ASHESI UNIVERSITY

SensePay: EMBEDDED IMU INTELLIGENCE AND MACHINE LEARNING FOR REAL-TIME RELAY ATTACK PREVENTION IN CONTACTLESS PAYMENTS

CAPSTONE PROJECT

B.Sc. Computer Engineering

Maxwell Bosiako Antwi

2025

Supervisor: Robert Sowah

ASHESI UNIVERSITY

SensePay: EMBEDDED IMU INTELLIGENCE AND MACHINE LEARNING FOR RELAY ATTACK PREVENTION IN CONTACTLESS PAYMENTS

CAPSTONE PROJECT

Capstone Project submitted to the Department of Engineering, Ashesi University in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Engineering.

Maxwell Bosiako Antwi

2025

Supervisor: Robert Sowah

DECLARATION

I hereby declare that this capstone is the result of my original work and that no part has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……

Candidate's Name:

Maxwell Bosiako Antwi

Date: 28th/04/2025

I hereby declare that the preparation and presentation of this capstone were supervised following the guidelines for the supervision of the capstone laid down by Ashesi University.

Supervisor's Signature:

……..

Supervisor's Name:

Dr. Robert Sowah

Date: 28/04/2025

ACKNOWLEDGEMENTS

My sincerest gratitude goes to my project supervisor, **Dr. Robert Sowah**, for his invaluable guidance, patience, and unwavering support throughout this capstone journey. His expertise and insightful

feedback have been instrumental in shaping the direction of this project and overcoming technical challenges.

Thanks really are to the classmates for their friendship and support, be it shared components, brainstorming sessions, or late-night debugging. In fact, it was the group work that made this undertaking seem less daunting and much more enriching.

I also wish to recognize Gabriel Odu Jnr at the Ashesi Support Center for his keen interest and invaluable assistance particularly during the testing phase. His support ensured the practical success of this project.

Finally, I acknowledge the contributions of all individuals, including lab assistants, who offered indirect but critical support. This project would not have been possible without their collective assistance.

ABSTRACT

As contactless payment systems continue to expand, they remain exposed to relay attacks that exploit the absence of physical proximity verification. This research proposes **SensePay**, a lightweight, orientation-based fraud detection framework that enhances NFC payment security by embedding motion sensing and machine learning into the transaction process. A 6-axis IMU captures real-time device orientation (pitch, roll) and transaction timing, which are analyzed by a trained One-Class SVM model running locally on an ESP32 microcontroller. Experimental evaluation demonstrated that the system achieves 96.7% classification accuracy and maintains sub-300 ms transaction latency, outperforming traditional statistical threshold methods. Both delay-based and orientation-based relay attacks were successfully detected, confirming the approach's robustness. These results highlight the practicality of integrating real-time, behavioral anomaly detection into existing payment terminals without sacrificing speed, power efficiency, or user convenience, paving the way for stronger and scalable contactless payment security.

Table of Contents

Chapter 1: Introduction

The rise of contactless payment technologies, powered by Near Field Communication (NFC) and Radio Frequency Identification (RFID), has revolutionized the way consumers conduct transactions. Over the past decade, the global economy has witnessed a dramatic shift toward cashless payment systems, with projections estimating a 460% increase in contactless transactions by 2029, reaching a value of approximately $15.7 trillion [1]. This widespread adoption is largely driven by the convenience, speed, and user-friendliness offered by contactless platforms.

However, alongside these advancements, critical security vulnerabilities have emerged. One of the most prominent threats is the **relay attack**, where malicious actors intercept and forward communication between a contactless card or mobile device and a payment terminal without modifying the transmitted data. By exploiting the lack of stringent physical proximity verification, attackers can complete unauthorized transactions while the victim remains unaware.

Although existing security protocols, such as those outlined in the EMV (EuroPay, Mastercard, Visa) standards, have introduced measures like the Relay Resistance Protocol (RRP) [4,5], these solutions remain limited. In particular, RRP focuses on securing communication channels but does not sufficiently address relay attacks that leverage minimal physical delays and optimized timing. Thus, as the reliance on contactless transactions grows, there is a critical need for more robust, context-aware mechanisms to ensure both the authenticity of the payment device and the proximity of the transaction.

This research addresses this gap by proposing a lightweight, real-time fraud detection framework that integrates behavioral motion sensing with embedded machine learning, aiming to strengthen security in NFC-based payment systems.

1.1 Background

Relay attacks have emerged as a critical threat to the security of cashless payment systems, exploiting inherent vulnerabilities in contactless communication protocols. In a typical relay attack, two colluding entities establish a covert communication channel between the victim's payment device and the legitimate point-of-sale (POS) terminal. As illustrated in ***Figure 1.1***, the mole remains in close proximity to the victim, capturing NFC signals, while the proxy forwards the communication to the terminal, effectively impersonating the victim's card without physical presence [2].

Such attacks deceive the terminal into believing the card is nearby, enabling unauthorized transactions to occur seamlessly. Experimental studies have demonstrated the severity of this threat; notably,

researchers proved that relay attacks remain effective even with mole and proxy devices separated by distances of up to 50 meters [3]. Furthermore, successful demonstrations using commercially available hardware as early as 2019 highlighted critical flaws in widely adopted systems [7,11].

Alarmingly, real-world tests revealed that popular contactless platforms such as Visa payWave and Mastercard PayPass are susceptible to relay attacks, despite compliance with existing security protocols [8,13]. These findings raise significant concerns regarding the adequacy of current defenses and reveal the urgent need for more context-aware security mechanisms that verify not only credentials but also genuine device proximity.

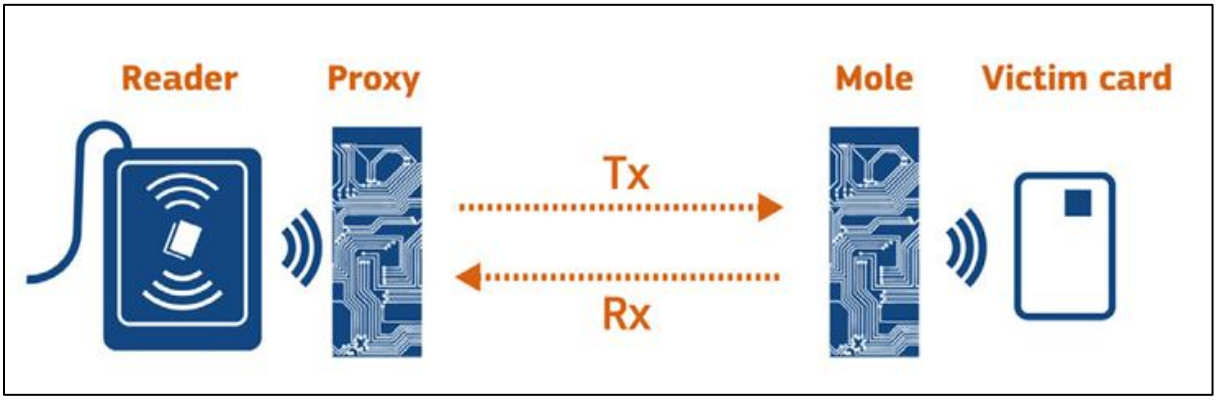


Fig. 1.1 Relay attacks against contactless card

1.2 Project Objectives

This project aims to develop a secure and practical technique for mitigating relay attacks in contactless payment systems. While cashless transactions have transformed modern commerce, the inherent lack of robust physical proximity verification exposes users to significant risks, particularly through relay attacks. This research focuses specifically on addressing these vulnerabilities by identifying, implementing, and evaluating effective techniques that enhance transaction security without introducing substantial complexity and latency.

The key objectives of the project are as follows:

- **Explore and assess potential solutions** for mitigating relay attacks, including motion-based and timing-based approaches.
- **Design and develop a functional prototype** that integrates the selected methods, ensuring compatibility with existing contactless payment infrastructure.

- **Test and evaluate the prototype** under simulated attack scenarios, including both delay-based and orientation-based relay attacks.
- **Analyze system performance** and provide actionable recommendations for integrating the proposed solution into existing payment systems with minimal disruption.

The project initially began with an exploratory investigation into the security limitations of NFC-based systems. Based on the findings, a structured engineering design approach was adopted, encompassing problem definition, requirement specification, solution development, and system-level evaluation. This methodology ensured that the proposed orientation-based fraud detection system was developed with a strong foundation in technical feasibility, performance-effectiveness, and real-world applicability.

Chapter 2: Literature Review and Related Work

This chapter will review existing research on relay attacks and contactless payment security. It will further dive into existing work related to this project.

2.1 Literature Review

The current body of knowledge on relay attacks, contactless payment vulnerabilities, and security protocols is studied in this section.

2.1.1 Attacks on Contactless Payment Systems

Near Field Communication (NFC) and Radio Frequency Identification (RFID) technologies have improved contactless systems, especially in payments and access control. However, both are highly susceptible to relay attacks, a form of man-in-the-middle attack that exploits the proximity-based security of these technologies. Despite the advantages offered by NFC and RFID in terms of convenience, their vulnerabilities have been the subject of extensive research. This section reviews existing studies that detail the execution of relay attacks and explores why these attacks are particularly challenging to prevent.

2.1.1.1 How Relay Attacks are Executed

Relay attacks establish a seamless communication link between two geographically separated locations. As introduced in *Section 1.1,* relay attacks allow the proxy token to masquerade as the legitimate token to the reader, thereby enabling unauthorized transactions or access [14].

The **Mafia Fraud** scenario is a classic analogy often used to explain relay attacks. Here, two chess grandmasters are unknowingly made to play against each other by an intermediary posing as a novice. The intermediary forwards the moves of one grandmaster to the other, making both believe they are playing the intermediary when, in fact, they are competing against each other. Similarly, relay attacks exploit the fact that the communication between the NFC or RFID reader and token happens invisibly through electromagnetic waves, allowing attackers to relay messages without either party realizing they are being deceived [15]. In the context of NFC, mobile phones equipped with NFC interfaces serve as both the mole-reader and proxy token. Attackers leverage the programmable nature of NFC technology by installing custom applications that facilitate communication relays between two phones. This makes the attack highly practical, eliminating the need for custom-built hardware [16]. Such

implementations have demonstrated the feasibility of launching real-world relay attacks against payment systems and other NFC-enabled services [15].

[16] demonstrate the practical feasibility of relay attacks using off-the-shelf smartphones running relay software like *NFCProxy and NFCGates*. All of their experiments to relay transactions across different NFC payment devices succeeded, showing that the attack works consistently and without requiring specialized hardware. By using communication technologies such as Wi-Fi or Bluetooth, attackers can extend the range of NFC signals beyond a few centimeters, undermining the short-range communication security inherent to NFC [16]. These attacks were tested on ISO 14443 proximity cards, where cryptographic protections were found insufficient to prevent them, as the communication itself was not altered, only forwarded [14].

The effectiveness of these attacks lies in their invisibility to the user and the system. Because the relay only forwards original messages between the token and the reader, it bypasses cryptographic protections such as encryption and challenge-response authentication. As shown in ***Figure 2.1***, the attacker acts as a middleman, relaying all communication without modifying it. This makes the attack extremely difficult to detect using traditional security mechanisms [16].

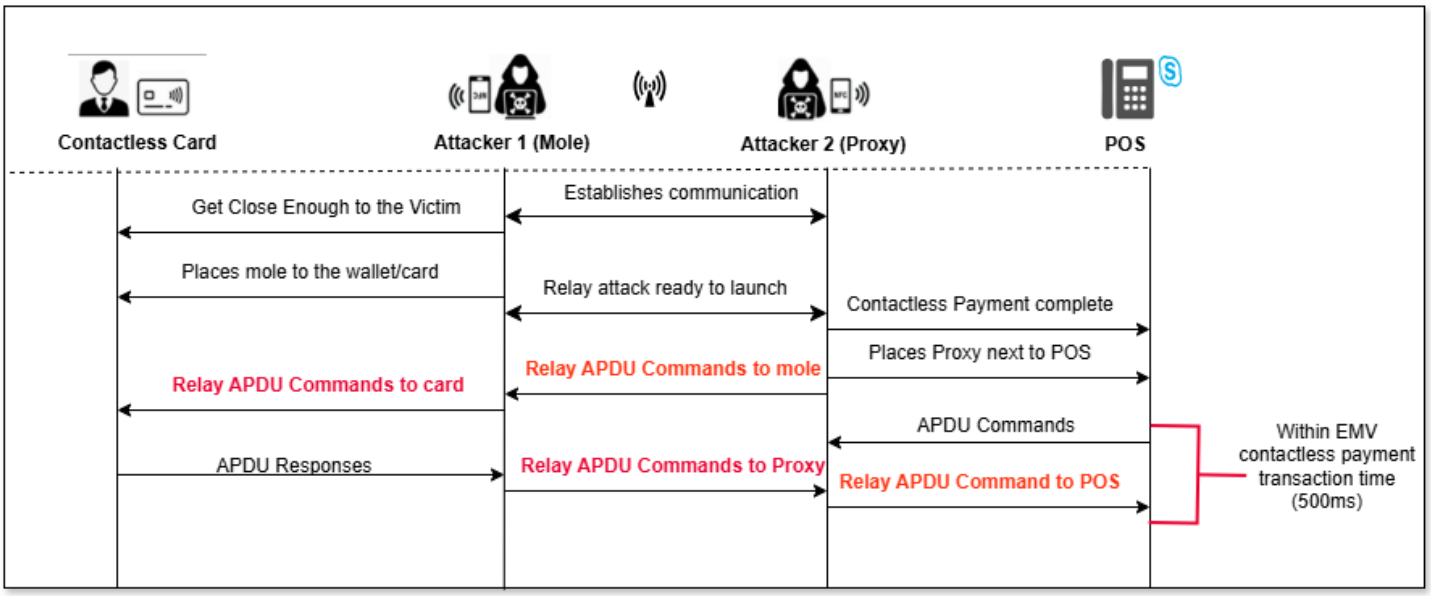


Fig 2.1. Data flow diagram of how relay attacks work

2.1.1.2 Challenges in Preventing Relay Attacks

Preventing relay attacks is challenging because they exploit proximity spoofing, deceiving the system into believing that the token is physically near the reader. NFC and RFID systems assume that proximity provides inherent security, making them particularly vulnerable when signals are relayed over longer distances [14]. Existing cryptographic protocols, such as those defined in ISO/IEC 14443, ensure data integrity but do not verify the physical location of the communicating devices [15]. One widely discussed defense mechanism is distance-bounding protocols, which estimate the distance between the card and the reader by measuring signals' time-of-flight (ToF). However, these protocols are not foolproof. [14] highlights that timing vulnerabilities arise when attackers introduce slight undetected delays, rendering the protocol ineffective for preventing relay attacks.

2.1.2 Current Solutions to Relay Attacks

a. Encryption in NFC Payments

Encryption ensures the confidentiality and integrity of data exchanged during contactless transactions. In NFC systems, AES (Advanced Encryption Standard) and RSA encryption algorithms are often employed to secure data transmitted between a token (such as a payment card) and a reader [16]. However, encryption alone is insufficient to prevent relay attacks. Since a relay attack only forwards the communication between the card and the reader without modifying it, the original encrypted data remains intact, allowing the transaction to proceed successfully.

As outlined by [18], relay attacks exploit the assumption that both parties in a transaction are nearby, bypassing encryption mechanisms by simply relaying encrypted packets over long distances [18][20]. Furthermore, NFC encryption protocols are vulnerable to man-in-the-middle (MITM) attacks, where attackers intercept and relay encrypted messages, making proximity verification critical for security.

b. EMV Protocol and Its Limitations

The EMV protocol (Europay, Mastercard, and Visa) is the global standard for contactless payment security, employing dynamic data authentication (DDA) and cryptographic keys to secure transactions. In EMV-compliant systems, each transaction generates a unique cryptogram, which prevents replay attacks. However, research highlights that EMV protocols are not designed to detect relay attacks effectively. As discussed by [21], the primary flaw in the EMV standard is its reliance on communication success to validate proximity. Attackers can relay communication between the token and reader, tricking the system into completing the transaction despite the physical separation between

the two parties [14][21]. Additionally, the cryptographic operations in EMV introduce latency and faster transactions, which are critical in retail environments and cannot afford the delays required for more robust verification protocols [20]. This limitation makes it challenging to adopt distance-bounding protocols within EMV systems, as such methods could slow down the payment process, resulting in poor user experience [16]. These attacks highlight the inability of existing EMV mechanisms to reliably verify the physical proximity of the card and reader.

c. Tokenization and Its Vulnerabilities

Tokenization is widely used in payment systems to replace sensitive payment information with a unique token for each transaction. This token cannot be reused, providing additional protection against replay attacks. However, tokenization has limitations in addressing relay attacks, as the tokens themselves can be relayed without being intercepted or altered [16]. As detailed by [22], tokenized payments still assume that the proximity between devices ensures security. Relay attacks do not abide by this and use techniques to relay valid tokens, unbeknownst to legitimate users, to enable attackers to transact from a remote location. Moreover, if the attackers compromise the relay channel, they can forward requests for tokens and responses in real time, thus making it exceedingly difficult to detect any fraudulent activity during the transaction.

2.1.3 Potential technologies to enhance security of contactless payment

2.1.3.1 NFC with Inertial Measurement Units (IMUs) for Relay Attack Mitigation

Combining Near Field Communication (NFC) with Inertial Measurement Units (IMUs) provides a potent defense for relay attacks against contactless payment systems. IMUs that include accelerometers and gyroscopes detect real-time discrepancies in motion and orientation between an NFC-enabled payment card or device and the terminal, allowing determination of whether the two are actually close to one another and making it much harder for attackers to perform relay attacks.

