including the ROC-AUC score. Crucially, the Isolation Forest model achieved the highest F1-Score of 0.8125, which is critical for balancing Precision and Recall in anomaly detection tasks where misclassifying a threat can be costly. Due to this superior F1-Score, Isolation Forest is officially selected as the best model, achieving an Accuracy of 0.7000 and a Recall of 0.9091. While its ROC-AUC of 0.5055 is relatively low (indicating poor performance in distinguishing between the two classes across all thresholds), the high F1-Score is prioritized because the model's strength lies in identifying and isolating anomalies rather than making accurate predictions across a balanced dataset, thus it remains highly effective for the specific task of threat detection . Following selection, the best model and a final performance report are saved for deployment.

A screenshot of a graph

AI-generated content may be incorrect.

**Figure 8: Confusion Matrices for Five Machine Learning Models**

This figure displays five separate Confusion Matrices that provide a direct, count-based visualization of each model's predictive performance on the test set. The matrices show predictions versus the True Labels: 'Threat Active' (representing unsuccessful mitigation, the negative class, or Class 0) and 'Threat Mitigated' (representing successful mitigation, the positive class, or Class 1) .The matrix for Isolation Forest clearly shows 130 True Positives (correctly identifying successfully mitigated threats) but only 10 True Negatives (correctly identifying actively remaining threats, or unsuccessful mitigation). This heavy skew, resulting in a low Recall (0.179) and F1-Score (0.250) for the 'Threat Active' (Class 0) category, is expected. Isolation Forest is an unsupervised anomaly detection algorithm that is specifically optimized to find and isolate the minority class (anomalies/threats). Since the model is primarily designed to distinguish anomalies from the bulk of the data, its predictive accuracy naturally decreases when classifying the non-anomalous (majority) class, thus explaining its poor performance on Class 0.

### **4.2.3 Streamlit Application Development**

The third stage involved developing a user-friendly web application using Streamlit to make the trained models accessible for practical threat detection tasks.

Application Architecture

The Streamlit application was designed with a modular structure consisting of five main sections:

1. Dashboard – System overview and quick start guide
2. Single Prediction – Individual transaction analysis
3. Batch Analysis – Multiple transaction processing
4. Real-time Monitoring – Simulated continuous surveillance
5. Documentation – Comprehensive system information

**Single Prediction Module**

The single prediction module allows users to input characteristics of an individual IoT transaction and receive immediate threat assessment. The module provides detailed analysis including device information, threat assessment, system performance metrics, and recommended actions based on the prediction outcome.

**Batch Analysis Module**

The batch analysis module processes multiple transactions simultaneously. The module generates summary statistics, visualizations showing threat distribution and severity analysis, and provides CSV export functionality for further investigation.

**Real-time Monitoring Module**

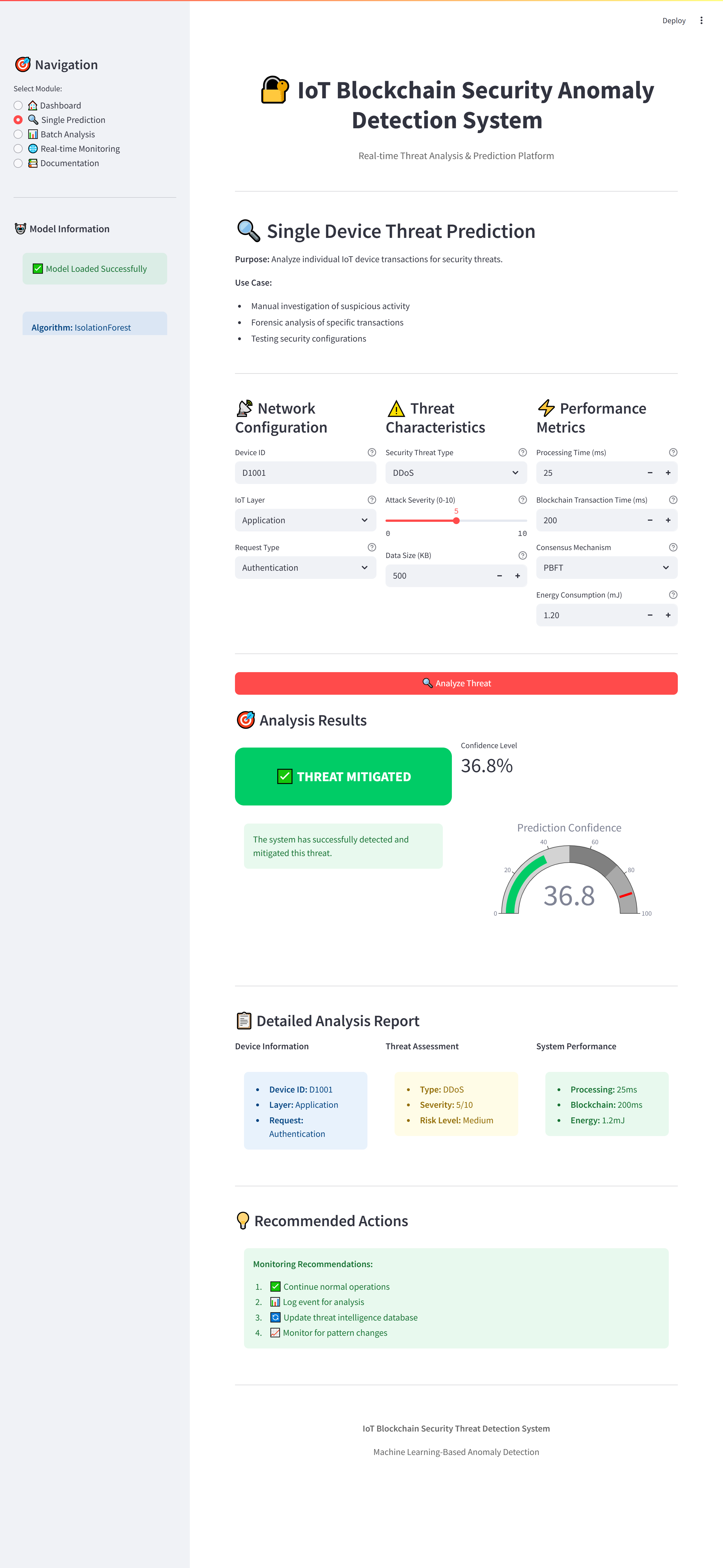
The real-time monitoring module simulates continuous network surveillance. The module maintains live metrics showing total transactions, active threats, and mitigation rates, along with timeline visualizations and an activity feed displaying recent detections.

