Detecting Wireless Relay Attacks in NFC using Deep-Learning

Maria Jawah, Jana Bakhalqi, and Talah Fairaq.

talahfairaq@gmail.com
jan.saleh22@gmail.com
maryils661@gmail.com

***Abstract*— This study explores the application of deep learning (DL) to enhance security in Near Field Communication (NFC) technology, which is widely used in secure access control and contactless payments. As NFC usage grows, concerns have emerged about security vulnerabilities, particularly relay attacks, where attackers relay signals without need to break encryption or other protective measures. Previous research focused on ambient-based, distance-bounding protocols and deep-learning with RF fingerprinting via Wi-Fi to mitigate such threats. In this study, this method will be enhanced for detecting NFC relay attacks using RF fingerprints and deep learning via Bluetooth. The results demonstrate how DL can identify abnormal patterns associated with relay attacks, laying the groundwork for enhanced NFC security.**

***Keywords*—NFC, Relay Attack, Machine Learning, Wireless Communication**

I. INTRODUCTION

Modern communication systems are increasingly vulnerable to wireless relay attacks, particularly in the case of Near Field Communication (NFC) technologies, which allow smartphones and other devices to function as short-range contactless communication devices. By just relaying the communication signal without examining or altering it, these attacks take advantage of weaknesses in data transmission security between devices, enabling attackers to retransmit data in an unauthorized manner. By transmitting the signal between the two interacting entities even when they are far apart, the attacker deceives them into carrying out NFC transactions. Even in cases when the data is encrypted, it is still possible to accomplish this. Relay attacks involve forwarding the communications signals by the attacker without any kind of analysis or modification. While there are various of solution that has addressed this issue such ambient-based, deep-learning, and distance-bounding protocols. Meanwhile, the use of Bluetooth and cellular networks remains a challenge [1].

The primary aim of this research is to develop method for detecting NFC relay attacks combined with deep learning. The scope encompasses the design and implementation of experiments to collect and analyze NFC signal data for Bluetooth, creating a dataset of Bluetooth NFC relay attacks, and collecting signals in the natural state to train an accurate machine classification of transmitted and normal signals. This approach seeks to enhance the overall security of NFC applications.

The motivation behind this study stems from the increasing use of NFC technology in various applications, including mobile payments and access control systems, which heightens the need for robust security measures. Relay attacks, which can occur even when data is encrypted, represent a significant threat that traditional countermeasures fail to adequately address. The objective of this research is to fill this gap by employing deep learning techniques to automatically extract distinctive features from NFC signal waveforms, thereby offering an effective solution for identifying relay attacks and improving the security framework of NFC systems [1].

II. BACKGROUND

1. *Near Field Communication (NFC)*

Near Field Communication (NFC) is a short-range wireless communication technology within the Radio Frequency Identification (RFID) family. It is employed for data transmission between two devices when they are placed nearby, specifically within a range of up to 10 centimeters. NFC technology facilitates contactless data exchange, allowing for convenient user interactions such as touch-and-go payments and automated access control systems [2].

An NFC connection always involves two devices: one called the initiator and the other, the target. The initiator is the device that starts the interaction by sending a request, and the target responds to it.

NFC devices come in two types: active and passive. Active devices are powered by batteries, while passive devices are powered by the electromagnetic field generated during communication with an active device. A key difference between the two is that passive devices can only be targets for connection, while active devices can function both as targets and initiators. Smartphones with NFC support can be an example of an active device, and a NFC tag can be an example of a passive device [3].

Wireless Relay Attacks in NFC

NFC-enabled devices have different modes of operation, they typically function in three modes: card emulation mode, peer-to-peer mode, and reader-writer mode. In card emulation mode, an NFC device acts as a reader, like an NFC tag, capable of securely storing data for applications such as electronic ticketing and payments. Peer-to-peer mode allows two NFC-equipped devices to exchange data by direct contact, enabling activities like data transfer between devices and printing through touch. Reader-writer mode enables an NFC device to interact with tags similarly to RFID tags, reading and writing data on RFID chips used for tracking and identification through radio waves. NFC's core applications involve connecting electronic devices, accessing digital content, and facilitating contactless transactions [4].