As outlined by [23], the card and terminal's physical alignment and movement patterns during a legitimate transaction are distinct from those during a relay attack, where attackers cannot precisely replicate the card's real-time motion or alignment. In a genuine payment scenario, the motion of the card and terminal is typically synchronized or stationary. However, during a relay attack, the physical separation of the legitimate card and the attacker's proxy device results in misaligned motion patterns. IMUs can detect these inconsistencies, flag such suspicious transactions, and prevent unauthorized access to the payment system [23].

Challenges of Integrating NFC with IMUs

While NFC-IMU integration offers significant security advantages, it also introduces technical challenges. One of the primary concerns is sensor noise and accuracy. IMUs can be influenced by environmental factors such as magnetic fields and electronic noise, resulting in a distortion of motion and orientation readings. Such distortions can lead to false positives, in which the genuine transaction is rejected as fraud or false negatives, where fraudulent transactions go undetected [23]. Another hurdle would be the cost and technology challenge of embedding IMUs in NFC-enabled devices. The addition of motion sensors will certainly increase the cost of manufacturing payment cards or terminals and would also require compatibility changes in the existing payment infrastructures.

2.1.3.2. Multichannel Protocols for Mitigating Relay Attacks

Multichannel protocols have emerged as an innovative approach to enhancing the security of contactless payment systems by addressing the vulnerabilities exploited during relay attacks. Multichannel protocols, unlike the traditional one-channel communications, make use of multiple independent channels to transmit authentication data from one point to another. An example is combining the NFC with Bluetooth or Wi-Fi. For example, the NFC channel is transmitting crucial authentication data regarding a transaction, while an auxiliary channel, for instance, BLE, relays complementary information. When an attacker relays the NFC data, this gets compared to the BLE communication, and this is how inconsistencies get noticed. The scheme devolves into one better aligned toward thwarting attackers who base their attempts on compromising a single communication channel [24].

Challenges in Using Multichannel Protocols

More channels included in a multichannel scheme, more complex the system develop, requiring additional hardware, software, and algorithms to ensure that the channels interact smoothly. Such complexity will always create latencies that will differ from one implementation to the other but can be quite significant in user experience during transactions. Environmental conditions mostly interfere or affect the quality of the signal on secondary channels such as BLE or Wi-Fi. So, with such developments, reliability issues come into light concerning these secondary channels [24].

2.1.3.3 Machine Learning for Anomaly Detection in Relay Attack Prevention

Machine learning (ML) has emerged as a powerful tool in enhancing the security of contactless payment systems by detecting anomalies that signify potential relay attacks. The models created with

ML are capable of capturing anomalies from human-normal transactional behavior, and such can form the basis of a strong countermeasure.

One promising approach is using **Markov chain models**, which analyze sequential data to detect anomalies in payment transactions. Relay attacks often introduce unusual patterns in the transactional process, such as increased time delays between authentication and payment. The Markov chain model tracks the probability of state transitions during a transaction and flags instances where observed behavior deviates significantly from the learned patterns. For example, in a study focusing on NFC relay attacks, the Markov chain model effectively detected anomalies by analyzing transaction wait time (TWT) data, achieving an accuracy rate of over 90% [25]. The model can identify subtle anomalies that traditional rule-based systems might overlook, offering a higher level of security for contactless payment systems.

Challenges in Implementing ML for Anomaly Detection in Relay Attack Prevention

A key challenge lies in the dependence on high-quality training data. Machine learning models require extensive datasets to learn the patterns of legitimate transactions and identify anomalies indicative of relay attacks. Unfortunately, access to comprehensive and representative datasets is limited. Most financial institutions have proprietary data that is often very proprietary. These data scarcity problems become significant hurdles for building generalized models. For instance, relay attacks occur so infrequently compared to normal transactions that it leads to data imbalance in the model which can misclassify attacks or fail to recognize some subtle signs of fraud. Data privacy regulations in force, such as the General Data Protection Regulation (GDPR), further impose constraints on the collection and storage of transactional data, hampering efforts to collect sufficient training samples. One of the difficulties posed by machine learning is the computational overhead, especially in those scenarios where you have real-time transaction processing.

2.1.3.4 UWB Technology

Ultra-wideband (UWB) transmits data speedily using pulse duration extremely short applying the FCC standard over a frequency band from 3.1 to 10.6 GHz [17]. UWB, unlike narrowband technologies, spreads the signal over a broader range of frequencies so as to have a high data rate and low spectral density, thus excluding spectral density features from interference or jamming [18]. UWB offers a promising defense against relay attacks using precise ToF measurements to confirm proximity between devices. Since UWB systems measure the exact time it takes for a signal to travel, they can identify unusual delays caused by relayed signals, even if those delays are in the microsecond range. For

example, if an attacker tries to relay a signal over Wi-Fi or Bluetooth, the added latency will exceed UWB's precise detection threshold, flagging the communication as fraudulent [19]. This makes it nearly impossible for UWB to deceive through relay attacks, as attackers would need to replicate the signal and the exact timing of the original communication, an exceedingly difficult task [18].

Challenges in UWB Payment Research

One challenge involves the latency introduced by ToF measurements, which may slow down transactions. Fast payment processes are crucial for user satisfaction in retail environments, and any delay can negatively impact the user experience. [18] while UWB offers sub-millisecond accuracy, processing the ToF data in real-time can be computationally intensive, especially when multiple users are involved simultaneously. While prototypes have demonstrated the effectiveness of UWB in preventing relay attacks under controlled conditions, there is limited research on how these systems perform under real-world conditions, such as during peak hours in retail stores where network interference and signal congestion are prevalent [19].

Given the limitations of timing-based protocols and the practical challenges of hardware-intensive solutions like UWB, this work explores the integration of motion sensing (via IMUs) and local anomaly detection for real-time fraud prevention. This approach aims to bridge the gap between computational efficiency, proximity verification, and ease of integration into existing POS systems.

Chapter 3: Systems Requirements and Designs

This chapter outlines the functional and non-functional requirements of the proposed solution to prevent relay attacks. It delves into architectural design, detailing the components and interactions necessary to implement the solution effectively

3.1 System Users

a. Consumers: The primary users of contactless payment systems that are directly affected by relay attacks. Consumers often encounter problems at the vulnerability stage, as their payment cards or devices are the targets of relay attacks.

b. Merchants: Merchants play a dual role as facilitators of contactless payment systems and as stakeholders impacted by fraudulent transactions. During the transaction phase, merchants face issues with relay attacks leading to fraudulent payments that could be challenged either by consumers or by payment organizations.

C. Payment Providers: Banks, credit card networks, or any payment gateway services are grouped as payment providers, and they are responsible for maintaining the entire contactless payment ecosystem's integrity. Payment providers will normally be actively engaged during the stage of solution implementation, where adequate and effective countermeasures against relay attacks are installed.

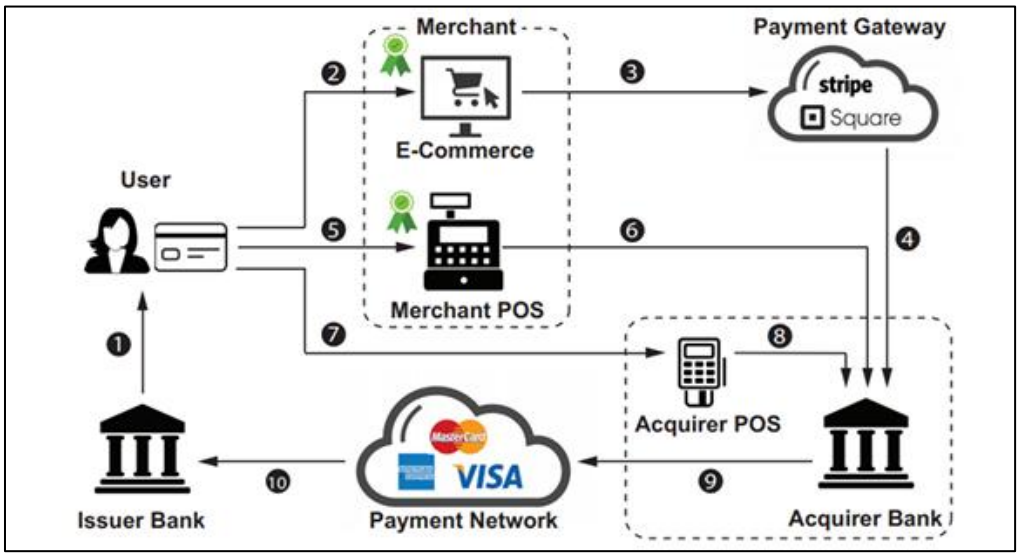


Figure 3.1 Contactless Payment Ecosystem

3.2 Systems Requirements

The system requirements are generally divided into functional and non-functional requirements to ensure the system combats relay attacks while maintaining usability and scalability.

3.2.1 Functional Requirements

- ***Proximity Verification:*** The system must determine if the payment device is physically close to the POS terminal to detect relay attacks.
- ***Anomaly Detection:*** The system must recognize the peculiarities in behavior of transactions or their communications that may indicate the presence of relay attack.
- ***Accuracy:*** The system must have high accuracy, having a very low false positive score.
- ***Real-Time Transaction Monitoring:*** The system must monitor actual transaction data in real time while marking out abnormal situations during the payment transaction.
- ***Low Latency***: The system has to process and validate transactions within 500ms time for contactless payment transactions.

3.2.2 Non-Functional Requirements

- ***Compatibility:*** The system should be compatible with as wide a range of payment devices as possible.
- ***Feedback Mechanism:*** The system will provide clear and immediate feedback to the user and to the merchant of the transaction status, any security issues will also be flagged.
- ***Integration into the Existing Infrastructure of the System:*** The system must easily integrate existing POS terminals and also integrate into EMV Protocols.

3.3 Systems Designs

This section provides an overview of the design selection process, system architecture, detailed design specifications for major components, hardware requirements, and necessary software for implementing the solution. The key idea is to verify the physical proximity and interaction of the payment device with the POS terminal.

3.3.1 Design Selection Process

To determine the most suitable design for proximity verification in contactless payment systems and to mitigate relay attacks, the following design choices were evaluated:

- *IMU + NFC integration*
- *Multi-Channel Protocols*
- *Machine Learning Anomaly Detection*
- *UWB Technology*
- *Distance Bounding Protocols*

Each design was assessed against the system requirements. As explained in chapter 2, these are the different possible designs that can be used to prevent relay attacks.

3.3.1.1 Design Matrix

The design matrix was applied as a systematic decision-making device to assess and compare alternate design proposals according to the established criteria. This provided an objective means of approaching the selection process, assigning weighted scores for each criterion until the chosen design was that which best met the demands of the project.

Designs	*Proximity Verification*	*Anomaly Detection*	*Real-Time Monitoring*	*Feedback Mechanism*	*Random Challenge Mechanism*	*Integration*	*Total*
IMU+NFC	5	4	4	4	5	5	**27** ✔ ✔
Multi-Channel Protocols	3	3	4	4	3	4	**21** ✘ ✘
Machine Learning (ML)	3	5	5	4	2	5	**24** ✘ ✘
UWB Technology	4	4	4	4	3	3	**22** ✘ ✘
Distance Bounding Protocols	4	3	3	3	4	3	**20** ✘ ✘

Scoring Key:

1 = Poor, 3 = Average, 5 = Excellent

3.3.1.2 Design Matrix Scoring Justification

The selection process prioritized components that excel in key functional requirements: ***Proximity Verification, Anomaly Detection, Real-Time Monitoring, and Integration with Existing Infrastructure***.

Criterion	IMU + NFC	Multi-Channel Protocols	ML Alone	UWB Technology	Distance Bounding
Proximity Verification	5 – Uses local motion data to confirm active proximity, making relay attacks harder	3 – Uses cross-channel consistency but lacks physical motion awareness	3 – ML cannot directly confirm spatial proximity without sensor input	4 – Accurate ranging but vulnerable to optimized relay attacks	4 – Distance estimation via timing; vulnerable to fast relays that mimic proximity
Anomaly Detection	4 – Detects abnormal motion or stillness indicative of proxy attacks	3 – May detect high-level anomalies, but lacks fine motion context	5 – Learns complex patterns and deviations from training data	4 – Can support anomaly detection but requires added sensor fusion	3 – Anomaly detection possible but requires heavy timing and cryptographic handling
Real-Time Monitoring	4 – IMU data can be streamed in real time with low latency (~ms-level)	4 – Possible, but less sensitive to rapid orientation changes	5 – ML enables continuous, dynamic monitoring if compute resources are adequate	4 – Real-time capable, but integration and timing jitter issues exist	3 – Monitoring possible but protocol-intensive
Integration	5 – Easily added to NFC-based systems, small sensor footprint, works with ESP32/NFC terminal stack	4 – Protocol layering can be added to existing networks	5 – Flexible in software, but hardware deployment on terminals is non-trivial	3 – Requires new hardware, new spectrum, and fails NFC back-compatibility	3 – Adds protocol and timing layers, not standardized across existing POS infrastructure

3.3.1.3 Conclusion on Design Choice

The IMU + NFC configuration emerged as the optimal design, scoring 27/30, owing to its high performance across all system criteria, particularly in physical proximity verification (5), challenge-responsiveness (5), and seamless integration into existing infrastructure (5). Unlike UWB and distance bounding, which demand specialized timing hardware or are vulnerable to optimized relay attacks, the IMU approach uses natural user interaction (device movement) as a biometric-style signature. ML-

based methods (standalone) scored well on anomaly detection (5) and monitoring (5) but were penalized for low compatibility with feedback mechanisms and challenge systems when not combined with physical sensing.

The solution introduces a novel feature:

- The system captures the card's orientation and transaction response time in real-time.
- These features are passed through a trained anomaly detection model embedded on the terminal, determining whether the interaction matches learned legitimate behaviors.
- Two main anomaly detection models, the statistical thresholding model and the ML model, are used for better accuracy.

This approach ensures that the payment device is physically present and actively used by the legitimate owner, making it exceedingly difficult for attackers to synchronize or spoof the motion and orientation during a relay attack.

3.4 System Architecture

The system is designed to prevent relay attacks in contactless payment systems by integrating an IMU (Inertial Measurement Unit) with NFC technology, utilizing a firmware-driven architecture to facilitate communication between the payment device and the POS terminal, shown in *Figure 3.1* below

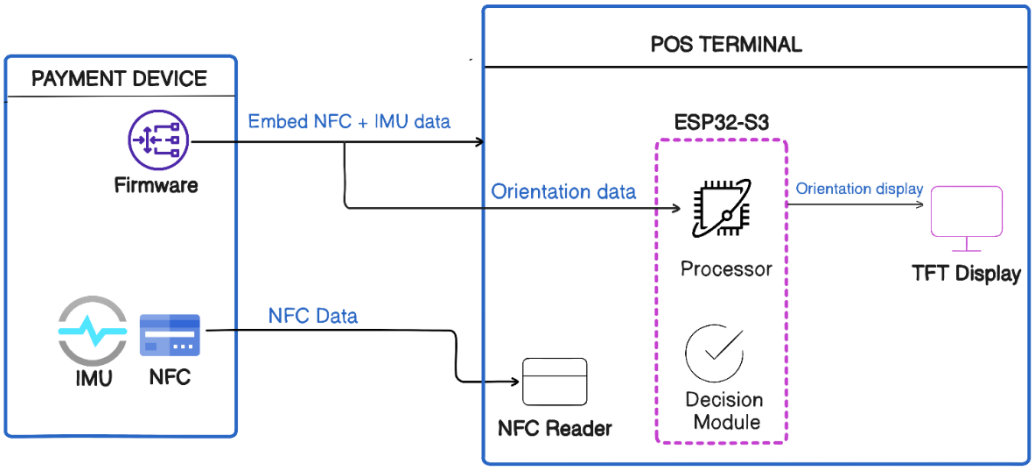


Figure 3.1 Systems architectural design, using IMU +NFC

3.4.1 High-Level Structure

a. Payment Device

The payment device consists of the following components:

a. **NFC Module:** Responsible for transmitting payment data to the terminal.
b. **IMU Module:** Captures motion and orientation data to ensure proximity verification.
c. **Firmware:** Embedded software that combines NFC and IMU data into a single message.

b. POS Terminal

At the terminal side, the combined NFC and IMU message is processed:

a. **NFC Reader:** Separates the NFC payment data from the IMU orientation data.
b. **Processor:** Analyzes the orientation data from the IMU.
c. **Decision Module:** Evaluates the validity of the orientation and proximity to detect potential relay attacks and determines whether to approve or reject the transaction.
d. **TFT Display:** Provides real-time visual feedback to the user.

3.4.2 System Workflow

1. **Initialization**: When the transaction terminal is powered, the ESP32 initializes the NFC reader and IMU, ready for payment.
2. **Card Placement and Orientation Capture**: Once the user places the card, the NFC reader begins authentication. Simultaneously, the IMU records real-time orientation data, specifically pitch and roll angles.
3. **Transaction Timing Measurement**: A timestamp is captured at the beginning and end of the transaction initiation to calculate the response time, which is essential for detecting timing-based relay attacks.
4. **Data Preprocessing**: The ESP32 preprocesses the captured orientation and timing data (normalization, conversion to features).
5. **Decision Inference**: The processed features are passed through the embedded decision model deployed on the ESP32. The model predicts whether the transaction is legitimate or anomalous based on learned patterns, then either approves or blocks the transaction.