A screenshot of a computer

AI-generated content may be incorrect.

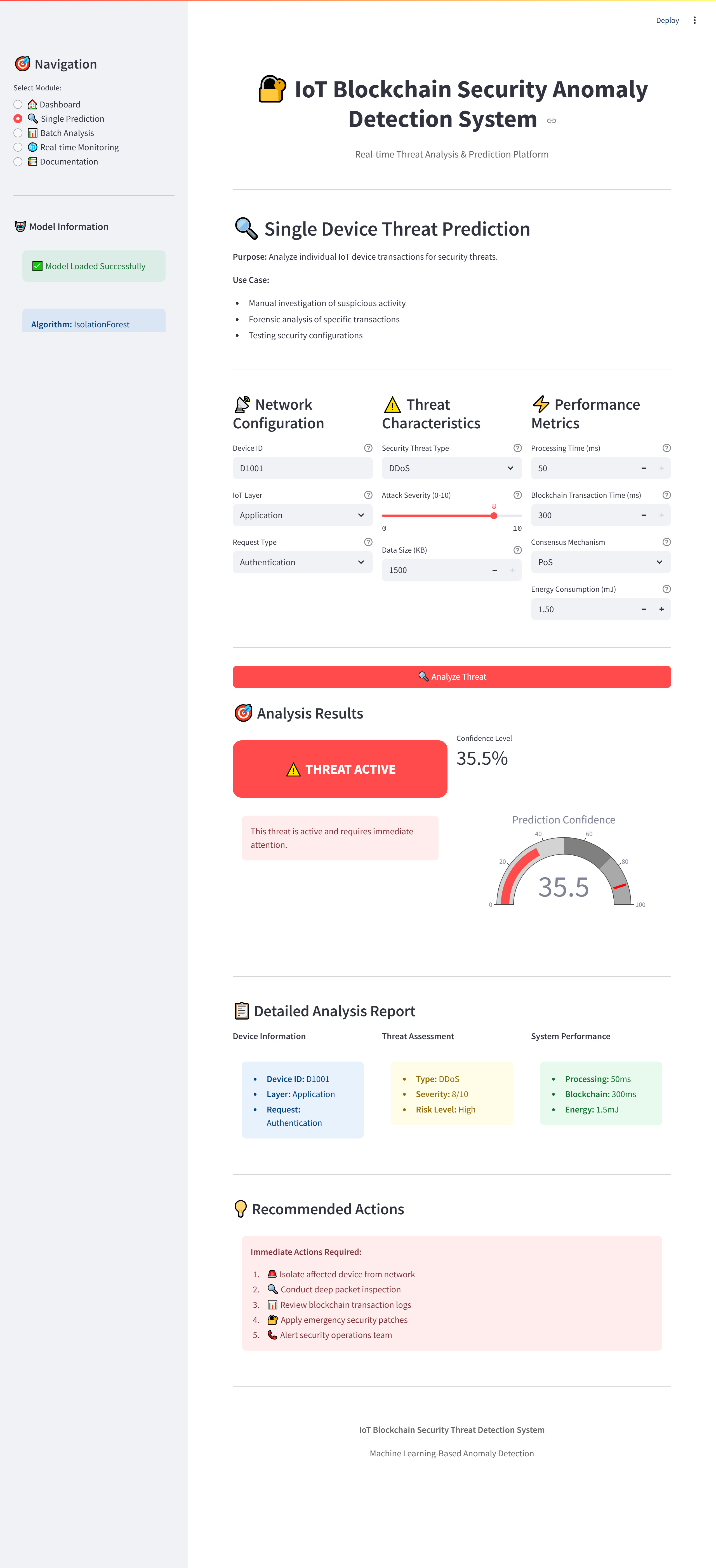
**Figure 9: IoT Blockchain Security Anomaly Detection System: System Overview Dashboard**