Near Field Communication (NFC) is transforming the way we interact with technology by enabling seamless and secure communication between devices at close range. This innovative technology is not only enhancing user experiences but also streamlining processes across various sectors. From contactless payments to smart advertising, NFC applications are becoming increasingly prevalent in our daily lives. Below are some notable applications of NFC technology:

1) **Payment**
   NFC technology has significantly transformed payment systems by enabling quick and secure contactless transactions. With the rise of mobile wallets like Apple Pay and Google Pay, users can simply tap their NFC-enabled smartphones or cards at point-of-sale terminals to make payments [3].
2) **Transportation**
   Contactless tickets for transit and ticketing streamline the user experience, making it easier to access transportation services and controlled locations. They offer convenience, environmental benefits by reducing resource consumption, and improve system monitoring for greater transparency [5].
3) **Marketing and advertising**
   Accessing information and redeeming money-saving offers from smart posters becomes convenient. Product details can be shared via NFC-enabled tags on posters, allowing readers to access and read them anytime, anywhere. This offers two advantages: readers no longer have to stand in front of the poster, drawing attention, and information is not restricted by banner size or visibility. As a result, the amount and quality of information exchanged significantly improve [5].

Challenges and Limitations of NFC

1) **Range limitations**
   NFC's operating range is restricted to 10 cm due to inductive coupling, which is notably shorter compared to Bluetooth's 10 meters and Wi-Fi's 100 meters [5].
2) **Security risks**
   Given that NFC technology relies on radio waves for data transfer, various security threats can arise during transmissions or transactions, including Eavesdropping, Data corruption, Data modification, Imposter attacks (man in the middle), and Theft (NFC device is stolen) [5].
3) **Vulnerability to relay attacks**
   Attackers can exploit security protocols based on proximity by executing relay attacks, where they position two communication devices between the target reader and tag. This extension of range by the attackers can lead to significant security weaknesses, particularly in the realm of NFC-based transactions [5].

2. *Wireless communication vulnerabilities*

Wireless relay assaults in NFC systems are made possible in large part by Wi-Fi and Bluetooth. Because they make it easier for unauthorized parties to access protected communications, these wireless technologies are significant when it comes to NFC relay attacks. NFC relay attacks can be carried out in a number of methods, including by connecting wirelessly over Bluetooth and Wi-Fi [6].

The proposed radio frequency specification known as Bluetooth allows speech and data communication over short distances between several devices. Bluetooth technology has the potential to greatly simplify low-bandwidth wireless communications use [7]. It is a means of connecting and exchanging data and information across video games, laptops, digital cameras and mobile phones. [Jana 4] Bluetooth allows communication with other Bluetooth-enabled devices because it is a communication standard. Bluetooth is comparable to any other common communication protocol you might use, such as SMTP, HTTP, FTP, or IMAP. The client is the entity that initiates the communication using Bluetooth, and the server is the entity that receives the communication. Bluetooth operates on a client-server architecture [7].

Wireless communications are vulnerable to several types of attacks such as Replay attack, Eavesdropping, Man-in-the-Middle attack (MitM), including Relay attack. Relay attacks are a form of security threat that allows hackers to intercept and manipulate communication between two parties by relaying or transmitting the message through an intermediary device. Since a passive tag draws power from the reader and responds to it, an attacker can use a fake reader to start communication with a genuine tag. The attacker can then relay the information gathered to a forge tag at a distant location which can then communicate with the legitimate reader as a copy of the original tag [8].

It can be done even when the information is protected by cryptographic methods. In a relay attack, the attacker simply relays the communication signal without analyzing or modifying it. The attacker tricks two communicating entities into making NFC transactions by relaying the signal between the entities even when they are far apart. Although many RF systems implement data encryption to protect the transmitted information, the cryptographic standard is unable to handle a relay attack because the attacker does not need the encrypted information or attempt to decrypt it [9].