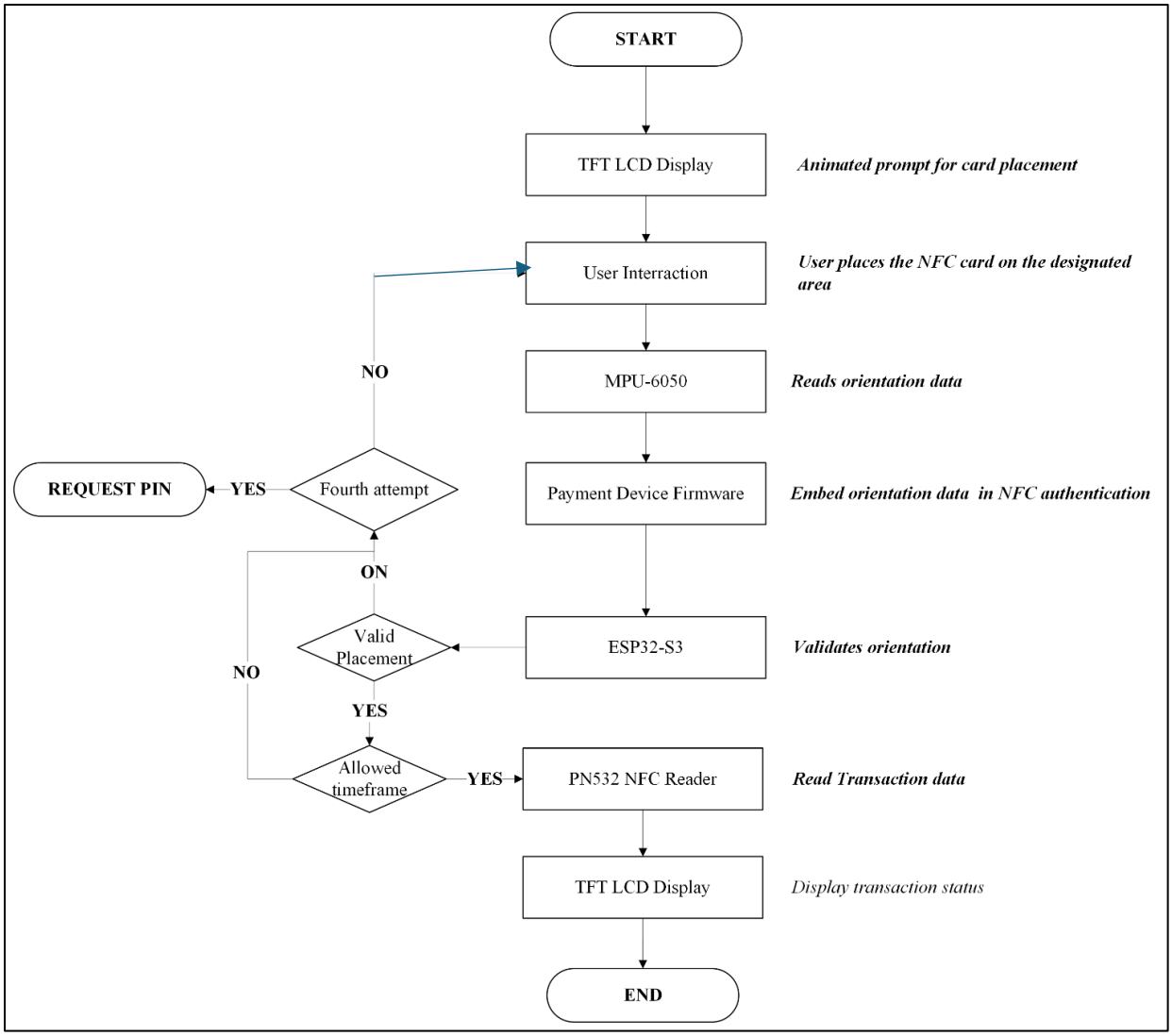


Figure 3.2 Systemic flow diagram for the NFC+IMU system for relay attack mitigation

3.5 Component Selections and Specifications

3.5.1 Component Selection Process

To ensure optimal performance, cost-efficiency, and energy consumption, a Pugh chart was used to evaluate the options for each critical subsystem. The system's design focuses on four main areas:

1. **NFC Module**: For secure communication and authentication in the payment process.
2. **Inertial Measurement Unit (IMU)**: To enable motion-based proximity verification.
3. **Microcontroller/Processor**: Responsible for real-time data processing.
4. **Display Options**: Providing visual feedback and interaction with users during the transaction process.

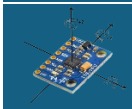
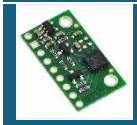
1. **NFC Module**

Option	Security Features	Ease of Integration	Cost	Power Consumption	TOTAL
 PN532 NFC	0	0	0	0	0 ✔ ✔
 RC522 NFC	-1	0	+1	0	0 ✘ ✘
 ST25R3911B	+1	-1	-2	-1	-2 ✘ ✘

Best Option: **PN532 NFC**

While the ST25R3911B offers better security, its high-power consumption and integration complexity reduce its suitability. Also, the RC522 has the same score as the PN532, it has very low security mechanisms, making it not the right choice for this project. The PN532 NFC provides a balanced baseline for secure and efficient implementation

2. **IMU (Inertial Measurement Unit)**

Option	*Proximity Verification*	*Real-Time Monitoring*	*Accuracy*	*Power Consumption*	*Cost*	*TOTAL*
 MPU-6050	0	0	0	0	0	0 ✔ ✔
 BNO055	0	+1	+1	-1	-2	-1 ✘ ✘
 LSM9DS1	0	0	0	0	-1	-1 ✘ ✘
 L3GD20	-1	-1	-1	+2	0	-1 ✘ ✘

Best Option: **MPU-6050**

Despite the BNO055's advantages in monitoring and accuracy, its higher cost makes the MPU-6050 the more practical choice, offering balanced performance without unnecessary overhead.

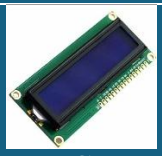
3. **Microcontroller/Processor**

Option	*Processing Speed*	*Real-Time Processing*	*Security features*	*Power Consumption*	*Cost*	*TOTAL*
 **Arduino Uno**	0	0	0	0	0	0 ✗ ✗
 **STM32**	+1	+2	+2	+2	+1	8 ✗ ✗
 **Raspberry pi**	+3	+1	+1	-1	-1	**2** ✗ ✗
 **ESP32-S3**	+2	+1	+3	+1	+2	9 ✓ ✓

Best Option: **ESP32-S3**

With its exceptional overall score of 9, the ESP32-S3 stands out for its processing capabilities and built-in security features, making it ideal for this project. Though STM32 is close, it consumes much power, making it less preferable.

4. Display options

Option	*Resolution*	*Refresh rate*	*Durability*	*Power Consumption*	*Cost*	*TOTAL*
 **LCD**	0	0	0	0	0	0 ✗ ✗
 **TFT Screen**	+2	+2	+1	-1	-1	3 ✓ ✓
 **LED Matrix**	-1	+1	+2	-2	-1	1 ✗ ✗
 **OLED**	+3	+2	-2	+1	-2	2 ✗ ✗

Best Option: **TFT Screen**

With the highest score of 3, the TFT Screen provides an optimal balance of performance and usability, making it the most suitable choice for dynamic display requirements

3.5.2 Hardware Specifications

a. Payment Device (NFC-IMU Card/Smartphone)

The payment device, which could be an NFC-enabled card or smartphone, serves as the primary interface for initiating contactless transactions. At the heart of this device is the PN 532 NFC chip, responsible for facilitating contactless data transmission at 13.56 MHZ between the payment device and the point-of-sale (POS) terminal at 2.7V to 5.4V. The PN 532 module is programmed to transmit NFC communication signals in writer mode. When a transaction is initiated, the NFC chip transmits essential payment information to the terminal, such as encrypted card details or tokenized data. It supports three communications interfaces: I2C, SPI, and Serial UART. The project will communicate with the POS using the Serial Peripheral Interface (SPI). PN 532 allows contactless communication using Higher Baud rates up to 424K Baud in both directions. This process ensures fast and secure data exchange, laying the foundation for a seamless user experience.

***Fig. 3.3**. PN 532 NFC/ RFID Module*

In addition to the NFC chip, the payment device is equipped with IMU sensors, specifically MPU6050 accelerometers and gyroscopes, that capture real-time motion and orientation data. MPU6050 is a Micro-Electro-Mechanical (MEMS) System and is the only 6-axis motion tracking device in the world designed for low power, low cost, and high-performance requirements. It has an MEM 3-axis for acceleration and an MEM 3-axis for gyro (X, Y, and Z axes simultaneously).

Accelerometer: Measuring ranges: ±2g ±4g ±8 g ±16g. Calibration tolerance: ±3%

Gyroscope: Measuring ranges: ±250/500/1000/2000 °/sar. Calibration tolerance: ±3%

The MPU6050 also has a (DMP) Digital Motion Processor, which is powerful enough to perform complex calculations. It works under a supply voltage of 2.3–3.4 V and consumption of 3.9 mA max. The device's firmware processes the captured IMU data, integrating this information with the NFC transmission.

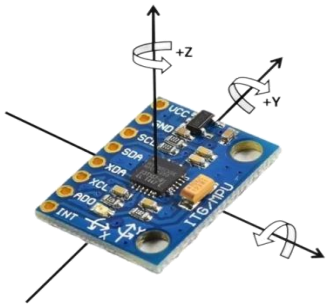


Fig. 3.4. MPU6050 Module

ESP32 low-powered microcontroller will process the NFC and IMU data on the payment card. These microcontrollers are typically powered by the device's battery, such as a lithium-ion cell, which is recharged via a USB-C port. Level shifters will also be used to ensure that the needed voltage is supplied from the ESP 32 since some components might require 5V and the ESP32 has 3.3V.

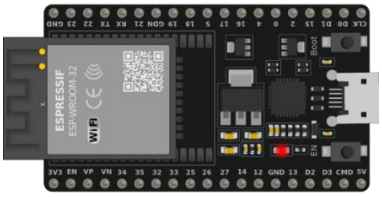


Fig. 3.5 Esp 32 Dev Board

b. POS Terminal

The POS terminal is the receiver and verifier in the contactless payment process. It is equipped with an NFC reader module (PN 532 NFC/ RFID Module, as shown in Figure 3 above) that captures the transmitted data from the payment device. The NFC chip will be programmed as a receiver or reader, which will only respond to communication from the payment device and ensure that the core payment information, such as encrypted card details, is received securely and accurately. Concurrently, the terminal receives the IMU data, which provides real-time information about the orientation and motion of the payment device.

The system uses 2.4. inch TFT LCD 240 x 320 touch screen for display. The display communicates via SPI communication protocol and uses the ILI9341 driver. The LCD uses either 3.3V or 5V in closed or opened mode. This step introduces an additional layer of verification, as the user must physically align the device according to the terminal's prompt.

Fig 3.6. 2.4. inch TFT LCD touch screen

The terminal's microcontroller or processor (Esp 32 Dev Board shown in Fig. 5 above) plays a central role in executing the verification algorithm, ensuring the transaction is analyzed in real time. The POS system will be powered by a rechargeable lithium battery of 5000mAh. This will ensure a constant power supply within the system.

3.5.3 Software Specifications

3.5.3.1 Payment Device Software

The payment device is equipped with embedded software to enable seamless communication with the NFC module and IMU sensors while processing motion data in real-time. This software operates on the selected microcontroller platform (ESP32-S3) and handles sensor data preprocessing and secure transmission to the POS terminal.

a. Firmware Development

The firmware is developed using embedded C or C++, which is selected for its low-level hardware access and efficiency. These languages allow direct manipulation of hardware registers and support real-time operations. Arduino IDE is employed.

b. IMU Data Processing

InvenSense Motion Driver (IMU Libraries), the MPU-6050 DMP (Digital Motion Processing) firmware library handles sensor fusion at the hardware level. This library offloads computational tasks from the microcontroller using the MPU-6050's internal DMP to preprocess the accelerometer and gyroscope data.

3.5.3.2 POS Terminal Software

The POS terminal software is designed to handle multiple functionalities, including secure communication, real-time orientation data analysis, and transaction processing. The software ensures that transactions are approved or rejected based on orientation verification and operates efficiently within the defined hardware and software constraints.

a. Operating Systems

The ESP32 operates to efficiently handle real-time tasks and meet the 500ms processing time constraint. It ensures concurrent execution of tasks such as NFC communication, IMU data analysis, and TFT display updates. Interrupt handling responds to critical hardware events like incoming NFC data or IMU sensor readings with minimal latency. High-priority tasks (e.g., NFC communication) precede lower-priority tasks (e.g., UI updates).

b. NFC Reader Driver

To manage communication between the ESP32 and the PN532 NFC module, the PN532 library is employed. This library is optimized explicitly for NFC operations and complies with the ISO/IEC 14443 communication standard, commonly used in payment systems. It allows the microcontroller to read and process NFC card or device data efficiently, ensuring compatibility with widely adopted payment protocols. The PN532 library is preferred for its easy integration with ESP32's architecture.

The **time delay** associated with NFC transactions, denoted as T_{trans}, is given by:

$$\boldsymbol{T_{trans}} = \frac{\boldsymbol{D\_data}}{\boldsymbol{R\_NFC}} \quad [3.1]$$

where:

- D_data is the amount of data being transmitted (in bits)
- R_NFC is the NFC transmission rate (in bits per second)

c. TFT Touchscreen Display

To operate the 2.4-inch TFT LCD, the software utilizes the Adafruit GFX and Adafruit ILI9341 libraries. These libraries are widely recognized for their robustness and compatibility with SPI-based TFT displays. The Adafruit GFX library handles the creation of graphical elements such as text, shapes, and images. In contrast, the Adafruit ILI9341 library is designed to manage low-level communication with the ILI9341 display controller.

d. Verification Algorithm

The orientation analysis is at the heart of the processing workflow, which plays a pivotal role in detecting fraudulent activity. This process uses quaternion-based sensor fusion algorithms to determine the payment device's orientation in real-time. The Euler Angle Algorithm will be used to calculate the system's pitch, roll, and Yaw angles. The Euler angles are derived from the rotation matrix, a 3x3 matrix that describes the orientation of an object in 3D space. The rotation matrix can be constructed from pitch, roll, and yaw angles as follows: For pitch (θx), roll (θy), and yaw (θz), the rotation matrix R is the product of three individual matrices:

1. **Pitch Rotation Matrix ($R_x(\theta_x)$)** $$= \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Cos(\theta x)} & \mathbf{-sin(\theta x)} \\ \mathbf{0} & \mathbf{sin(\theta x)} & \mathbf{cos(\theta x)} \end{bmatrix} \quad [3.2]$$

2. **Roll Rotation Matrix ($R_y(\theta_y)$)** $$= \begin{bmatrix} \mathbf{cos(\theta y)} & \mathbf{0} & \mathbf{sin(\theta y)} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{-sin(\theta y)} & \mathbf{0} & \mathbf{cos(\theta y)} \end{bmatrix} \quad [3.3]$$

3. **Yaw Rotation Matrix ($R_z(\theta_z)$)** $$= \begin{bmatrix} \mathbf{cos(\theta z)} & \mathbf{-sin(\theta z)} & \mathbf{0} \\ \mathbf{sin(\theta z)} & \mathbf{cos(\theta z)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \quad [3.4]$$

The total rotation matrix R is the product of the individual rotation matrices:

$$\mathbf{R = Rz(\theta z) \times Ry(\theta y) \times Rxx(\theta x)} \quad [3.5]$$

This matrix R represents the combined rotations of pitch, roll, and yaw. The Euler angles can be extracted from this matrix using **inverse trigonometric functions**.

The system compares the calculated orientation against an expected orientation profile. Angular deviations exceeding a predefined threshold $\boldsymbol{T}$ trigger an anomaly flag.

$$\mathbf{Valid} = \begin{cases} \boldsymbol{true, if} \ \ | \boldsymbol{target\_pitch - pitch} | \leq \boldsymbol{T} \ \boldsymbol{and} \ \ | \boldsymbol{target\_roll - roll} | \leq \boldsymbol{T} \\ \mathbf{false, otherwise} \end{cases} \quad [3.9]$$

3.5.3 System Power Consumption

Component	*Mode*	*Current (A)*	*Voltage (V)*	*Power Calculation*	*Power (W)*
Microcontroller	Active Mode	0.160	3.3	P=0.160×3.3	0.528
	Sleep Mode	10 µA (0.00001)	3.3	P=0.00001×3.3	0.000033
MPU6050 (IMU Sensor)	Active Mode	0.0039	3.3	P=0.0039×3.3	0.01287
PN532 (NFC Module)	Active Mode	0.100	3.3	P=0.100×3.3	0.330
	Idle Mode	0.020	3.3	P=0.020×3.3	0.066
TFT Touch Screen	Active Mode	0.150	3.3	P=0.150×3.3	0.495

Total Power for the Terminal *= 0.495 + 0.330 + 0.01287 + 0.528 = 1.36587 W*

Total Power for the payment device *= 0.330 + 0.01287 + 0.528 = 0.87087W*

Voltage Regulator Efficiency = 90%

$$\textbf{Adjusted Power} = \frac{\textit{Total Power}}{\textbf{0.9}} \qquad [3.10]$$

Adjusted power for Terminal = $\frac{1.36587}{0.9}$ = 1.52W

Adjusted power for Payment device = $\frac{0.87087}{0.9}$ = 0.968W

The NFC card uses energy harvesting to get charged for transactions. These passive NFC systems can harvest approximately 1–6 mW, depending on the strength of the RF field and the efficiency of the energy-harvesting circuitry [26]. However, the IMU in our system requires 12 mW, meaning careful system considerations must be taken in order to integrate the component into existing systems. This is further discussed in ***Section 6.2*** below.