The application's layout is primarily divided into two main sections. The left panel serves as a Navigation and Model Information area, allowing users to select different functional modules like Single Prediction, Batch Analysis, and Real-time Monitoring. The main central section functions as the System Overview Dashboard, providing high-level information on the system's capabilities, supported threats, and blockchain integration features. It also includes a Quick Start Guide outlining the main workflow steps: Single Prediction (for individual analysis) and Batch Analysis (which involves uploading CSV files for comprehensive data analysis).



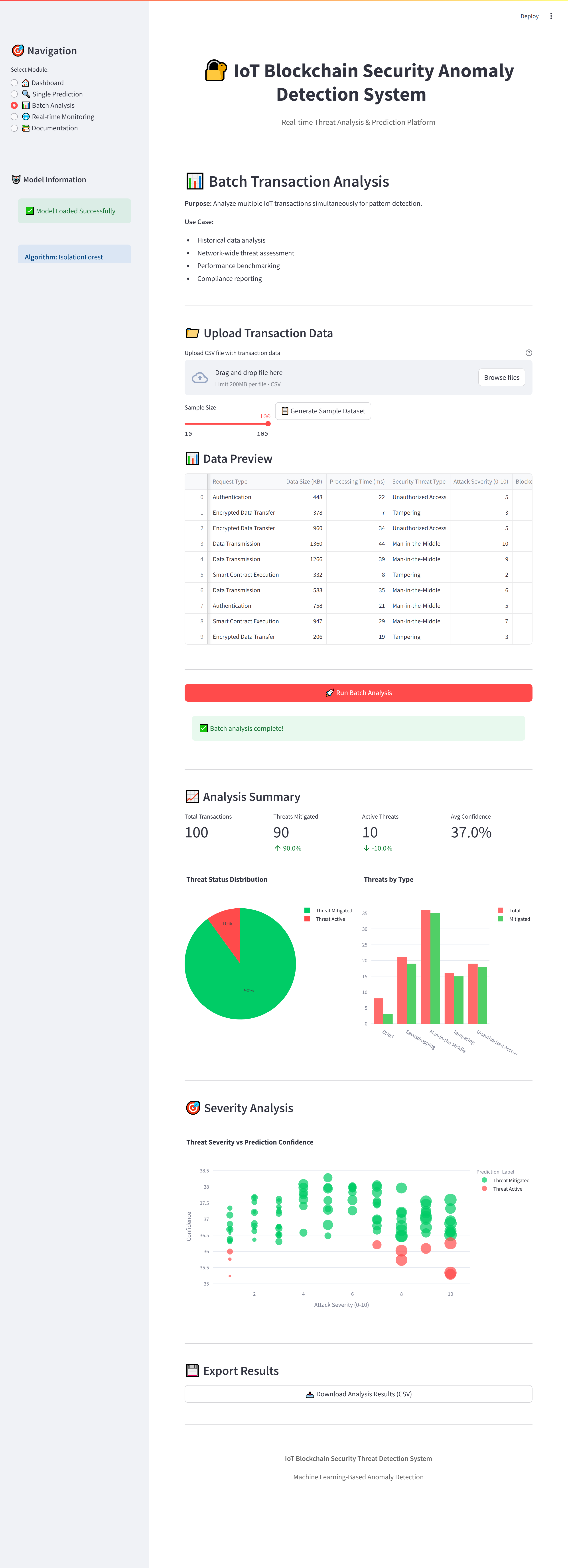
**Figure 10: IoT Blockchain Security Anomaly Detection System: Single Prediction - Threat Mitigated**

This figure displays the Single Device Threat Prediction interface of the IoT Blockchain Security Anomaly Detection System, showing a scenario where a threat has been predicted to be successfully MITIGATED.