Relay assaults have become a serious hazard in the field of wireless communications and have already been used. The Analogue Relay Attack on the Physical Layer Applied to Bluetooth by Paul Staat et al. is an illustration of one of these attacks. Bluetooth technology, which is extensively utilized in many different applications, such as smart locks and car keyless entry systems, is vulnerable.

Where a hostile actor can establish a phony channel of communication between safe equipment. In order to get beyond security measures meant to prevent unwanted access, the article introduces a novel Analog Relay Attack on the Physical Layer that takes advantage of inexpensive radios to increase communication range and tamper with distance measurements. The authors show through extensive testing that relay attacks against Bluetooth-based access control systems are successful [10].

3. *Machine and Deep Learning*

Machine learning has revolutionized the field of cybersecurity by enabling systems to automatically detect, classify, and respond to attacks. ML models, especially in anomaly detection and classification, can analyze large datasets to identify patterns of malicious behavior that might be missed by traditional security systems. In the context of NFC and relay attacks, machine learning offers promising solutions by analyzing communication patterns, timing anomalies, and device behaviors to distinguish between legitimate interactions and potential attacks [3].

The subset of machine learning has been a game-changer in anomaly detection and cybersecurity. Unlike traditional machine learning, which relies heavily on manual feature engineering, DL models automatically extract high-level features from raw data, making them particularly effective for detecting complex patterns and subtle anomalies. In relay attack detection, DL models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) can capture intricate timing differences, data packet patterns, and signal variations that traditional systems may overlook. These models can analyze the continuous stream of NFC communications, identifying even the slightest deviations indicative of a relay attack [11].

III. CURRENT DETECTION METHODS OF RELAY ATTACK

1. DISTANCE-BOUNDING

Distance-bounding protocols work by measuring the round-trip time (RTT) of signals between two communicating entities such as a reader and a tag.

The idea is that if the RTT exceeds a certain threshold, it indicates that the communication may be compromised such as relayed by an attacker.

It involves a quick phase of bit exchanges where the reader transmits a single bit and begins a timer. The tag then replies with a bit that stops the timer. The reader calculates the propagation time based on the round-trip duration. After conducting a series of n rounds (where n is a security parameter), the reader determines if the tag is within a specified distance. To accurately measure the propagation time, the tag's processing time needs to be minimal and consistent [12].

2. AMBIENT BASED

Ambient-Based methods rely on environmental conditions such as temperature, humidity, light to verify the proximity of devices. If the conditions are similar, they may falsely indicate that two devices are close when they are not.

Ambient-based methods assume that if the environmental conditions around the NFC tag and the reader are similar, they are likely in close proximity and thus should be communicating directly without interference from an attacker.

Modern smartphones and tablets are equipped with an array of such sensors. A smartphone or payment terminal's physical surroundings might offer a variety of distinctive features that are specific to that place, such as the sound and lighting of a calm, well-lit space. Only a legitimate terminal and payment instrument pair are co-located using this information. Since the real terminal and payment instruments have different ambient environments, which should be inferred from their sensing readings, relay attacks should then be identified [13].

3. ISO/IEC 14443-A

Is an international standard for contactless smart cards, ISO/IEC 14443, covers the mechanisms for contactless interactions and associated transmission protocols for proximity integrated circuit cards (PICCs) and proximity coupling devices (PCDs). This standard specifies two types of communication interfaces: Type A and Type B. In this case, we concentrate on NFC-A, which is compatible with Type A of ISO/IEC 14443.

The 13.56 MHz carrier frequency is used by the NFC-A protocol. When the PCD is initializing and preventing collisions, it sends a request (REQA) instruction to see if a PICC (NFC tag) is within its radio frequency range. A type A request (ATQA) is answered by the PICC, moving the system from the IDLE to the READY state. When collision avoidance is effective, the PICC goes into the ACTIVE state and begins to receive application-specific signals from the top levels. The ATQA answer is an essential component for RF fingerprint extraction and is crucial in NFC transactions [6].