Chapter 4: System Implementation and Integration

This chapter details the payment device's and terminal prototype's development and integration. The process involved initial simulations, schematics, hardware assembly, sensor calibration, data fusion, and the implementation of security measures.

4.1 Prototype Development

The implementation's initial phase involved simulating the system's core functionality using the WOKWI online platform. The simulation focused on the IMU (MPU6050) and its ability to detect anomalies in orientation data. The simulation allowed for the adjustment of the MPU6050's orientation to mimic real-world scenarios, such as card placement and orientation adjustments.

4.1.1 Simulation Overview

The simulation was designed to replicate the behavior of the IMU in detecting anomalies.

1. The system prompted the user to place the card in a specific orientation, shown in ***Fig 4.1***.
2. The user could then adjust the orientation of the MPU6050 to mimic card placement and orientation adjustments, as depicted in **Figure 4.2.**
3. The system provided feedback on whether the orientation was within the acceptable tolerance range or if an anomaly was detected, as illustrated in **Figure 4.3**.

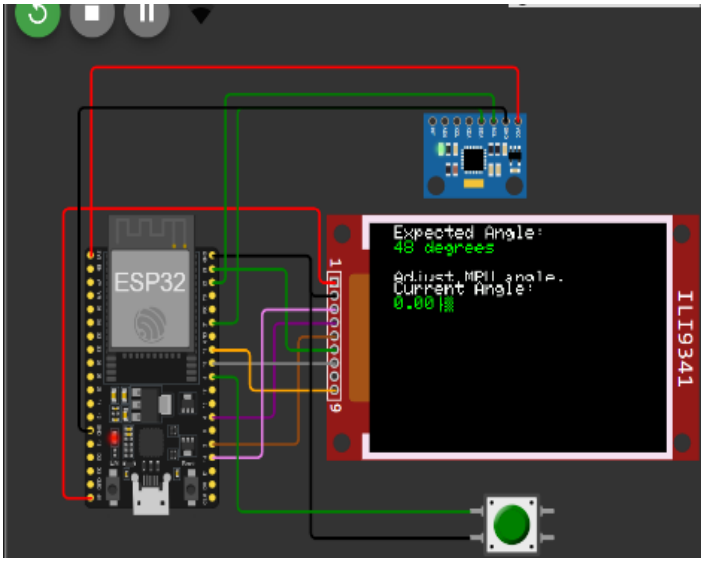


Figure 4.1. System initialization

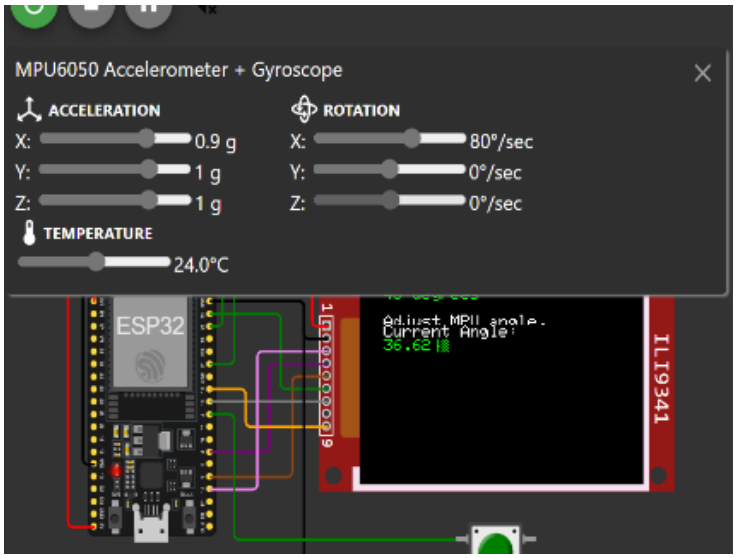


Figure 4.2. Adjust MPU

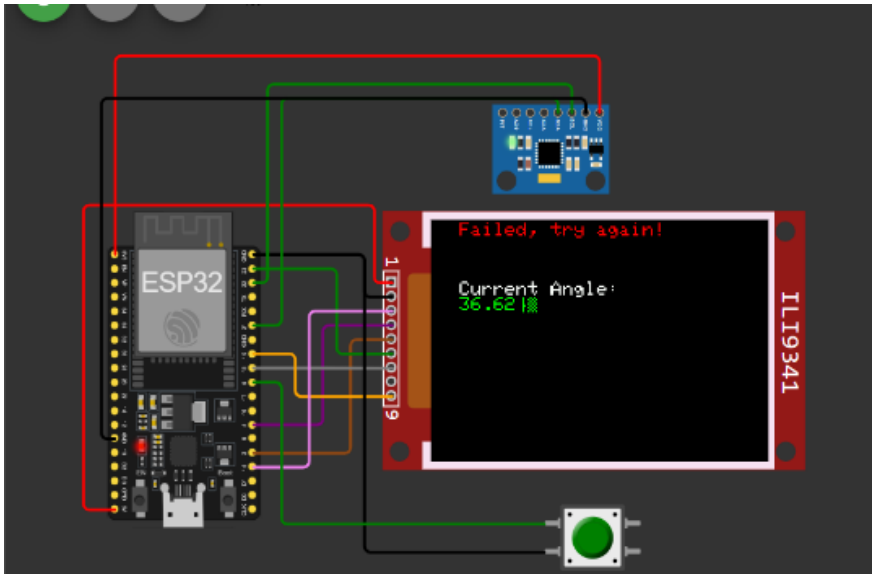


Figure 4.3. Feedback to user

4.1.2 Simulation Outcomes

The simulation allowed for the refinement of the anomaly detection logic, ensuring that the system could accurately identify deviations from the expected orientation. This phase was critical in validating the feasibility of the proposed approach before moving to the hardware implementation.

4.2 Payment Device Implementation

The physical implementation of the payment device involved designing the schematic, assembling the hardware, and developing the firmware required for NFC and IMU data fusion and encryption.

4.2.1 Hardware Design and Assembly

4.2.1.1 Schematic Diagram

A schematic diagram was designed to illustrate the connection between the NFC module (PN532) and the IMU (MPU6050). The schematic diagram of the system is shown in ***Figure 4.4***. The components were interfaced using the I2C protocol, with both PN532 and MPU6050 configured as slave devices with their respective addresses declared in the code.

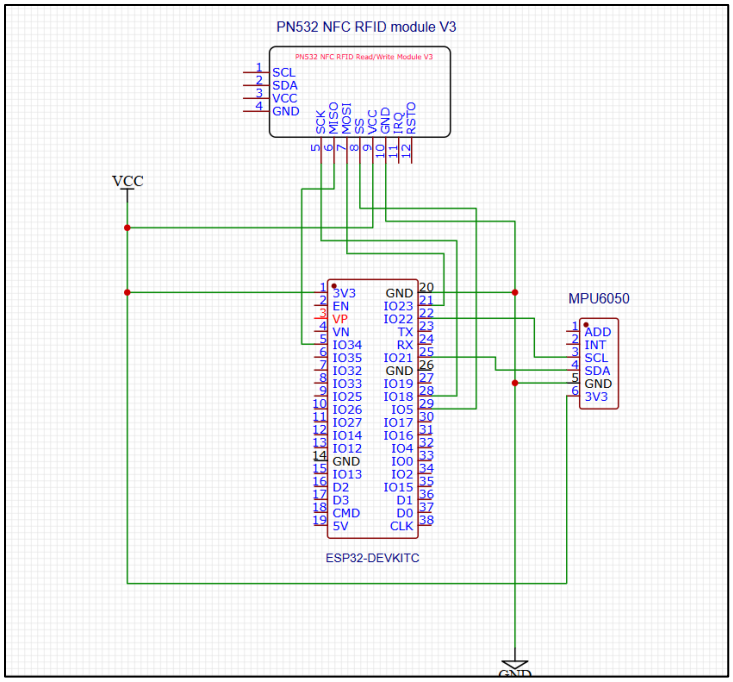


Figure 4.4 Schematic diagram of payment card

4.2.1.2 Circuit Assembly

The components were assembled on a breadboard to verify connectivity and signal integrity, which was later soldered on a pelf board. The PN532 and MPU6050 were connected to the same I2C pins (SDA and SCL) on the microcontroller.

4.3 Firmware Development

4.3.1 NFC Implementation

The PN532 module was configured in Peer-to-Peer (P2P) mode as a card emulator. Using the I2C protocol, the PN532 and MPU6050 module was interfaced with the microcontroller via the **Wire.h** library. Device addresses for both peripherals were declared, ensuring proper communication over the I2C line shared.

- ***PN532****: 0x24*
- ***MPU6050****: 0x68*

Additionally, Advanced Encryption Standard (AES-128) was integrated to secure transmitted data using **AESLib.h**. The encryption process included defining an AES key and initialization vector (IV) to transform the transmitted data into a secure format using the equation equation:

$$C=E(K, IV, P) \quad [4.1]$$

- *C is the ciphertext (encrypted data), E is the AES encryption function, K is the AES key, IV is the initialization vector, P is the plain text (unencrypted data).*

4.3.2 IMU Implementation

The MPU6050 IMU sensor was programmed to provide orientation data. The **MPU6050_6Axis_MotionApps20.h** library was utilized to leverage the onboard Digital Motion Processor (DMP), reducing processing overhead and filtering to reduce noise. The sensor measures raw gyroscope and accelerometer data, which is calibrated to remove bias. The calibration process involves calculating the average offset over a fixed number of samples (*N=2000*):

$$\boldsymbol{RateCalibrationRoll} = \frac{\boldsymbol{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{RateRoll_i} \qquad [\boldsymbol{4.2}]$$

$$\boldsymbol{RateCalibrationPitch} = \frac{\boldsymbol{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{RatePitch_i} \qquad [\boldsymbol{4.3}]$$

$$\boldsymbol{RateCalibrationYaw} = \frac{\boldsymbol{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{RateYaw_i} \qquad [\boldsymbol{4.4}]$$

The calibrated orientation data is then calculated as:

$$\boldsymbol{RateRoll = RateRoll_{raw} - RateCalibrationRoll} \qquad [\boldsymbol{4.5}]$$

$$\boldsymbol{RatePitch = RatePitch_{raw} - RateCalibrationPitch} \qquad [\boldsymbol{4.6}]$$

$$\boldsymbol{RateYaw = RateYaw_{raw} - RateCalibrationYaw} \qquad [\boldsymbol{4.7}]$$

The calibrated readings were filtered and converted into Yaw, pitch roll angles using the Euler angles in ***Chapter 3***. A 3D real-time orientation of the IMU was visualized using MATLAB.

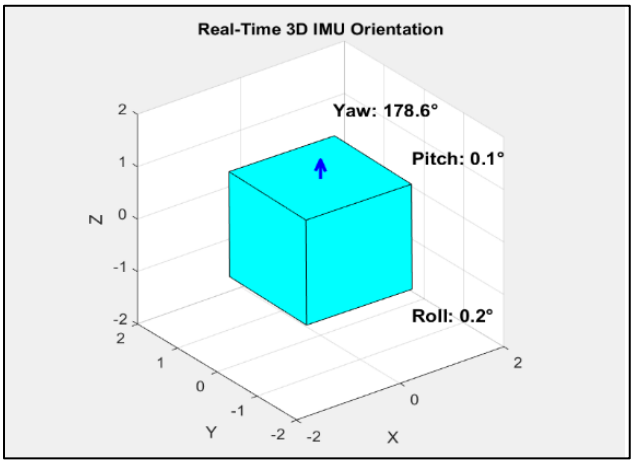


Figure 4.5 *Real-time 3D visualization of IMU readings in MATLAB*

4.3.3 Data Fusion and Encryption

The processed orientation data was fused with NFC transaction data into a structured binary packet of 13 bits. The *Table 4.1* below shows how the packet is structured for communication.

Bit Index	*Content*	*Size (Bits)*
0	Start Marker (0XAA)	1
1- 4	Yaw (φ)	4
5- 8	Pitch ($\emptyset$)	4
9-12	Roll (θ)	4

Table 4.1 NFC+IMU packet for transaction

The packet was then encrypted using AES encryption defined in equation ***[4.1]*** above, resulting in a 16-bit encrypted payload for transmission.

4.4 Implementation of the POS Terminal

The Point-of-Sale (POS) terminal is a critical component of the secure payment system. It is responsible for initiating communication with the payment device, decrypting the received data, and validating the transaction based on orientation and response time. This section details the design, hardware assembly, and programming of the POS terminal.

4.4.1 Hardware Implementation

The POS terminal integrates the PN532 NFC module and a TFT (Thin-Film Transistor) display for user interaction, connected via I2C bus and SPI respectively as shown in *Figure 4.6*

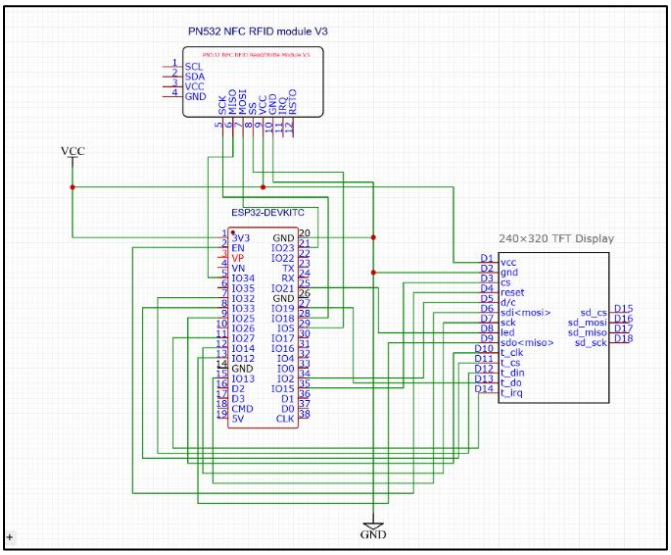


Figure 4.6 Schematic diagram of POS Terminal

4.4.2 Firmware Implementation

4.4.2.1 Decryption of the Communication Packet and Orientation Data Extraction

The POS terminal decrypts the received data using the AES algorithm. The same AES key and initialization vector (IV) used by the payment device are employed for decryption. The decryption process is described by the following equation:

$$P=D\ (K,\ IV,\ C) \qquad [4.11]$$

- *C is the ciphertext (encrypted data), D is the AES decryption function, K is the AES key, IV is the initialization vector, P is the plain text (unencrypted data).*

The decrypted data is then validated by checking the start marker (0xAA) and the packet length (13 bytes). Once the decrypted data is validated, the orientation data (yaw, pitch, and roll) is extracted from the packet to verify the legitimate transactions shown below:

```
// Check if the decrypted data is valid
if (decryptedData[0] == 0xAA && decrypted_len == 13) {  // Check for the start marker and correct length
  float yaw, pitch, roll;
  memcpy(&yaw, &decryptedData[1], sizeof(yaw));
  memcpy(&pitch, &decryptedData[5], sizeof(pitch));
  memcpy(&roll, &decryptedData[9], sizeof(roll));
```

4.4.5 Anomaly Detection and Security Measures

For anomalies detection, two main methodologies were employed. **Both statistical and Machine learning (ML) models** were used to detect relay attacks on the payment system. Over ***1000 real-time*** transactions were conducted and the orientation data (Yaw, Pitch, Roll) and the transaction time were collected for both the statistical and ML model training.

4.4.5.1 Statistical Threshold-Based Model

The dataset data was analyzed to calculate the mean and standard deviation of the orientation data. The mean (μ) and standard deviation (σ) were calculated using the following equations:

$$\boldsymbol{\mu = \frac{1}{N}\sum_{k=1}^{N} x_k} \qquad [4.12]$$

$$\boldsymbol{\sigma = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(x_k - \mu)^2}} \qquad [4.13]$$

$$\boldsymbol{CI = \mu \pm Z.\ \frac{\sigma}{\sqrt{N}}} \qquad [4.14]$$

- $N=1000$: Number of samples.
- X_k: Individual sample.

- *Z-score= 1.*98 at a 95% confidence interval

4.4.5.1.1 Target Orientation Generation and analysis

The orientation values for yaw, pitch, and roll were further visualized as shown in *Figure 4.7*. The graph shows that yaw varies significantly during transactions, as the card can be freely rotated around the Z-axis depending on how the user places it. Due to this inconsistency, yaws will not be used in decision-making. However, the pitch and roll values indicate a clear pattern, where legitimate transactions occur when the card is positioned close to ***±180*** degrees, with a mean of **~168±*12*** degrees. Using the 95% interval, the expected orientation range was found around ***|156| to |192|*** degrees for most legitimate transactions, as per the dataset. This insight was used as a basis for further analysis in determining valid transaction orientations.