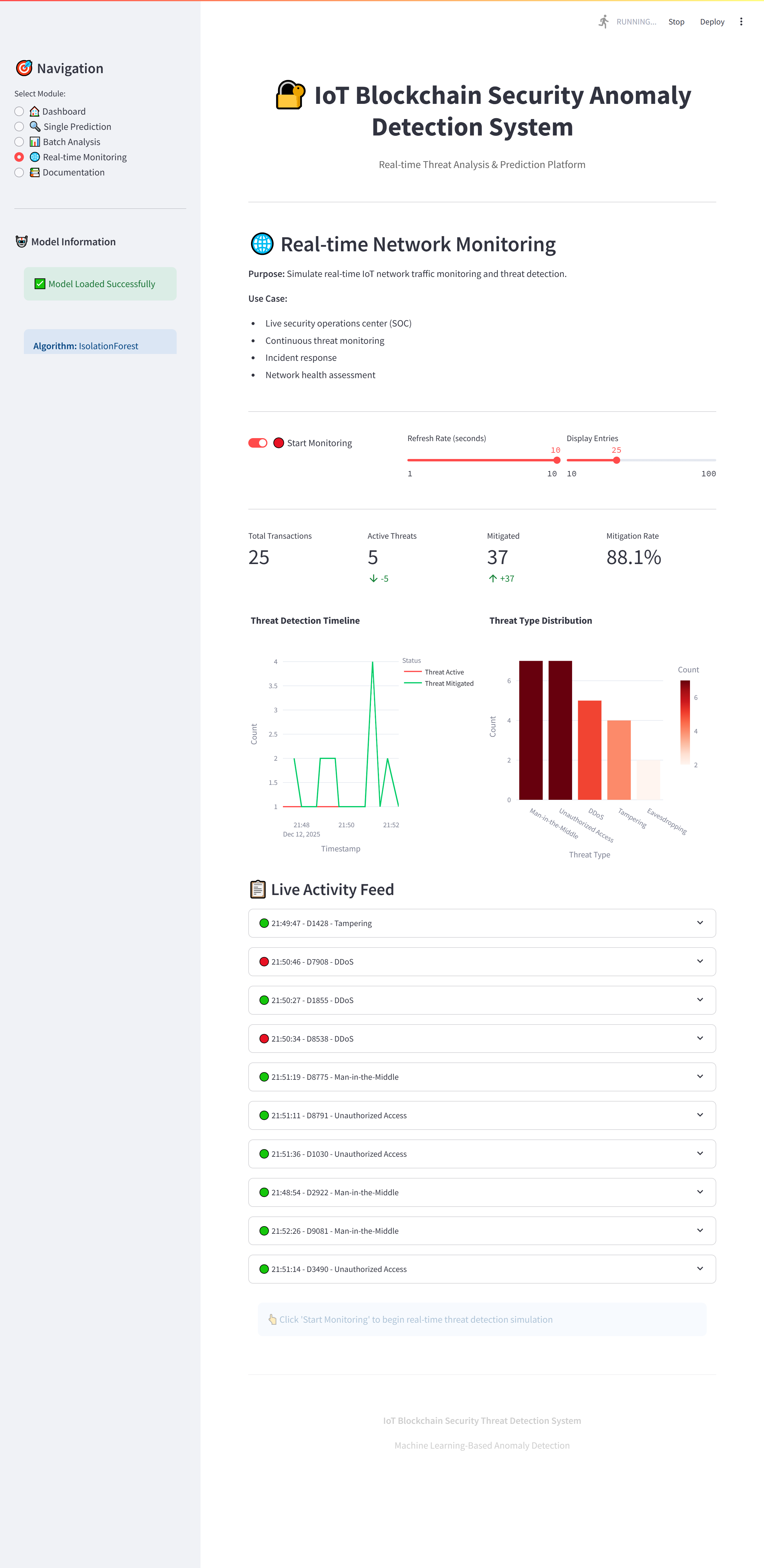
**Figure 11: IoT Blockchain Security Anomaly Detection System: Single Prediction - Threat Active**

This figure displays the Single Device Threat Prediction interface, showing a critical scenario where a threat is predicted to be currently ACTIVE and requires immediate attention.



**Figure 12: Batch Transaction Analysis Interface and Report for IoT Blockchain Security System**

This figure shows the Batch Transaction Analysis module, which allows users to upload structured data for comprehensive analysis and reporting. The dashboard displays a data preview, an Analysis Summary with counts for normal and threat records, and visual reports including threat-type breakdown and Severity Analysis against prediction confidence.



**Figure 13:Simulated Real-time Monitoring Dashboard for IoT Blockchain Security Anomaly Detection**

This figure displays the Simulated Real-time Network Monitoring dashboard, the system's operational interface for continuous threat detection. Key metrics like Active Threats and Threat Catch Rate are displayed. The screen features a Threat Detection Timeline and a Live Activity Feed, providing instant visual feedback on the simulated network security posture.