4. ISO/IEC 7816-4

The ISO/IEC 7816-4 standard is used as a relay attack detection method by leveraging time-based analysis at the application layer to distinguish between normal and suspicious NFC transactions. It involves the exchange of small, fixed-size Application Protocol Data Units (APDUs), which, under regular conditions, allow for stable round-trip times (RTTs) between the NFC device and reader. By establishing an upper limit on these RTT values and setting a standard deviation threshold, the system can detect abnormal delays introduced by relay attacks, where data relays significantly slow down the response. Operating at the application layer offers several advantages: it avoids stringent timing demands associated with lower-layer protocols (such as ISO/IEC 14443). This approach allows for practical relay attack detection, leveraging the standardized communication structures of ISO/IEC 7816-4 to identify and prevent security threats [14].

IV. Related Work

Symon J. (2018) proposed an anomaly detection system for detecting relay attacks in Bluetooth communications on Android devices. The authors suggest analyzing Bluetooth signal strength and response timing to monitor discrepancies in these parameters, aiming to identify unusual patterns that may indicate the presence of a relay device. The approach leverages machine learning algorithms for anomaly detection, balancing security with computational efficiency. Key detection methods include Timing Analysis, which relies on Round-Trip Time (RTT) to detect delays introduced by relay devices, as well as Signal Strength and Proximity analysis, which uses the Received Signal Strength Indicator (RSSI) to detect unusual proximity changes. Additionally, the authors discuss Cryptographic Measures, such as distance-bounding protocols that estimate the physical distance between devices by measuring response times to cryptographic challenges, helping to detect attacks that artificially shorten device distances. The primary goal of the machine learning model is to utilize sensors and radio signals on Android smartphones to detect unauthorized access attempts while operating within hardware constraints. However, the system’s effectiveness is limited by Android-specific permissions and can be bypassed by low-latency relay attacks, where delays are minimized and thus harder to detect [15].

Thorpe, C., Tobin, J., & Murphy, L. (2020) proposed an application-layer countermeasure for detecting relay attacks on NFC communication using the ISO/IEC 7816-4 protocol. This system leveraged round-trip time (RTT) measurements of APDU command-response pairs to detect relay attack-induced delays. The approach achieved a 100% detection rate with low false positive rates (0.38%–0.86%) across extensive testing. By setting an upper limit for RTT, transactions with excessive delay—indicative of relay attack interference—are identified and terminated. The authors implemented a relay attack on NFC devices using standard Android smartphones to demonstrate the feasibility and risks associated with these attacks. Their attack showed a successful transaction completion within 1-4 seconds, despite the extra delay introduced. So, by using the ISO/IEC 7816-4 protocol that measures RTT of specific APDU command-response pairs. If RTT exceeds a certain threshold, the transaction is flagged as a potential relay attack. However, it’s limited to NFC technologies using the ISO/IEC 7816-4 protocol and may struggle in environments with high network interference. Furthermore, relies heavily on the assumption that RTT delays indicate a relay attack, which may not cover cases where sophisticated attackers optimize delay times [14].

While distance bounding method is a security mechanism designed to prevent relay attacks in wireless communication, such as NFC (Near Field Communication). It works by measuring the round-trip time (RTT) of messages exchanged between two parties, allowing the initiator to estimate the distance to the responder. The protocol typically involves sending challenges, to which the responder replies with corresponding responses. By analyzing the RTT, the initiator can determine if the responder is within an acceptable range; if the response time exceeds a predetermined threshold, it indicates a potential relay attack or that the responder is too far away to be valid. This ensures that the communicating devices are genuinely close to each other, enhancing security in transactions and access control systems.

In Chong Hee Kim and Gildas Avoine study [12], they critique existing distance bounding protocols, noting that they typically use binary challenges and lack final signatures. This results in a high probability of success for adversaries, calculated as (3/4)^n where n is the number of rounds. The authors propose a new protocol that employs mixed challenges, both random and predefined challenges. This approach aims to reduce the adversary's success probability to the optimal bound of (1/2)^n.

The reader and tag exchange nonces and compute sequences that include both random and predefined challenges. The challenges are designed such that the tag can detect if an adversary is trying to preemptively obtain responses to challenges.