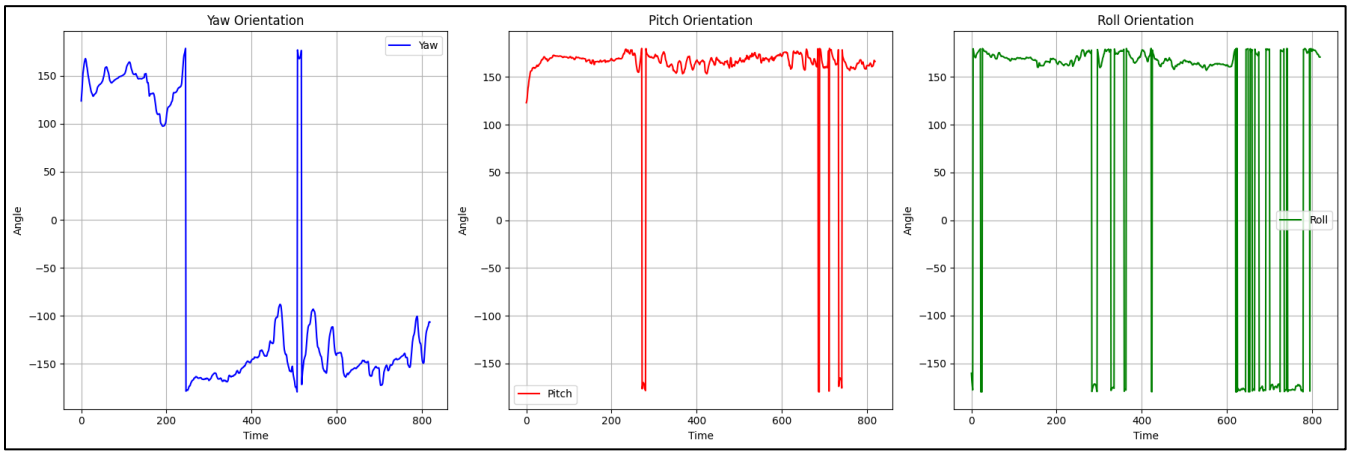


Figure 4.7. *Visualization of orientations data for legitimate transactions*

4.4.5.1.2 Orientation Validation

The received orientation data is compared to the target orientation range using a tolerance *(T)* in the equation 3.9 above. If the orientation mismatch exceeds the tolerance (T=±12.0), from the confidence interval, the transaction is flagged as anomalous.

Response Time Validation

To prevent relay attacks, the response time is validated. Legitimate transactions typically occur within 500 ms: $\boldsymbol{T_r \leq 500}$**.** If the response time exceeds this threshold, the transaction is flagged as suspicious.

Multi-Attempt Authentication

If the card placement orientation check fails three times, the system requests a PIN for additional verification.

4.4.5.1 Machine Learning-Based Model

Machine learning approach was employed as an alternative to improve detection accuracy and adaptive learning. This part will dive into data collection and cleaning, model selections, training the model and how it will be integrated for real-time fraud detection.

4.4.5.1.1 Data collection

Unsupervised models were considered for training, given that real-world fraud cases are often scarce hence it will be difficult to collect data to train the supervised model. Hence, the same legitimate transactions dataset used for the Statistical Model were used.

Feature Engineering, Selection, and Preprocessing

The original dataset included yaw, pitch, roll, and transaction time. However, based on the preliminary analysis, yaw was found to be less useful due to its minimal variation across different users as shown in ***figure 4.8*** *below*. Thus, roll, pitch, and transaction time were selected as the final features for training. Also, the data had different units, degrees (for angles) and milliseconds (for time). To ensure uniformity and convergence during training, the data was normalized using Min-Max scaling. The final dataset used for model training had a dimensionality of ***(1000, 3).***

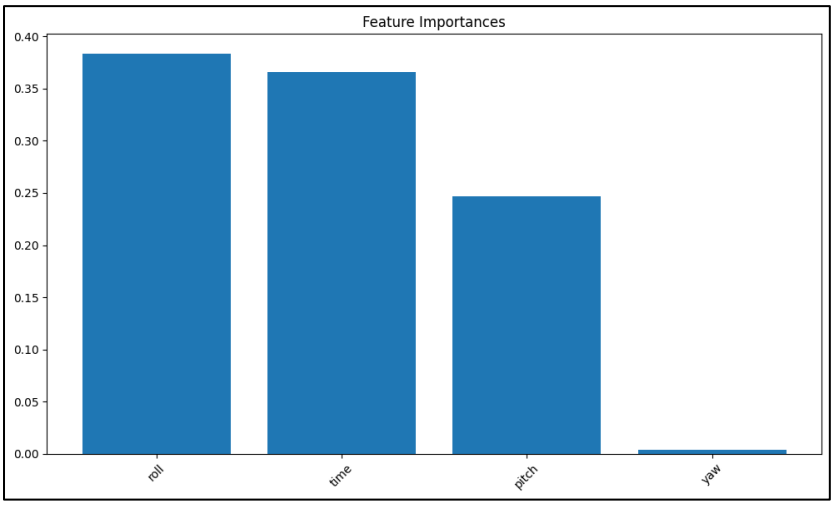


Figure 4.8 A graph showing how each feature is important for model training

4.4.5.1.2 Model Selections and Training

Two models were considered based on their suitability for anomaly detection and their light weight to support embedded systems (microcontrollers). These were the One-Class Support Vector Machine (OC-SVM) and Autoencoder, which are both Unsupervised.

Model	Purpose	Why Chosen
OC-SVM	Anomaly detection in legitimate transaction data	• Lightweight (suitable for microcontrollers) • Kernel-based (handles non-linear patterns) • Controllable sensitivity via ***nu*** parameter • Minimal memory footprint post-training
Autoencoder	Dimensionality reduction & reconstruction-based anomaly detection	• Learns compressed representations (efficient for embedded systems) • Robust to noise in IMU data • Flexible thresholding via reconstruction error • Can quantize to 8-bit for edge deployment

a. One Class Support Vector Machine (OC-SVM)

This is a version of the Support Vector Machine that is trained with only normal datasets (in this case, legitimate transactions) to detect outliers or anomaly. In our payment terminal context, the model learns the boundary of legitimate transactions using orientation patterns (roll, pitch) and timing data, then flags deviations as potential fraud.

$$K(x_i, x_j) = e^{(-y||x_i - x_i||^2)} \quad \textbf{[4.15]}$$

The model uses the Gaussian kernel (rbf) shown in ***equation 4.15*** to measure the similarities between the data points. The most critical hyperparameter considered is the ***nu (v),*** which controls the trade-off between catching fraud and minimizing false alarms. As shown in ***figure 4.9, t***wo ***nu*** values were used to simulate their influence on the decision boundaries and accuracy. Frame 1 has ***nu = 0.02*** and frame 2 has ***nu = 0.25***.

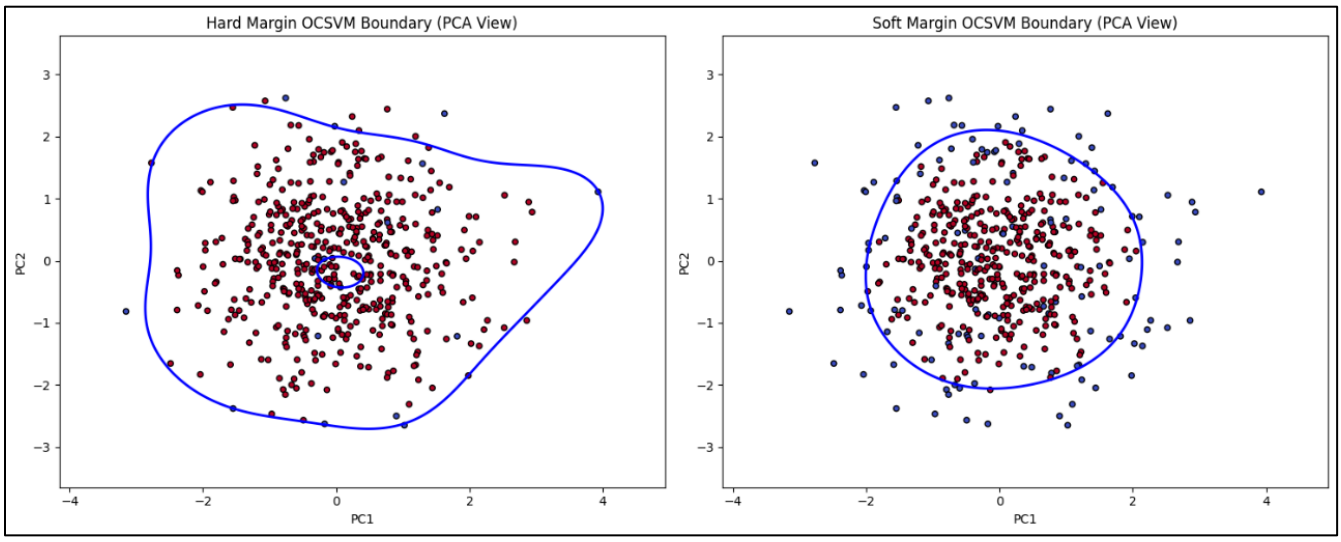


Figure 4.9 A graph showing the decision boundaries based on varying nu values

The choice of ***nu*** is critical to ensuring the high performance of the model. The ***nu = 0.02*** was chosen for training our model, assuming that 2% of the data are fraudulent. The model's performance was tested with 500 normal transactions and observing its ability to classify them as legitimate. The model achieved ***96% accuracy, 98% recall (fraud detection rate), and 98% f1-score*** as depicted in the confusion matrix below.

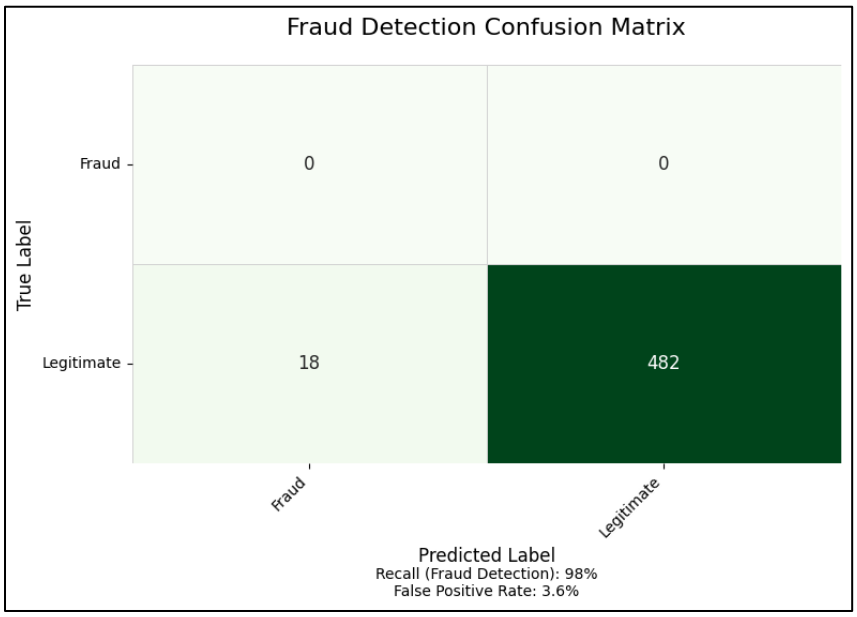


Figure 4.10 Confusion matrix showing how the OC-SVM performs.

b. Autoencoder

An Autoencoder neural network was designed with a symmetric architecture comprising encoding and decoding layers. The model was trained to reconstruct legitimate transaction data using these nodes, ***input=4, encode=8, bottleneck=2, decode=8, output=4*** shown in *Figure 4.11*

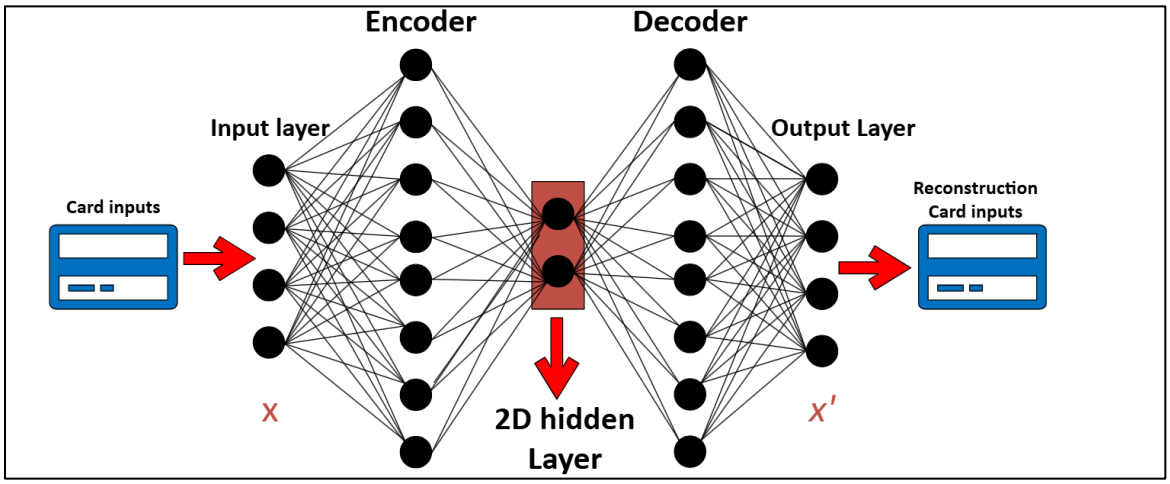


Figure 4.11. Architecture of Autoencoder with card input data

The encoder compresses the input and captures the relevant features, represented at the hidden layer. The decoder reconstructs the input from the compressed form and make it similar as possible to the original input. It uses the MSE to calculate the loss function with the equation ***4.16 below***.

$$\boldsymbol{L(\emptyset, \theta) = \frac{1}{N} \sum_{i=1}^{N} ||x_i - g_{\emptyset,}(f_\theta(x_i))||^2} \quad [4.16]$$

$$\boldsymbol{Anomaly = ||x - x'||^2} \quad [4.17]$$

$$\boldsymbol{Fraud = \begin{cases} true, & if\ Anomaly > T \\ false, & otherwise \end{cases}} \quad [4.18]$$

Transactions with a reconstruction error higher than a defined threshold **(T)** were flagged as anomalies. To select a suitable threshold, we simulated two scenarios with a higher threshold and lower threshold as shown in ***figure 4.12*** below. In frame 1, we assumed only 3 % of the transactions were fraud and in frame 2 we assumed 10% were fraud.

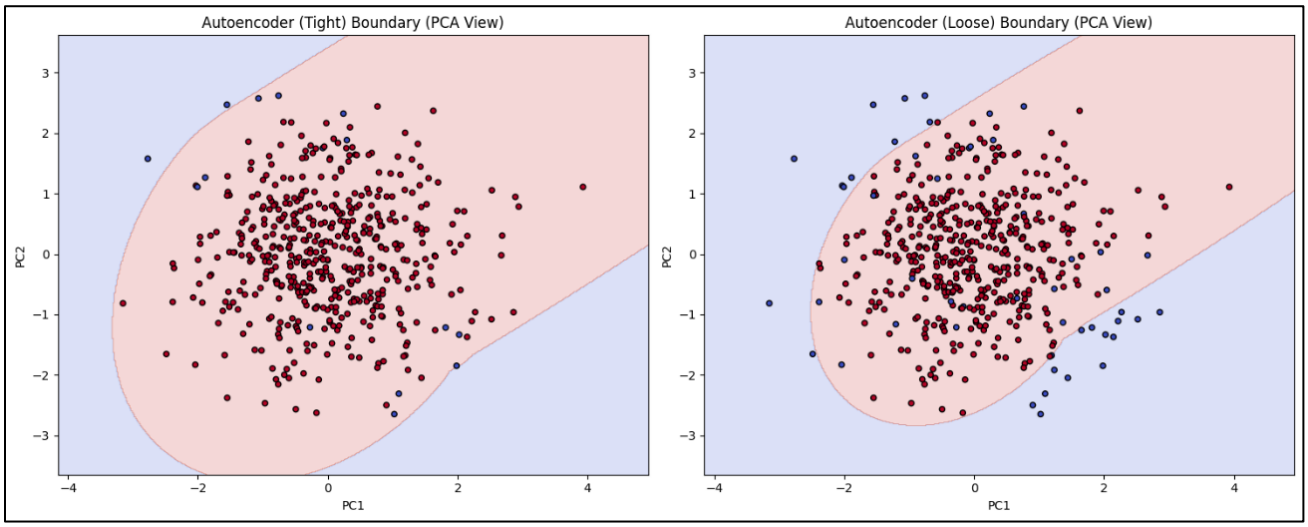


Figure 4.12 A graph showing the decision boundaries based on varying Thresholds

The figure shows that that the model is able to perform well with 2% fraud assumption, showing 97% accuracy in prediction and 98% recall, depicted by ***Figure 4.13***. It is then concluded that to achieve a very good model performance, a relatively higher threshold should be chosen to show there is always a low frequency of fraud available.

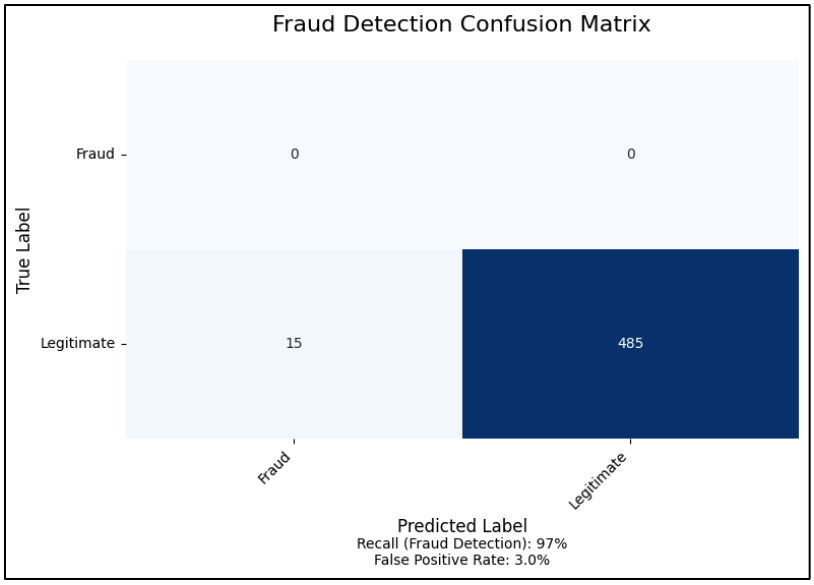


Figure 4.13 Confusion matrix showing how the Autoencoder Model performs.