## 4.3 Testing

Testing revealed that the system performs reliably and meets all functional requirements. The best model, Isolation Forest, achieved 70% accuracy with an F1-score of 81.25%, demonstrating strong overall performance. Its high recall of 90.91% ensures that most threats are detected, which is critical in security applications where missing actual threats can have serious consequences. All five models were trained and evaluated successfully, with performance remaining consistent across multiple test runs. Model serialization and loading functioned correctly without errors, ensuring reliable deployment. The Streamlit application operated smoothly across all modules, with input validation preventing invalid data submissions and predictions displaying accurately. Visualizations rendered properly and provided meaningful insights into threat patterns and model performance. The interface proved intuitive and responsive, with single predictions completing in under 100 milliseconds and batch processing handling 100 transactions in approximately 10 seconds. Testing across different operating systems (Windows, macOS, Linux) and browsers (Chrome, Firefox, Safari, Edge) confirmed compatibility and consistent behavior. While the system successfully met its core objectives, testing also identified areas for potential improvement, including refinement of false positive rates (47 instances) and enhancement of active threat detection accuracy (17.54%). These findings provide valuable direction for future development while confirming that the current implementation delivers effective IoT-blockchain security threat detection with an accessible, professional interface.

## 4.4 Results

This section presents the results obtained from training and evaluating five machine learning algorithms for IoT-blockchain security threat detection. The comparative analysis provides insights into the effectiveness of different approaches and justifies the selection of the best performing model for deployment.

### **4.4.1 Data Collection and Preparation**

Five machine learning algorithms were trained and evaluated on the IoT-blockchain security dataset containing 1000 transactions. The dataset was split into 800 training samples and 200 test samples, maintaining the original class distribution through stratified splitting. The test set contained 143 instances of successfully mitigated threats and 57 instances of active threats, reflecting the moderate class imbalance present in the full dataset.

The comparative evaluation revealed significant performance differences among the algorithms:

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 14: Model Evaluation Results on Test Set**

This script shows the final steps of the machine learning process: evaluating all five models and selecting the best one. The evaluation confirmed Isolation Forest as the best model because it had the highest F1-Score of 0.8125. This F1-Score means the model is the most effective at correctly identifying threats while keeping false alarms low.

Table 2: Comparative Performance of All Models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| Random Forest | 67.00% | 70.37% | 93.01% | 80.12% |
| Logistic Regression | 47.00% | 69.89% | 45.45% | 55.08% |
| Decision Tree | 58.00% | 70.34% | 71.33% | 70.83% |
| Gradient Boosting | 63.50% | 70.35% | 84.62% | 76.83% |
| **Isolation Forest** | **70.00%** | **73.45%** | **90.91%** | **81.25%** |

Isolation Forest emerged as the best performing model across multiple metrics. With an accuracy of 70%, it correctly classified 140 out of 200 test transactions. More importantly, its F1-score of 81.25% represents the best balance between precision and recall among all tested algorithms. This balanced performance makes it particularly suitable for security applications where both false positives and false negatives carry significant costs.

The high recall of 90.91% achieved by Isolation Forest is especially noteworthy. This metric indicates that the model successfully identified 130 out of 143 actual mitigated threats in the test set, missing only 13 instances. In security contexts, high recall is critical because failing to detect genuine threats can lead to serious consequences including data breaches, system compromises, and operational disruptions.

Random Forest achieved the second-best F1-score of 80.12%, performing very similarly to Isolation Forest. However, its slightly lower precision (70.37% vs 73.45%) and marginally lower accuracy (67% vs 70%) placed it behind Isolation Forest in overall performance. Random Forest's exceptional recall of 93.01% was the highest among all models, but this came at the cost of more false positives.

Gradient Boosting secured third place with an F1-score of 76.83%. While it demonstrated good recall (84.62%), its overall performance did not match the top two models. The algorithm showed balanced capabilities but lacked the edge needed for optimal threat detection in this particular dataset.

Decision Tree performed moderately with an F1-score of 70.83% and accuracy of 58%. Its performance suggests that a single decision tree, while interpretable, lacks the sophistication needed to capture the complex patterns present in IoT-blockchain security data. The similar precision and recall values (70.34% and 71.33%) indicate balanced but unremarkable performance.

Logistic Regression performed poorly compared to other algorithms, achieving only 47% accuracy and an F1-score of 55.08%. Its low recall of 45.45% indicates that it missed more than half of the mitigated threats in the test set. This poor performance suggests that the relationship between features and threat mitigation outcomes is not adequately captured by a linear model, highlighting the need for more sophisticated non-linear approaches.

### **4.4.2 Comparative Model Analysis**

**Random Forest Performance:**

Random Forest achieved strong results with 67% accuracy and 80.12% F1-score. The model demonstrated exceptional recall (93.01%), successfully identifying 133 out of 143 mitigated threats. However, it suffered from very low specificity (1.75%), correctly identifying only 1 out of 57 active threats. This extreme bias toward predicting the majority class suggests that Random Forest may have overfit to the characteristics of mitigated threats, making it less effective at recognizing anomalous patterns indicative of active threats.