The paper considers scenarios where noise may affect communication, analyzing how the protocol holds up under such conditions. The proposed protocol improves upon previous protocols by effectively using mixed challenges, thus lowering the success probabilities for adversaries. However, Distance bounding protocols primarily depend on accurate time measurements. In noisy environments or with signal interference, this accuracy can diminish, potentially allowing relay attacks to go undetected. While mixed challenges improve security, the predefined nature of some challenges may still be predictable for a sophisticated adversary who could prepare responses in advance [12].

Konstantinos Markantonakis et al. [13] investigate how ambient sensors (such as accelerometers, gyroscopes, and environmental sensors) can be employed to detect anomalies in the physical context of NFC interactions. By monitoring variables like device movement and environmental conditions, it aims to identify when an attack might be occurring.

The researchers conducted experiments to evaluate the effectiveness of ambient sensor data in distinguishing between legitimate NFC transactions and those involving unauthorized relay attack. They employed various algorithms to analyze the sensor data and detect deviations from normal behavior.

The study involves conducting a series of experiments where ambient sensor data is collected during NFC transactions. The authors analyze this data to identify patterns and anomalies that could suggest a relay attack. They used three approaches to evaluate ambient sensors, their first approach was Similarity Analysis and Evaluation which focused on assessing the effectiveness of various ambient sensors in detecting relay attacks by analyzing the similarity of measurements from the payment terminal (PT) and payment instrument (PI). The method addressed the diversity in sensor data formats by using the Haversine formula for location measurements and applying Mean Absolute Error (MAE) and Correlation Coefficient for other sensors after linear interpolation to account for clock discrepancies. For three-dimensional sensors, vector magnitudes were calculated to simplify comparisons. A Python application was developed to compute the False Positive Rate (FPR) and False Negative Rate (FNR) by contrasting legitimate and unauthorized transaction measurements. The goal was to identify an optimal threshold for similarity metrics that minimizes errors while maximizing legitimate transaction acceptance, with the Equal Error Rate (EER) providing a critical point of balance. Additionally, the method evaluated transaction and sensor failure rates, revealing potential usability issues when quick device movements lead to missed data, thus highlighting the challenges in reliable NFC transaction security.

Their second approach was Machine Learning Analysis which enhanced the detection of relay attacks by applying supervised machine learning algorithms to sensor data. Unlike traditional similarity metrics that treat all measurements equally, this method recognizes that different time slots may hold varying importance for distinguishing between genuine and unauthorized transactions. Each pair of measurements from the payment terminal (PT) and payment instrument (PI) is treated as a labeled observation, with the absolute differences as features. The performance is evaluated using the Equal Error Rate (EER), determined through 10-fold stratified cross-validation to ensure reliability. Six machine learning algorithms, including Random Forest, Naive Bayes, and Support Vector Machines, were tested, revealing that while decision tree-based methods showed the best results—particularly with pressure sensor data—overall discrimination accuracy remained insufficient for practical use in NFC transaction security.

The third approach, which focuses on deep learning, is divided into two methods. Method 3 utilizes fully-connected artificial neural networks (ANNs) to enhance relay attack detection through simpler feed-forward architectures with input, hidden, and output layers. This method aims to improve performance by tuning hyperparameters, but despite showing advancements over traditional techniques, the overall accuracy remains insufficient for practical applications. In contrast, Method 4 employs convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to better handle time-series data and capture complex spatial and temporal dependencies. This method evaluates multiple architectures, including LSTM units to address the vanishing gradient problem in RNNs, resulting in marginally better performance than Method 3, although further refinement is still necessary for effective real-world use. However, the findings reveal significant challenges, while ambient sensors can capture valuable contextual information, they are not sufficiently responsive to detect relay attacks effectively under realistic timing constraints. The limitations stem from the inherent delays in sensor data processing and the rapid nature of NFC interactions, which can complicate the reliable detection of malicious activities in real-time [13].