4.4.5.1.3 Model Comparison

Both models performed well and can be used for fraud detection. However, based on our design requirements, we compared the two models to identify the one most suitable for implementation shown in Table 4.2.

METRICS	Autoencoder	OC-SVM
Inference time	3-5ms(quantized)	1-3ms
Memory Usage	15-25 kb (after 8-bit quantization)	5-10 kb
Training Complexity	high	Low
Library support	FTLite	Plain C/C++ (no dependencies)
Parameter Tuning	Complex layers	Simple (nu, gamma only)

Table 4.2 Comparing both models based on system design requirements

OC-SVM was chosen over the autoencoder given that it requires less resources for training and can be run on edge devices such as C/C++ code. This means that it can be used directly as a library, making it more lightweight and easier for implementation.

4.4.5.1.4. Model Integration into the Pos Terminal

To enable real-time fraud detection directly on the POS terminal, the trained model was optimized and deployed to run on an embedded system. The model was first converted into the **TensorFlow Lite (.tflite)**

format and subsequently transformed into a **C-compatible byte array** using the xxd utility with the command:

[xxd -i model.tflite > fraudModel.h]

This header file (fraudModel.h) was then included in the Arduino development environment and compiled alongside the firmware. In the final deployment, after the POS terminal reads the **orientation (pitch, roll)** and **transaction response time**, these three features are passed into the embedded model for classification. The integration ensures that all security checks are conducted **locally on the ESP32**, eliminating the need for cloud inference and enabling low-latency, secure, and privacy-preserving decision-making directly at the terminal.

Chapter 5: Testing and Results

This chapter presents the results of evaluating the proposed system under two different anomaly detection methodologies: ***Statistical Thresholding Approach and a Machine Learning-based Model.*** Both methods were tested on the same dataset, collected under controlled conditions.

5.1 Testing Objectives

a. ***Functionality Testing***: Does the system correctly process legitimate transactions?

b. ***Security Testing:*** Can the IMU effectively differentiate between legitimate and relayed transactions?

c. ***Performance Testing:*** How fast does the system authenticate transactions?

d. ***Robustness Testing***: Does the system perform well under different conditions (motion, varied orientations, interference)

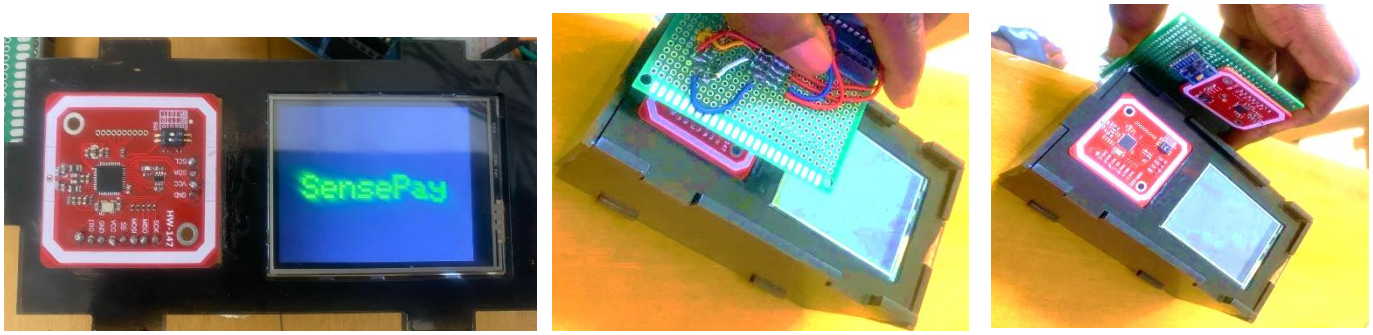


Fig 5.1 *a. SensePay setup* *b. Legitimate transactions* *c. orientation-based relay*

5.2 Testing Methodologies

5.2.1 Functional Testing (Legitimate transactions)

To evaluate the system's ability to correctly process legitimate transactions, we allowed 5 people to perform 20 legitimate transactions each as shown in ***Fig 5.1.b*** above, 10 when the system was running with ML anomaly detection and 10 when it was running with statistical thresholding, yielding n=50 on each model. ***Table 5.1*** and ***Figure 5.2*** summarize the performance metrics recorded.

Table 5.1: Summary Statistics for Legitimate Transactions

Metric	Statistical Model	OC-SVM Model
Mean Acceptance Rate	70%	96%
Mean Pitch (Degrees)	157.60	160.15
Standard Deviation (Pitch)	±22.02	±8.28
Mean Roll (Degrees)	162.92	166.16
Standard Deviation (Roll)	±22.92	±10.82

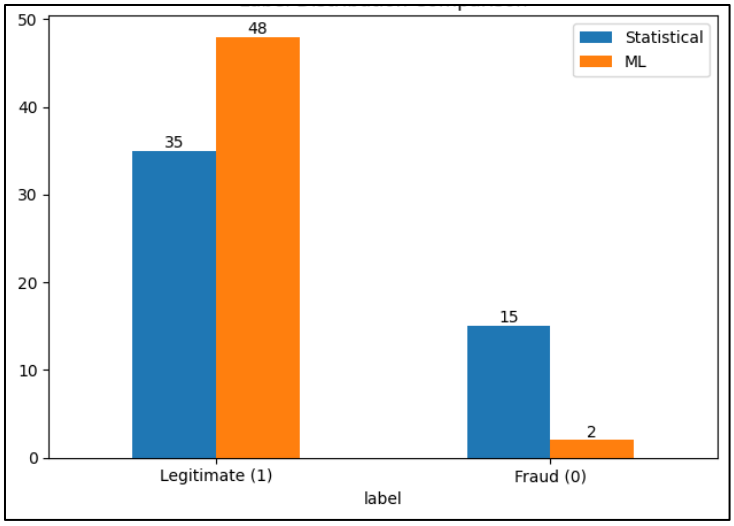


Figure 5.2 A graph showing accepted transactions under both models

5.2.1.1 Transaction Orientation Analysis

Using the logged orientation data, a **95% confidence interval (CI)** was calculated for the pitch and roll values to estimate their statistical reliability. This yielded a pitch with confidence range of approximately [160° ± 12°] for the statistical model and [155°±8°] for the ML model as well as [170°±10°] roll for the statistical model and [168°±8°] for the ML. The distribution of pitch and roll angles for accepted transactions was further visualized using line graph to assess symmetry and spread as shown in ***Figure 5.3***. The pitch angles are tightly clustered around approximately **158–165°,** especially in the ML model, indicating a typical device tilt during normal use, showing about **15°** tilt from **180°.** The Roll angles were more clustered at **170°** for both models.

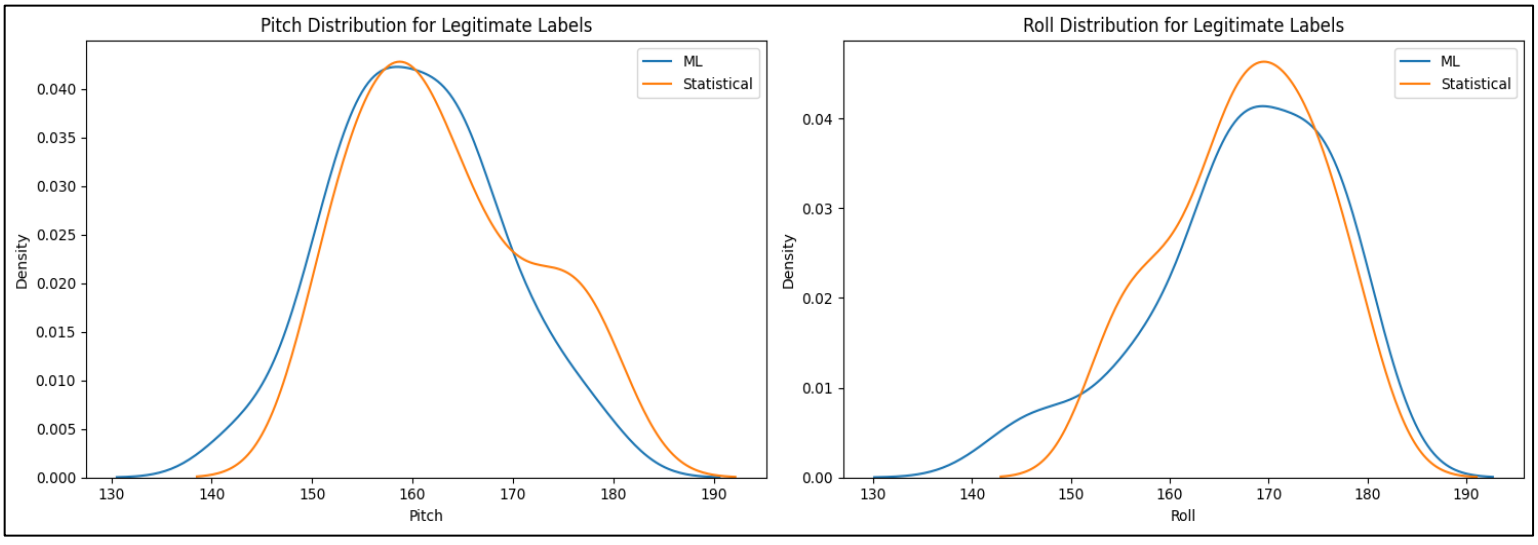


Figure 5.3 A line graph showing the orientation data from legitimate transactions

5.2.1.2 Post-hoc Analysis of failed transactions

Out of 50 transactions for the statistical models, 15 were incorrectly flagged as relay attacks. A post-hoc analysis was performed to investigate the causes of these false negatives. Approximately 27% of these cases involved borderline pitch angles within tolerance but were paired with slightly out-of-bound roll values, 60% with roll angle within tolerance but paired with out of bound pitch values and lastly approximately 3% with both angles deviated. Further analysis revealed these root causes:

a. **Gyroscopic drift:** In 9 out of 15 cases, the gyroscope drift bias exceeded **3°/s** during card placement (versus typical **0.5°/s**), triggering anomaly flags.
b. **NFC Signal Attenuation:** The remaining failure occurred when the card was misaligned by >30° from the terminal's antenna reducing the signal strength, hence causing failure.

These insights indicate that rigid thresholds lack the flexibility to account for real-world user variability, while it further emphasizes frequent re-calibration for more reduced noise.

5.2.2 Attack Simulation (Security Testing)

This section evaluates the system's ability to detect suspicious relayed transactions by simulating common attack scenarios. We specifically tested two types of relay attack simulations, *delays in response time and abnormal device orientation,* to assess how well the Statistical and Machine Learning (OC-SVM) models can differentiate between legitimate and suspicious activity.

5.2.2.1 Relay Attack Simulation with Injected Delay

In this scenario, artificial delays were introduced to mimic relay attacks. According to literature, normal legitimate transactions typically complete within approximately **500ms**. Given that our system completes transactions in **~285ms** on average *(see Section 5.3 below)***,** we added a randomized sleep delay ranging from **300ms to 400ms** during transaction execution. This ensured that the total observed response time exceeded 500ms, thereby simulating the delay incurred during a relay attack.

As demonstrated in *Figure 5.4*, the systems successfully detected all attacks exceeding the empirically derived threshold of 322.2 ms (calculated as $\mu + 3\sigma$, where $\mu = 285$ ms and $\sigma = 12.4$ ms). Both models' sensitivity to time-based anomalies likely stems from their ability to learn fine-grained variations in response time beyond the preset thresholds.

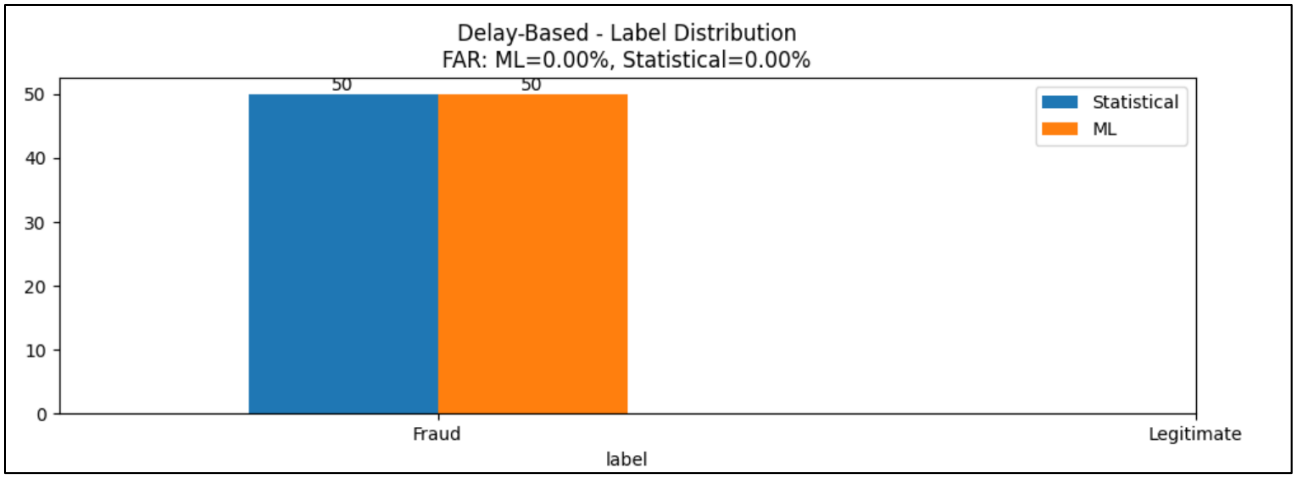


Figure 5.4 A graph showing delay-based relay attacks on both models

5.2.2.2 Relay Attack Simulation with Abnormal Orientation

This test involved real-time transactions where the device was intentionally tilted at unusual angles, often near **90°**, to replicate compromised or awkward handling that differs from the legitimate pitch/roll range **(see Figure 5.1.c).** The goal was to observe whether both models could detect these as suspicious orientation patterns.

As illustrated in ***Figure 5.5,*** the OC-SVM model outperformed the Statistical model, achieving a 94% detection rate compared to 84%. The OC-SVM model's higher dimensionality handling allows it to capture subtle orientation deviations across pitch and roll simultaneously, while the Statistical model relies heavily on static range rules, leading to more false negatives.

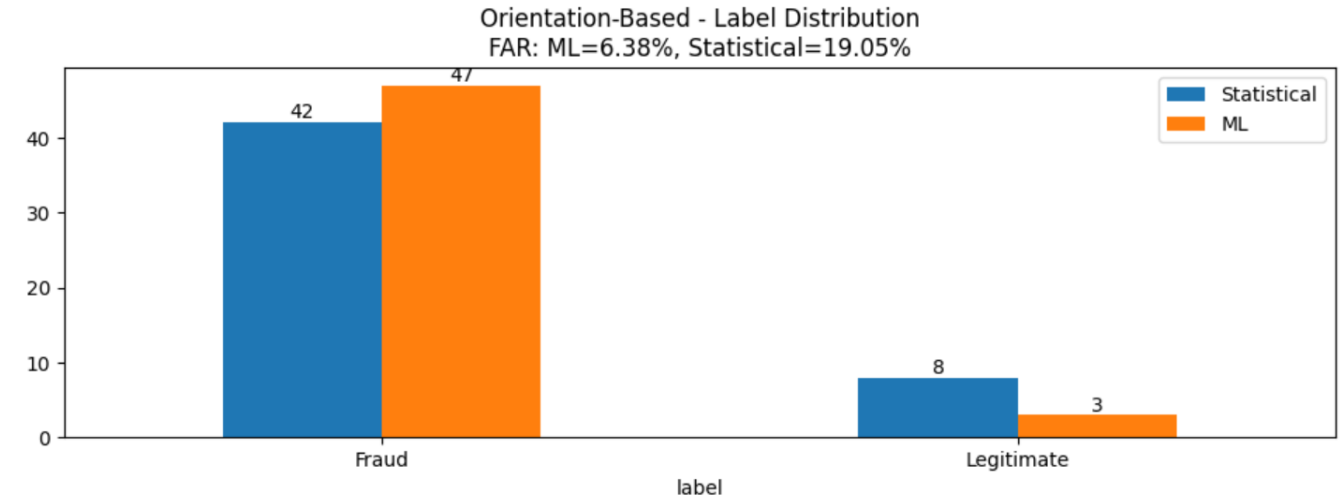


Figure 5.5 A graph showing results for orientation-based relayed attack

Overall Observation:

Both models effectively detected simulated delay-based attacks with a **zero false acceptance rate (FAR)**. For orientation-based attacks, the ML model showed superior accuracy with a FAR of **6.4%** and **19% FAR** for the statistical model. These results highlight the ML approach's robustness .

5.2.3 Performance Testing (Speed & Accuracy)

5.2.3.1 End-to-end Processing Latency Breakdown

To evaluate the performance, timestamps were recorded at critical processing stages while performing transactions: *IMU reading, DMP processing, data encryption, NFC transmission, and total end-to-end processing*. The goal was to determine the latency introduced by integrating IMU-based orientation validation into the transaction flow and the results were recorded in Table 5.2.