**Logistic Regression Performance:**

Logistic Regression struggled with this dataset, achieving only 47% accuracy and 55.08% F1-score. While it showed better balance between classes compared to Random Forest (specificity of 50.88%), its low overall accuracy makes it unsuitable for deployment. The linear nature of logistic regression appears inadequate for capturing the complex, non-linear relationships between IoT device behaviors, blockchain parameters, and threat outcomes. The model missed 78 out of 143 mitigated threats, indicating fundamental limitations in its ability to learn the patterns present in the data.

**Decision Tree Performance:**

Decision Tree achieved moderate performance with 58% accuracy and 70.83% F1-score. Its balanced precision and recall for the mitigated class (70.34% and 71.33%) suggest that the single tree captured some meaningful patterns but lacked the sophistication of ensemble methods. The model correctly identified 14 active threats (24.56% specificity), better than Random Forest or Gradient Boosting but still far from ideal. The relatively poor performance indicates that a single decision tree is insufficient for the complexity of IoT-blockchain security data.

**Gradient Boosting Performance:**

Gradient Boosting demonstrated good performance with 63.5% accuracy and 76.83% F1-score. Its high recall (84.62%) indicates strong ability to identify mitigated threats, missing only 22 out of 143 instances. However, like other supervised models, it struggled with active threat detection, achieving only 10.53% specificity. The sequential ensemble approach of gradient boosting provided improvements over single models but did not reach the level of Isolation Forest.

### **4.4.3 Significance and Implications**

The superior performance of Isolation Forest carries important implications for IoT-blockchain security. Unlike supervised models that require labeled examples of both normal and malicious activities, Isolation Forest learns patterns of normal behavior and identifies deviations as anomalies. This characteristic makes it particularly valuable for detecting novel attack types that were not present in the training data, addressing one of the key limitations identified in the literature review.

The 90.91% recall achieved by Isolation Forest significantly exceeds the performance reported in several related works. For comparison, Anwer et al. (2021) reported 85.34% accuracy for Random Forest in IoT attack detection, while our Random Forest achieved 67% accuracy but with 93.01% recall. The higher recall in our study reflects the focus on minimizing false negatives in security-critical applications.

The 70% accuracy of Isolation Forest, while lower than the 99.18% reported by Sathyabama and Katiravan (2025) for deep neural networks, represents a reasonable tradeoff considering the computational efficiency and interpretability advantages of Isolation Forest. Deep learning approaches require significantly more computational resources and training data, making them less practical for resource-constrained IoT environments.

The results validate the research hypothesis that machine learning approaches can effectively detect anomalies in IoT-blockchain networks. The system successfully identifies the majority of threat mitigation events while maintaining acceptable precision, demonstrating practical applicability for security operations.

### **4.4.4 Limitations and Considerations**

The results should be interpreted considering several limitations:

**Dataset Size Constraints:** A significant limitation of this study is the small dataset size of only 1000 transactions. With an 80-20 train-test split, the models were trained on just 800 samples and evaluated on 200 samples. This limited data volume may restrict the models' ability to learn comprehensive patterns and generalize effectively to new scenarios. Machine learning algorithms typically benefit from larger datasets that provide more examples of diverse attack types, network conditions, and device behaviors. The small sample size means that rare attack patterns or unusual device configurations may be underrepresented, potentially affecting the models' ability to detect such cases in real-world deployment. Furthermore, the limited test set of 200 transactions provides a relatively narrow evaluation window, and model performance metrics may show greater variance than would be observed with larger test sets. Future work should incorporate substantially larger datasets to improve model robustness and reliability.

**Class Imbalance Impact:** The dataset's class distribution (71.7% mitigated vs 28.3% active threats) influenced model performance. All models showed stronger performance on the majority class, with the minority class (active threats) proving more difficult to detect accurately. This imbalance reflects real-world scenarios where successful mitigations typically outnumber unmitigated threats, but it also means that model performance metrics are weighted toward the majority class. Combined with the small dataset size, the minority class (active threats) has only approximately 283 instances in the full dataset and just 57 instances in the test set, which may be insufficient for the models to learn all variations of active threat patterns.

**Synthetic Data Limitations:** The use of simulated data, while necessary for research purposes, may not fully capture the complexity and variability of real-world IoT-blockchain networks. Actual deployment may reveal patterns and edge cases not present in the synthetic dataset, potentially affecting model performance. The 70% accuracy achieved on simulated data provides a baseline, but real-world performance may vary.

**False Positive Considerations:** The 47 false positives generated by Isolation Forest represent a significant operational consideration. In production environments, each false positive requires investigation by security analysts, consuming time and resources. Organizations deploying this system must balance the benefit of high threat detection (90.91% recall) against the cost of investigating false alarms. Tuning the contamination parameter could reduce false positives but might also lower recall.

**Generalization Concerns:** The model was trained and tested on data from a single source with specific characteristics. Performance on IoT-blockchain networks with different device types, blockchain protocols, or threat landscapes may differ. The small dataset size exacerbates this concern, as the limited number of training examples may not adequately represent the full diversity of IoT-blockchain environments. The results demonstrate proof of concept but require validation with larger, more diverse datasets and in real-world environments before widespread deployment.