The security risks associated with relay attacks in near-field communication (NFC) systems are discussed in this research By Wang, Y., Zou, J., & Zhang, K. (2023) [1], with a focus on radio-frequency identification (RFID) technology. Despite being widely employed in applications like keyless entry systems and mobile payments, NFC is susceptible to a number of assaults, including relay attacks, due to its open wireless nature. An attacker employs two devices, one of which is positioned close to an NFC tag and the other close to an NFC reader. In order to get around security measures like encryption at the application layer, the attacker can fool the reader into thinking it is interacting with the genuine tag by sending signals between these devices. Prior research has mostly concentrated on environment-based solutions or defenses to relay assaults on distance-detection algorithms. This document describes the protocols for contactless smart cards and their connection with readers. It concentrates on NFC-A technology, a common kind of NFC technology that complies with the ISO/IEC 14443A standard. The primary focus is on how an NFC tag (PICC) and an NFC reader (PCD) communicate. The PCD sends a request command (REQA) to detect the PICC as part of their communication, and the collision avoidance loop makes sure that several PICCs may interact without interfering with one another. Subsequently, the PCD chooses which PICC to contact.

The first state, IDLE, is one of the states that the PICC goes through as it awaits a request. ACTIVE, which initiates communication after completing the collision prevention procedure, and READY, which is triggered upon getting an ATQA answer. The PICC then responds with the ATQA answer (Reply to Request Type A), which is a crucial response as it shows that the PICC is prepared for communication and can serve as a special identification or "fingerprint" for RF data extraction. Applications such as tracking, identification, and security may benefit from the ability to extract radio frequency fingerprints from ATQA, a generic dataset. There are four components to this standard. It details the measurements and physical attributes of NFC cards, Radio Frequency Interface, The protocols for radio frequency communication, initialization, and collision prevention are described, along with the steps for initializing the card and managing several cards in the field. It also describes the commands and their corresponding responses during communication and suggests an NFC sequence detection method based on the radio frequency fingerprint of the wireless signals transmitted at the physical layer.

Using radio frequency fingerprinting, the research introduces a unique method for detecting NFC relay assaults. Using this technique, the special properties of electromagnetic signals sent during an NFC communication are examined.

The lack of a publicly available dataset for detecting NFC relay attacks has led to a dearth of research on countering relay attacks with radio frequency fingerprinting and deep learning techniques.

By establishing an SDR-based testbed, simulating relay attacks, gathering a large dataset of NFC signals, and classifying the signals using CNN, this all-encompassing method showed that deep learning approaches are both feasible and successful in identifying NFC relay assaults. Creating a testbed allowed for the creation of the NFC Relay attack detection dataset. A sniffer coil, an NFC reader, and a software-defined radio (SDR) platform were utilized to record NFC signal information. Four NFC tags from the same batch were utilized by the team, guaranteeing data consistency. Relay attack devices were developed in two varieties: Wired relay apparatus By sending messages across a physical link, this device mimics a direct relay attack. wireless relay apparatus In a more realistic assault situation, when physical connections do not limit distances, this configuration enables the signal to be sent over Wi-Fi. After gathering information from both relayed and conventional NFC signals, the scientists produced a varied dataset consisting of **66,366 samples**. **10 MSPS** was the high sample rate used to capture the signals in order to preserve the detailed features required for analysis. ATQA directives, which are necessary for recognizing NFC transactions, were present in parts in every sample. Wireless relay attacks wired relay attacks, and signals from standard NFC tags were all included in the dataset. Each signal sample is standardized to 1800 sampling points to guarantee completeness, and it includes conventional NFC signals, wired sent signals, and one wireless transmitted signal.

The dataset in the deep learning model was trained using a "deep convolutional neural network (CNN)". When it comes to evaluating radio frequency signals, CNNs are especially good at processing data that has a network-like structure, such time series signals or pictures.