Stage	*Mean (us)*	*Std Dev (us)*	*Min (us)*	*25% (us)*	*Median (us)*	*75% (us)*	*Max (us)*
IMU Read	2412.64	45.37	2288	2384	2384	2468	2472
DMP Processing	4319.12	776.12	3936	3952	4016	4035	6152
Data Encryption	3205.84	149.04	2776	3228	3232	3316	3316
NFC Transmission	214858.08	124.47	214588	214813	214880	214922	215088
Total Processing	224795.68	622.47	224140	224397	224592	224820	226524

Table 5.2 Timing Statistics for Different Transaction Stages

Analysis

The results in ***Table 5.3*** below demonstrate that the bulk of the transaction processing time is concentrated in the NFC transmission phase, accounting for **over 95%** of the total processing time. The contributions from the IMU read, DMP processing, and data encryption stages are relatively minimal, collectively accounting for **less than 5%** of the total time. This confirms that the addition of the IMU module and associated computations introduces negligible latency to the system. The IMU read and DMP processing operations together average under **7 milliseconds**, and encryption adds only around **3 milliseconds**. These values are significantly overshadowed by the time taken for the NFC communication itself, which is an inherent requirement of the contactless system.

Processing Stage	*Time (%)*
IMU Read	1.07%
DMP Processing	1.92%
Data Encryption	1.43%
NFC Transmission	95.58%

Table 5.3 Time Distribution by Stage as a Percentage of Total Processing Time

5.2.3.2 Response Time Comparison: ML vs Statistical Model

The response time analysis in ***Figure 5.6a*** below shows that the ML-based model does not cause any noticeable delay compared to the statistical approach, with both averaging around **285.26 ms** during a legitimate transaction**.** Notably, all transactions were completed well below the standard **500-millisecond threshold** commonly associated with legitimate contactless transactions. This real-time efficiency is possible because the ML model runs locally on the microcontroller as C code, avoiding the overhead of server communication. Under simulated attacks, **delay-based relays** raised response times to **~660 ms**, while **orientation-based relays** remained around **285 ms**, similar to legitimate behavior. As shown in *Figure 5.6. b,* this highlights that time alone is insufficient to detect all attack types, reinforcing the benefit of integrating ML for deeper anomaly detection.

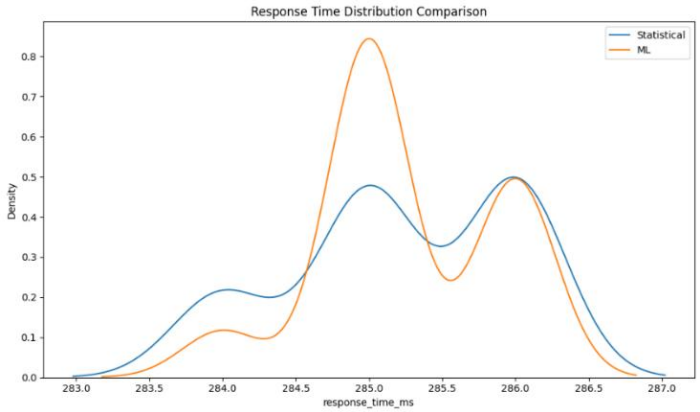


Figure 5.6a A line graph showing the response time for both models in milliseconds

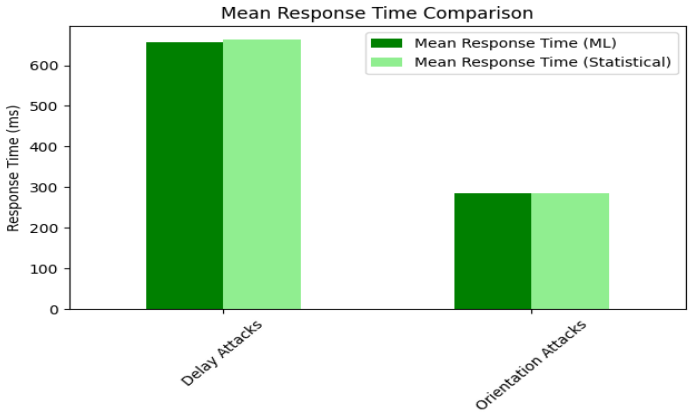


Figure 5.6b Mean response comparison

5.2.3.2 Accuracy Testing

We quantitatively evaluated the performance of the two models, subjecting them to accurate testing using a balanced dataset comprising both legitimate and simulated relay attack transactions. Each model processed 50 legitimate transactions and 100 relay attack attempts (50 delay-based, 50 orientation-based),

and performance metrics were derived from the resulting classification outcomes. The following standard metrics were used to assess model accuracy:

$$\boldsymbol{FRR} = \frac{False\ Negatives\ [FN]}{Total\ Legitimate\ Transactions} \quad [5.1]$$

$$\boldsymbol{FAR} = \frac{False\ Positive\ [FP]}{Total\ Relay\ Attacks} \quad [5.2]$$

$$\boldsymbol{Accuracy} = \frac{TP+TN}{Total} \quad [5.3]$$

$$\boldsymbol{Precision} = \frac{TP}{TP+FP} \quad [5.4]$$

$$\boldsymbol{Recall\ (Sensitivity)} = \frac{TP}{TP+FN} \quad [5.5]$$

A. Statistical Threshold-Based Model

	Predicted Legitimate (1)	***Predicted Relay Attack (0)***
Actual Legitimate (1)	TP = 35	FN = 15
Actual Relay Attack (0)	FP = 8	TN = 92

Table 5.4 Confusion Matrix table for accuracy analysis on Statistical Model

The statistical model, though relatively simple and lightweight, exhibited noticeable sensitivity to legitimate variability, resulting in a comparatively high false rejection rate as shown in ***Table 5.4***. While its low FAR (0.08) suggests decent attack filtering capacity, it lacks robustness against borderline legitimate behaviors.

B. Machine Learning-Based Model (OC-SVM)

	Predicted Legitimate (1)	***Predicted Relay Attack (0)***
Actual Legitimate (1)	TP = 48	FN = 2
Actual Relay Attack (0)	FP = 3	TN = 97

Table 5.5 Confusion Matrix table for accuracy analysis on ML Model

The ML-based approach significantly outperformed the statistical model across all measured metrics. Its ability to adaptively learn boundary patterns allowed effective separation between legitimate behavior and anomalous activity, minimizing both false positives and false negatives with about 94% precision in ***Table 5.5***. This confirms the OC-SVM model's strength in generalizing across varying attack vectors.

5.2.3.2 Statistical Analysis

a. McNemar and Chi-Square Test

A Chi-Squared Test of independence was conducted to assess whether model types significantly influence detection outcomes. Also, to determine whether the performance difference was significant, McNemar's test was applied. The results for both were statistically significant, $\chi^2 (1, N = 100) = 10.2025, p = 0.0014$, significantly lower than ***0.05***, indicating that prediction accuracy was dependent on the choice of detection model. This result further confirms the McNemar's test that the OC-SVM model provides a statistically significant performance enhancement over the threshold-based model.

b. ROC - AUC Analysis

A further ROC-AUC analysis on sensitivity and specificity on both models reveals that the OC-SVM model outperforms the Statistical model, achieving a higher Area Under the Curve (AUC) of 0.96 compared to 0.81 shown in ***Figure 5.7***. This indicates that OC-SVM has a stronger ability to distinguish between classes, with a significantly lower false positive rate and higher true positive rate.

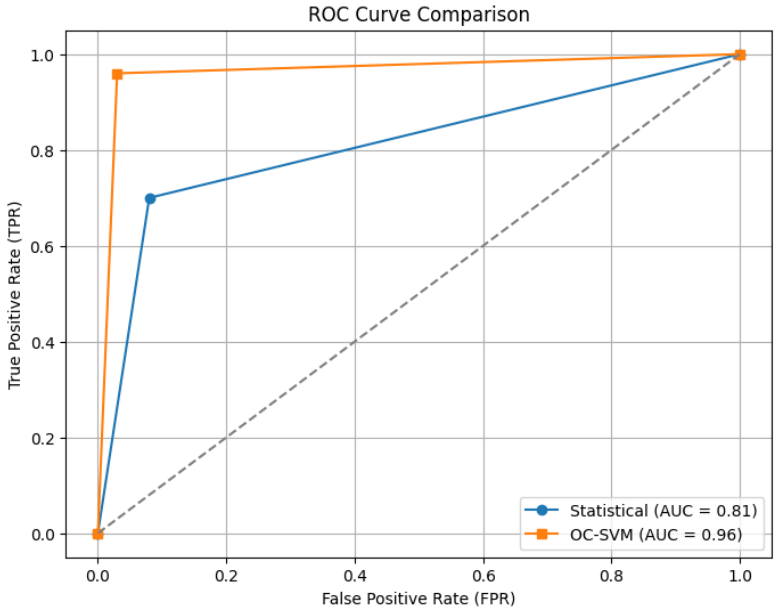


Fig 5.7 ROC-AUC curve for both model for performance testing

5.2.3 Robustness Testing (Motion-based Testing)

To evaluate the robustness of the system in real-world scenarios, a dynamic test was conducted wherein the subject interacted with the system while shaking. This type of testing is critical for determining

how the system performs under motion, which is typical in everyday mobile or wearable device use cases.

A total of **50 transactions** were performed under these motion-induced conditions. The outcomes were classified into three categories: *successful, failed, and blocked*. ***Figure 5.8*** below shows the distribution. The results are as follows:

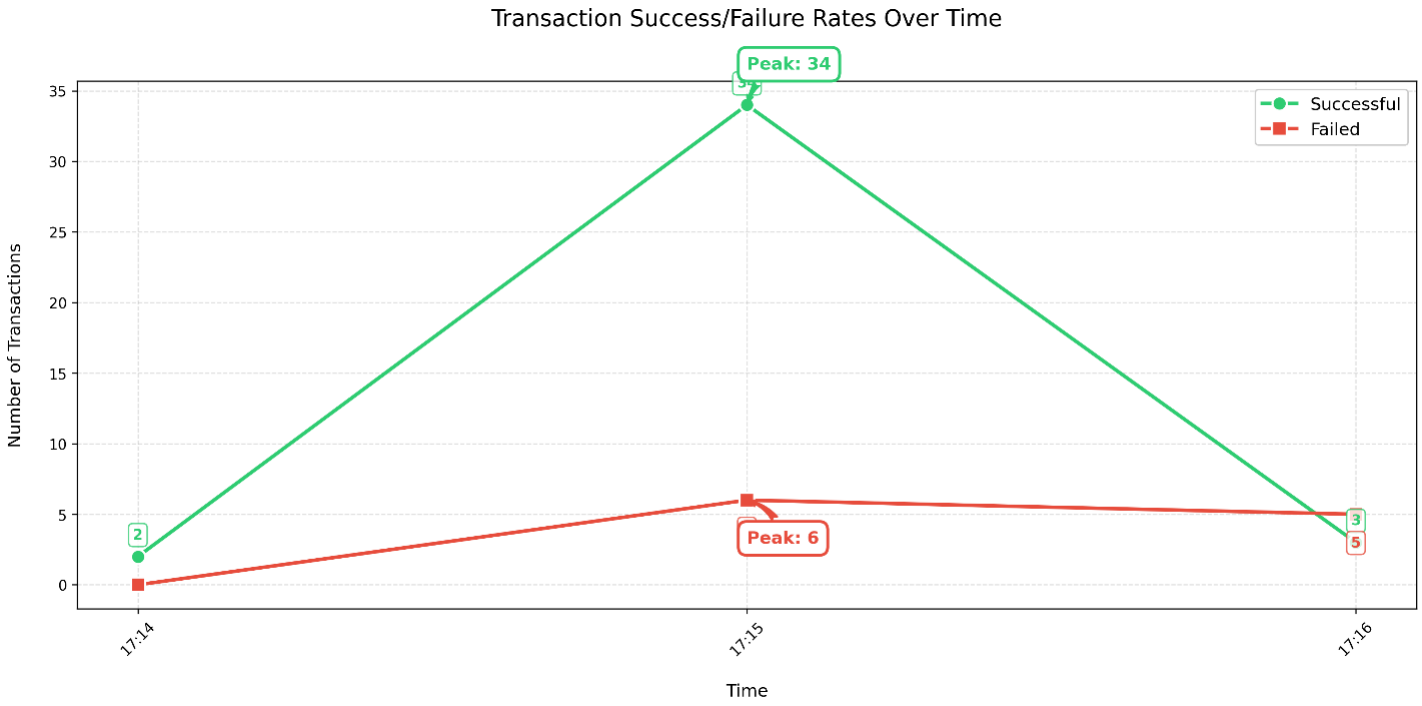


Figure 5.8: Line graph showing distribution of transaction outcome under motion conditions

These results indicate that the system retained a **high success rate of 78%**, demonstrating its potential to operate reliably even in the presence of physical disturbances. The **22% failure rate**, however, highlights limitations in handling dynamic inputs, which may be attributed to factors such as signal disruption, misalignment in sensor timing, or the challenge of accurately capturing biometric or positional data while in motion. Notably, **no transactions were blocked** by the security layer, suggesting a low false positive rate during motion, though this may also imply under-sensitivity to abnormal behaviors in such scenarios.

Chapter 6: Discussion, Conclusion, and Future Work

6.1. Discussion of Results

The experimental evaluation confirmed that both statistical time-series analysis and machine learning techniques are effective for detecting relay attacks in contactless payment systems. However, the comparative analysis demonstrated that the One-Class Support Vector Machine (OC-SVM) model consistently outperformed the baseline statistical method across multiple evaluation criteria.

In terms of accuracy, the machine learning-based approach achieved a superior classification rate of 96.7%, compared to 79% for the statistical model. Furthermore, the OC-SVM exhibited a significantly lower False Rejection Rate (FRR) of 4%, compared to 30% for the statistical approach, indicating a substantially better ability to correctly identify legitimate transactions without undue rejections. Similarly, the False Acceptance Rate (FAR) for the OC-SVM model was observed at 3%, outperforming the 8% FAR recorded for the statistical model, although neither approach fully achieved the EMV-recommended FAR of 1%.

Critically, response time analysis demonstrated that both models maintained an average transaction latency of approximately 285 ms, well within the acceptable 500 ms limit for contactless payments. This result validates that integrating the anomaly detection layer, especially the ML model, does not compromise transaction speed or user experience. Notably, even during orientation-based relay attacks, transaction response times remained indistinguishable from those observed during legitimate transactions. This observation reinforces the need for multidimensional anomaly detection approaches, as timing alone cannot reliably distinguish all forms of relay attacks.

When evaluating model performance across different attack vectors, both approaches effectively detected delay-injected relay attacks, attributable to their divergence from expected timing profiles. However, for orientation-based attacks, significant differences emerged. The statistical model, constrained by static pitch/roll thresholds, exhibited a higher rate of false negatives. In contrast, the OC-SVM model leveraged its ability to model complex, non-linear feature distributions, achieving a 94% detection rate compared to 84% for the statistical method. This adaptability highlights the ML approach’s robustness in scenarios involving user variability and more sophisticated attack patterns.

Overall, the findings demonstrate that while simple thresholding offers some protection, machine learning-based anomaly detection provides a substantially more resilient and scalable solution for securing contactless transactions against evolving relay attack strategies.

6.2 Practical Implementation Considerations

The proposed fraud detection framework introduces a novel distributed architecture in which the inertial sensing component is integrated into the payment device (e.g., smartphone or smart card), while the anomaly detection model (One-Class SVM) is deployed locally on the POS terminal. This design enables real-time fraud detection without requiring external server communication, thereby preserving user convenience and transaction speed.

6.2.1 System integration

To facilitate widespread deployment, the system can be integrated across multiple platforms, each offering unique advantages depending on the user environment.

6.2.1.1 Integration with mobile Devices

Modern smartphones are equipped with high-precision IMUs, making them well-suited for motion-based verification during NFC transactions. In this scenario, a dedicated mobile payment application or embedded SDK would access the device's orientation at the point of NFC interaction. Real-time pitch and roll data would be extracted, pre-processed, and evaluated against the embedded anomaly detection model directly within the app.

Advantages:

- No hardware modifications required on the user side.
- Leverages existing smartphone IMU accuracy.
- Localized data processing enhances privacy and security.

6.2.1.2 Integration with smart cards

In environments where smartphone usage is impractical, next-generation smart cards equipped with ultra-low-power IMUs (e.g., Bosch BMI270) offer a viable alternative. Orientation data is collected during the card’s presentation to the terminal and transmitted via NFC for fraud analysis.

Design Considerations:

- Duty cycling techniques and onboard energy storage elements (e.g., capacitors) are critical to operate within NFC's typical 1–6 mW energy harvesting constraints.
- Selected IMUs must consume less than 50 µW average power to ensure compatibility with passive card architectures.
- Mechanical design must maintain compliance with EMVCo form factor and durability standards.

6.2.1.3 POS Terminal ML Inference Integration

The fraud detection model, after being trained offline, is compressed into a lightweight format (e.g., TensorFlow Lite or C array representation) and embedded within the POS terminal firmware. This allows real-time classification of transaction legitimacy based on incoming orientation and timing features without external network dependencies.

Benefits:

- Entire fraud detection process is performed locally, preserving transaction speed.
- Average decision latency remains well within acceptable real-time limits (<300 ms).
- Architecture is compatible with most modern POS systems through firmware updates, minimizing hardware overhaul.

6.3 Future Works

While the proposed system demonstrated promising results, several areas remain for future enhancement. First, **expanding the dataset** with more diverse user behaviors and transaction environments will improve the model's generalization. Additionally, **exploring alternative lightweight models,** such as quantized neural networks or ensemble learning, could offer better adaptability with minimal computational overhead.

From a hardware perspective, future iterations can **investigate integrating IMUs into flexible smart card substrates** and improving energy harvesting techniques for passive operation. Another important area is the **development of custom testing software** capable of simulating relay attacks that include

orientation data. Current tools like NFCGate focus solely on relaying NFC data without considering device spatial behavior. Creating specialized software that can inject both timing delays and manipulated orientation profiles would enable more realistic and comprehensive testing of the system's security under real-world attack conditions.