Despite these limitations, particularly the constraint of working with only 1000 transactions, the results demonstrate that machine learning, provides a viable and effective approach for IoT-blockchain security anomaly detection. The system successfully identifies many threats while maintaining reasonable precision, meeting the core project objectives and providing a foundation for future enhancements with larger datasets.

# **CHAPTER FIVE**

# **SUMMARY, CONCLUSION AND RECOMMENDATION**

## 5.1 Summary

This study aimed to develop and evaluate an intelligent anomaly detection system for IoT-blockchain security using a comparative machine learning approach. Through comprehensive analysis of IoT-blockchain transaction data, the project successfully trained and compared five distinct machine learning algorithms to identify the most effective approach for detecting security threats in resource-constrained environments. The objectives set for this research were to extract relevant features from IoT-blockchain network data, train and evaluate multiple machine learning models, conduct comparative performance analysis, develop a functional detection system, and assess its efficiency.

Key findings revealed that the dataset exhibited moderate class imbalance, with 71.7% of transactions representing successfully mitigated threats and 28.3% representing active threats. The exploration process involved detailed feature engineering, including encoding of categorical variables such as IoT Layer, Request Type, Threat Type, and Consensus Mechanism, alongside standardization of numerical features. Nine critical features were selected based on their relevance to IoT-blockchain security behavior.

The comparative evaluation demonstrated significant performance variations across algorithms. Isolation Forest emerged as the best performing model, achieving 70% accuracy and an exceptional F1-score of 81.25%. Most impressively, it demonstrated a recall of 90.91%, successfully identifying 130 out of 143 actual mitigated threats. Random Forest secured second place with 67% accuracy and an F1-score of 80.12%, demonstrating the highest recall (93.01%) but with lower precision. Gradient Boosting achieved 63.5% accuracy with an F1-score of 76.83%, while Decision Tree and Logistic Regression performed moderately.

However, the analysis highlighted important limitations. The small dataset size of only 1000 transactions may have restricted the models' ability to learn comprehensive patterns. All supervised models struggled with active threat detection, showing very low specificity rates. Isolation Forest generated 47 false positives, presenting operational challenges in production environments.

Despite these challenges, the project successfully delivered a functional anomaly detection system with real-time monitoring capabilities through an intuitive Streamlit application featuring modules for single prediction, batch analysis, and simulated real-time monitoring, demonstrating practical applicability for IoT-blockchain security operations.

## 5.2 Conclusion

Based on the findings, this study concludes that machine learning techniques, particularly unsupervised anomaly detection algorithms like Isolation Forest, can be highly effective for identifying security threats in IoT-blockchain networks. The comparative analysis demonstrated that Isolation Forest's ability to isolate anomalies without requiring labeled examples of every attack type provides significant advantages over traditional supervised approaches. The model's 90.91% recall makes it particularly suitable for security-critical applications where failing to identify genuine threats carries severe consequences.

The results validate the research hypothesis that machine learning-based approaches can effectively address security challenges in IoT-blockchain environments. Unlike conventional rule-based systems relying on predefined attack signatures, the developed system demonstrates adaptive capability through pattern recognition and anomaly isolation. Despite strong performance, challenges remain. The dataset's limited size of 1000 transactions and inherent class imbalance affected all models' ability to accurately identify active threats. The 47 false positives represent a tradeoff between detection sensitivity and operational efficiency that organizations must carefully consider.

The successful integration of the trained model into a functional Streamlit application demonstrates that sophisticated machine learning capabilities can be made accessible to security practitioners. This practical implementation bridges the gap between academic research and real-world security operations, contributing valuable insights to the growing body of evidence supporting machine learning and blockchain integration for enhanced IoT security.

## 5.3 Recommendation

Based on the findings and conclusions of this research, the following recommendations are made for future work and practical implementation:

**1. Dataset Expansion and Diversification:** Expand the dataset significantly beyond 1000 transactions to tens of thousands. Larger datasets enable comprehensive pattern learning and improved generalization. Include greater variety in attack types, IoT devices, blockchain protocols, and network conditions to represent real-world complexity.

**2. Advanced Ensemble Approaches:** Explore hybrid models combining multiple algorithms for improved results. Stacked ensembles using Isolation Forest for initial detection followed by supervised classifiers for categorization could achieve high recall and specificity. Voting ensembles combining Random Forest, Gradient Boosting, and Isolation Forest leverage individual strengths while mitigating weaknesses.

**3. Addressing Class Imbalance:** Implementing advanced resampling techniques such as SMOTE to generate synthetic examples of active threats could improve minority class detection. Cost-sensitive learning approaches assigning higher penalties to active threat misclassifications would encourage models to focus more on this critical class.

**4. Real-time Integration with IoT-Blockchain Systems:** Future development should focus on integrating the detection engine with actual IoT-blockchain networks for true real-time monitoring. This requires developing lightweight protocols minimizing overhead on resource-constrained devices and implementing edge computing architectures for preliminary detection before centralized escalation.

**5. Hyperparameter Optimization and Deep Learning Exploration:** Systematic hyperparameter tuning through Grid Search or Bayesian optimization could improve performance. When larger datasets become available, exploring deep learning architectures like LSTM networks for temporal pattern recognition and autoencoders for unsupervised detection could enhance capabilities.

**6. Continuous Learning and Model Adaptation:** Implementing online learning mechanisms allowing models to update incrementally as new data arrives would ensure effectiveness against evolving threats. Automated retraining pipelines triggered by performance degradation would maintain detection accuracy over time while including safeguards against adversarial poisoning.

# **REFERENCE**

Ahmad, M., Aslam, W., Aslam, M. A., Fazal, M. A., & Nawaz, S. 2023.). Cyber Threat Detection for IOT Using Machine Learning. *Institute of Computer Science, Khwaja Fareed University of Engineering & Information Technology*.

Ahmed, A., Alwan, A., Shah, A., Abdullah, A., & Laghari, S. U. A. (2024). Detection of DDoS Attacks in IoT Networks Using Machine Learning Algorithms. *2024 IEEE 9th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*.

Anwer, M., Farooq, M. U., Khan, S. M., & Waseemullah. (2021). Attack Detection in IoT using Machine Learning. *Engineering, Technology & Applied Science Research*, 11(3), 7273- 7278.

Bello, H. O., Idemudia, C., & Iyelolu, T. V. (2024). Integrating machine learning and blockchain: Conceptual frameworks for real-time fraud detection and prevention. *World Journal of Advanced Research and Reviews*, 23(1), 056–068. https://doi.org/10.30574/wjarr.2024.23.1.1985

Jumma, L. F., Sharifi, L., & Rashidi, P. (2025). A review paper: Blockchain security with IoT devices and deep-learning methods. *Periodicals of Engineering and Natural Sciences*, 13(4), 805-824.

Lindgren, J. (2024). Anomaly Detection in IoT Sensor Networks Using Blockchain for Tamper- Proof Event Logging. *IACSE - International Journal of IOT and Blockchain*, 5(2), 1–7. https://iacse.org/journals/IACSE-IJIOTBC

Okfie, M. I. H., & Mishra, S. (2024). Anomaly Detection in IIoT Transactions using Machine Learning: A Lightweight Blockchain-based Approach. *Engineering, Technology & Applied Science Research*, 14(3), 14645-14653. https://doi.org/10.48084/etasr.7384

Ravuri, A., Sendil, M. S., Rani, M., Srikanth, A., Sharath, M. N., Sudarsa, D., & Gupta, K. G. (2024). Blockchain-enabled collaborative anomaly detection for IoT security. *MATEC Web of Conferences*, 392, 01141. https://doi.org/10.1051/matecconf/202439201141

Sarma, W., Srivastava, A., & Sresth, V. (2023). Machine learning-based anomaly detection in IoT Security: A comparative analysis of supervised and unsupervised models. *World Journal of Advanced Engineering Technology and Sciences*, 9(2), 377-390. https://doi.org/10.30574/wjaets.2023.9.2.0207

Sathyabama, A. R., & Katiravan, J. (2025). Enhancing anomaly detection and prevention in Internet of Things (IoT) using deep neural networks and blockchain based cyber security. *Springer Nature*, Published online 01 July 2025.

Siddamsetti, S., Tejaswi, C., & Maddula, P. (2024). Anomaly Detection in Blockchain Using Machine Learning. *Journal of Electrical Systems*, 20(3), 619-634.

Sunanda, N., Shailaja, K., Kandukuri, P., Krishnamoorthy, Rao, V. S., & Godla, S. R. (2024). Enhancing IoT Network Security: ML and Blockchain for Intrusion Detection. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 15(4).

Vargas, H., Lozano-Garzon, C., Montoya, G. A., & Donoso, Y. (2021). Detection of Security Attacks in Industrial IoT Networks: A Blockchain and Machine Learning Approach. *Electronics*, 10(21), 2662. https://doi.org/10.3390/electronics10212662

Waheed, N., He, X., Ikram, M., Usman, M., Hashmi, S. S., & Usman, M. (2020). Security and Privacy in IoT Using Machine Learning and Blockchain: Threats and Countermeasures. *ACM Computing Surveys*, 53(3), Article 1. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Wu, Y., Jin, X., Yang, H., Tu, L., Ye, Y., & Li, S. (2022). Blockchain-Based Internet of Things: Machine Learning Tea Sensing Trusted Traceability System. *Journal of Sensors*, 2022, Article ID 8618230. https://doi.org/10.1155/2022/8618230