6.4 Conclusion

This research presented a novel motion-based fraud detection framework designed to enhance the security of contactless payment systems. By integrating IMU-based orientation sensing on the payment device with a lightweight machine learning model embedded directly within the POS terminal, the system introduces a real-time behavioral authentication layer that strengthens traditional NFC security.

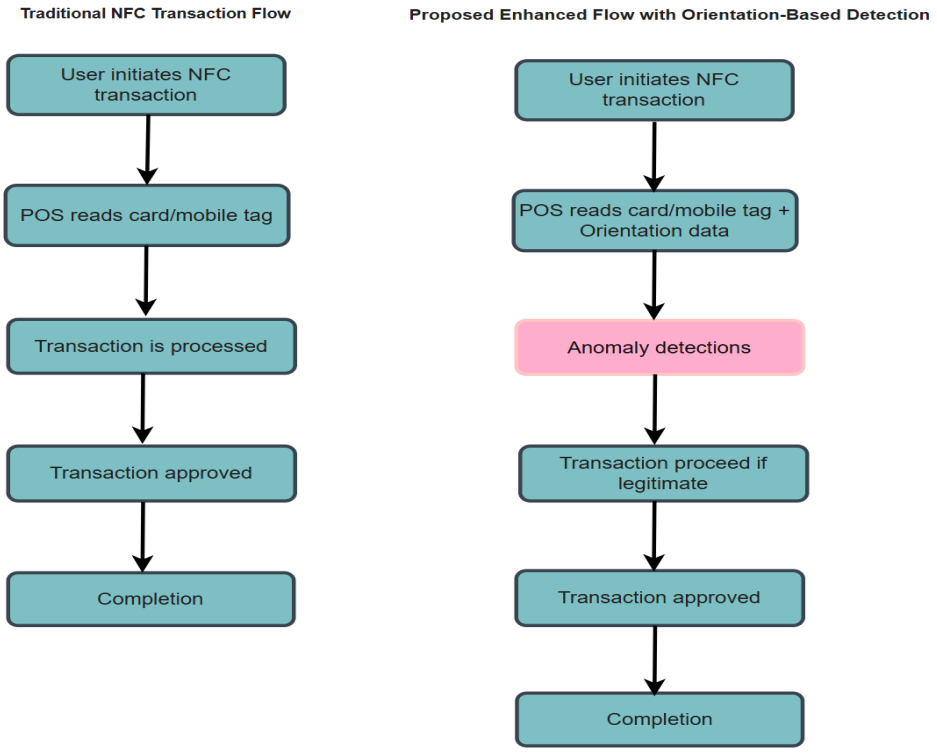
Through analysis of pitch, roll, and transaction timing features, the approach reliably distinguishes legitimate transactions from relay attacks. Experimental validation demonstrated significant improvements over conventional statistical thresholding, achieving a detection accuracy of 96.7% with minimal latency overhead (~285 ms average), thus preserving the real-time performance required in commercial settings.

The system's architecture is adaptable across smartphones, smart cards, and upgraded POS terminals, ensuring flexibility and ease of integration within existing NFC infrastructures. By combining motion verification with embedded machine learning, this work offers a scalable, efficient, and practical solution to evolving threats such as time-optimized and orientation-based relay attacks.

Overall, the findings lay a strong foundation for advancing the next generation of secure, behavior-aware contactless payment system

Appendix

Figure 1. This figure denotes how original NFC transactions are being improved with SensePay.



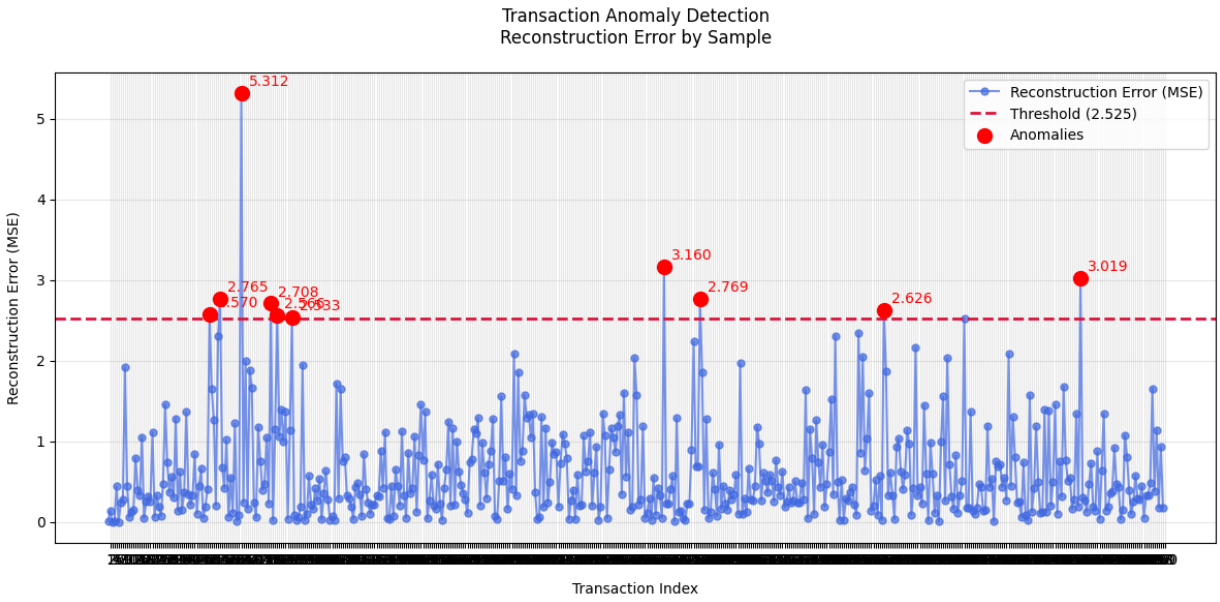


Figure 4. Reconstruction error threshold for Autoencoder

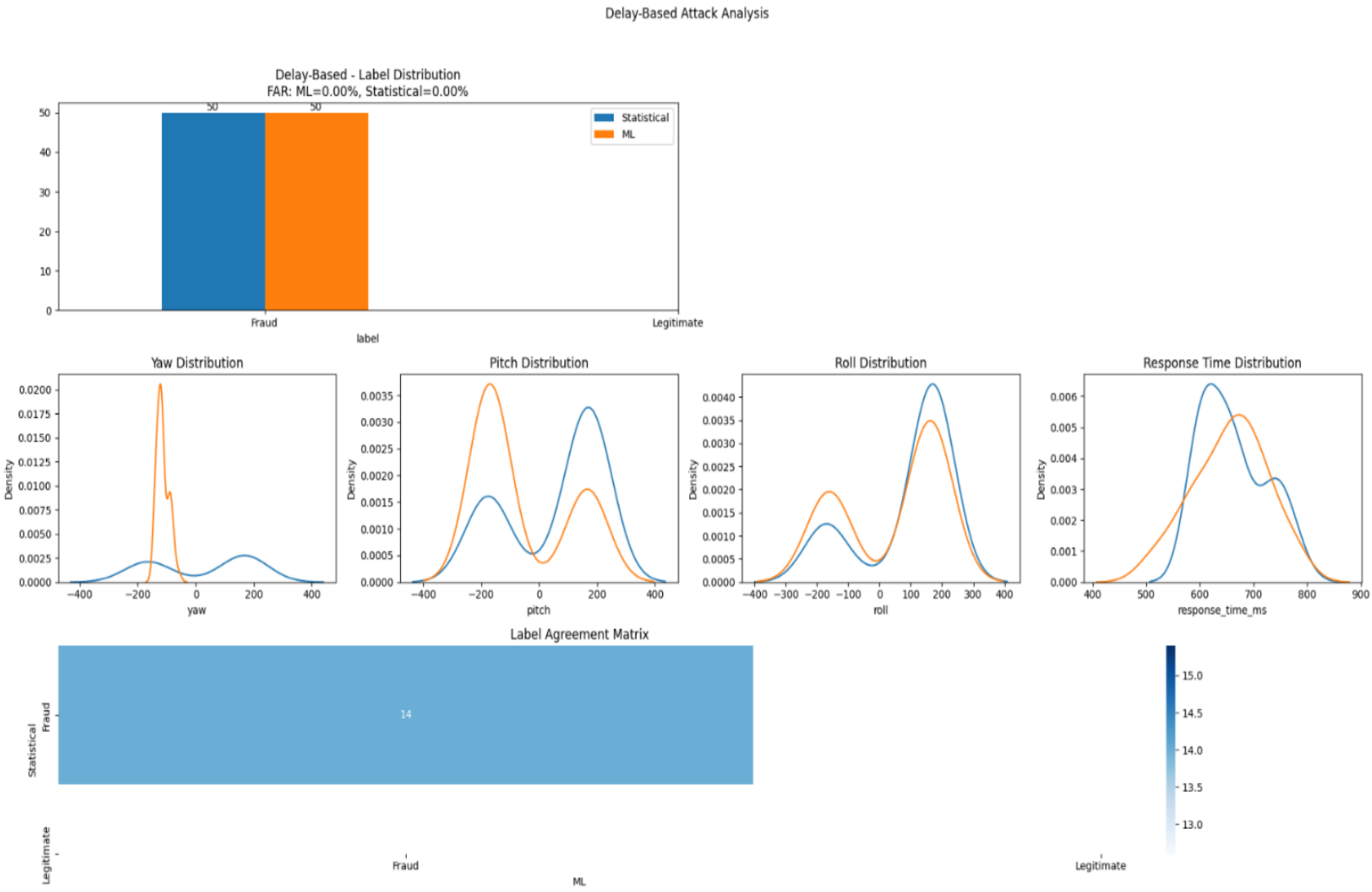
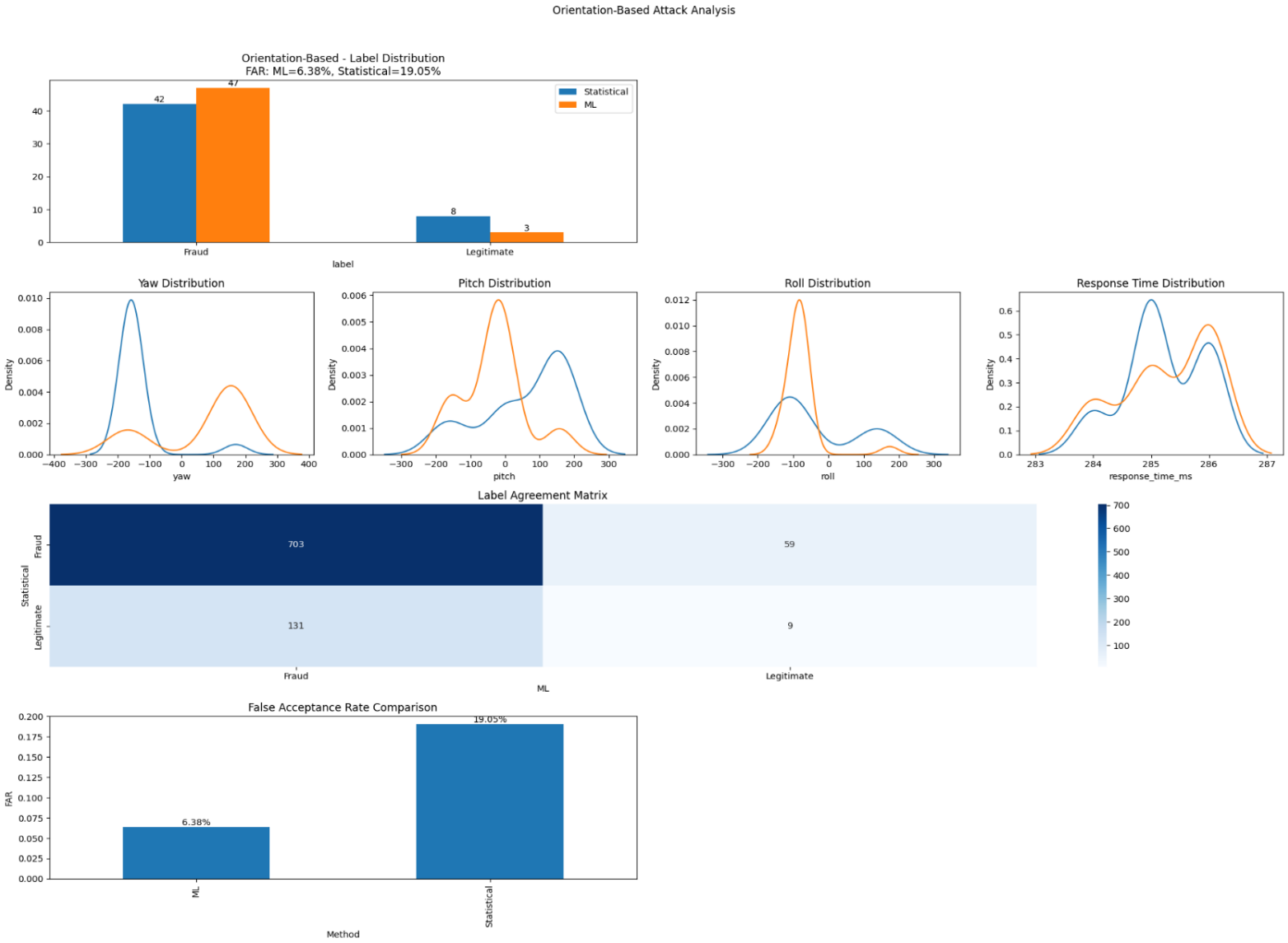


Figure 2. Delayed-based relay attack Analysis

Figure 3. Orientation-based relay attack analysis



Reference

[1] Juniper Research Jordan Rookes. *Contactless Payment Transactions to Hit $15.7 Trillion Globally by 2029, as Soft Point-of-Sale & Ticketing Rollouts Accelerate Growth*. Contactless Payment Transactions to Hit $15.7 Trillion Globally by 2029 (juniperresearch.com), June 2024.

[2] Luigi Sportiello (2019). *"Internet of Smart Cards": A pocket attacks scenario*

[3] G.P. Hancke, "A practical relay attack on ISO 14443 proximity cards", Technical Report, University of Cambridge, Computer Laboratory, UK, pp. 1–13, 2005.

[4] Daniel C. et Al.(2024) *PURE: Payments with UWB RElay-protection*

[5] Steven J. Murdoch & Ross Anderson.(nd). *Security Protocols and Evidence: Where Many Payment Systems Fail*

[6] P.S. Sharma, et. Al. (2020). *Ultra-Wideband Technology: Standards, Characteristics, Applications*

[7] TapTrack. "NFC RELAY ATTACKS." Retrieved from https://taptrack.com/nfc-relay-attacks/

[8] Drimer, S., & Murdoch, S.J. "Relay attacks on card payment: vulnerabilities and defences." 2007.

[9] Identity Management Institute®. "Relay Attack Risks and Prevention." Retrieved from https://identitymanagementinstitute.org/relay-attack-risks-and-prevention/

[10] ResearchGate. "A countermeasure against relay attack in NFC payment." Retrieved from https://www.researchgate.net/publication/323373346_A_countermeasure_against_relay_attack_in_NFC_payment

[11] Eprint IACR. "Another Look at Relay & Distance-based Attacks in..." 2018.

[12] CS Birmingham. "Protecting Contactless EMV cards from relay attacks." Retrieved from https://www.cs.bham.ac.uk/~tpc/Relay/

[13] ResearchGate. "A Practical Generic Relay Attack on Contactless Transactions by Using NFC Mobile Phones." Retrieved

from https://www.researchgate.net/publication/307688701_A_Practical_Generic_Relay_Attack_on_Contactless_Transactions_by_Using_NFC_Mobile_Phones

[14] Yu-Ju Tu & Selwyn P. (2020). *On addressing RFID/NFC-based relay attacks: An overview*

[15] Luigi S. (2019). *"Internet of Smart Cards": A pocket attacks scenario*

[16] Lishoy F. et. Al. (2005). *Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones*

[17] Manoj J. (2010). *Ultra-Wide Bandwidth*

[18] P.S. Sharma et. Al (2020). *Ultra-Wideband Technology: Standards, Characteristics, Applications*

[19] Fabian M. et al. (2024). *Performance Analysis of an Ultra-Wideband Transceiver for Real-Time Localization*

[20] Arwa A. (2020). *Security Issues in Near Field Communications (NFC)*

[21] Nour E, Emmanuel B. & Guy P. (n.d) *An Overview of the EMV Protocol and Its Security Vulnerabilities.*

[22] Nour E. & Guy P. (2016). *Security Enhancements in EMV Protocol for NFC Mobile Payment*

[23] Mahsid M. et al (2021). *OPay: an Orientation-based Contactless Payment Solution Against Passive Attacks*

[24] Frank S. et al (2012) *Multichannel protocols to prevent relay atttacks*

[25] Mul. I et. Al (2019) *Detection of Near Field Communication (NFC) Relay Attack Anomalies in Electronic Payment Cases using Markov Chain*

[26] "RF430CL330H: Harvested Power from a typical NFC-enabled smartphone of RF430CL330HTB using the onboard antenna," *Other Wireless Technologies Forum - Other Wireless - TI E2E Support Forums*. https://e2e.ti.com/support/wireless-connectivity/other-wireless-group/other-wireless/f/other-wireless-technologies-forum/665580/rf430cl330h-harvested-power-from-a-typical-nfc-enabled-smartphone-of-rf430cl330htb-using-the-onboard-antenna