In order to classify, they trained a convolutional neural network (CNN). A fully connected layer comes after three convolutional layers in the CNN. 72 size 8 kernels make up the first convolutional layer, 64 size 6 kernels make up the second, and 32 size 6 kernels make up the third. A softmax function was used to classify the data into many classes after each convolutional layer was followed by a ReLU activation function. With the Adam optimizer, the CNN was trained using cross-entropy as a loss function. The dataset was split into training (70%), validation (20%), and test (10%) sets. In order to monitor the validation loss and avoid overfitting, early halting was used. A deep neural network (DNN) and a support vector machine (SVM) were used to compare the CNN's performance. Measures including precision, accuracy, recall, and F1 score were computed to assess how well the model distinguished between transmitted and normal NFC signals. According to the findings of the experiment, the suggested RF fingerprinting technique can accurately and efficiently differentiate between transmitted and normal signals. According to this research, RF fingerprinting provides a feasible and workable defense against NFC relay attacks, significantly boosting the security of NFC communications.

When CNN was constructed and trained on their dataset, it successfully distinguished between fresh data samples in the experimental dataset, normal signals in the test set, and transmitted NFC signals. Furthermore, their deep learning approach eliminated the need for further human fingerprint feature extraction on the unprocessed data samples.

This study employs RF fingerprinting to detect NFC relay attacks and wireless devices. Device-specific characteristics retrieved from the wireless signals that radio frequency (RF) devices emit are known as

"RF fingerprinting." RF fingerprinting is a significant area of study in wireless signals that is now interacting with machine learning techniques. Since relay devices alter the signal waveform in their own unique ways, extracting RF fingerprint information from transmitted signals is a crucial step in our suggested approach.

As a result, their objective is to detect relay assaults by reviewing and recognizing sent radio frequency signals. Deep learning has demonstrated the ability to automatically generate RF fingerprints from unprocessed signal data in related research on RF fingerprinting. The research is based on data from a single wireless communication technique, namely WiFi. This limited emphasis restricts how far the findings may be applied. Other wireless technologies, such Bluetooth and cellular networks, should be investigated in future studies in order to document a wider variety of transmission patterns and vulnerabilities. In addition to increasing the dataset's diversity, this extension could shed light on the viability of assaults on other platforms. Radio frequency signature interpretation is not covered in detail in this work. To differentiate between harmful and authorized communications, it is essential to comprehend these signs. Future research should concentrate on creating techniques for interpreting and analyzing these signals, since this might strengthen the suggested solutions' resilience [6].

V. GAP ANALYSIS

Table 1 : Related Work Comparison

Ref	***Methods***	***Layer***	***Features***	***Datasets***	***Accuracy***	***Limitations***
[15]	Timing Analysis and Signal Strength Analysis Machine Learning (ML) for Anomaly Detection	Physical Layer	RSSI and Timing Anomaly Detection ML Integration enhances detection optimizing accuracy over time.	The data set used in Detecting Relay Attacks Against Bluetooth on Android contains **1,200** instances. subsets of 10% (121 instances) and 1% (12 instances) used to test the classifier's performance on smaller data samples.	Wi-Fi: up to 98.3%. Combined wireless signals (Bluetooth, Wi-Fi, Cell): up to 98.3%. Bluetooth alone: up to 86.7%.	Limited to Android OS. Restricted by Android permissions. Timing detection mechanisms insufficient against low-latency attacks. RSSI values can be influenced by environmental factors. Bluetooth signal is less reliable due to high variability and short range.
[14]	distance-bounding	Application Layer	100% detection accuracy Low false positive rate between 0.38% and 0.86%	**10,000** NFC transactions across a variety of contactless cards and payment terminals, recording the round-trip times (RTTs) for both uninterrupted and relayed transactions.	Demonstrated a 100% detection rate for relay attacks. The false positive rate was low, ranging from 0.38% to 0.86%. Added only a 0.22-second delay per transaction	Works with systems utilizing the ISO/IEC 7816-4 protocol. The method is mainly effective against common relay attack scenarios over Wi-Fi but may be less effective if relay methods introduce minimal RTT delay. Adds a slight delay (0.22 seconds) to transactions.
[12]	Distance-bounding Mixed Challenge protocol	Application layer	Utilizes mixed challenges to enhance security Measures round-trip time (RTT) to verify proximity Do not use final signature which is good because signatures increase computational and communication Error detection	N/A	Reduces adversary success probability to (1/2)^n	Adversary can still have a probability of 50% attack success
[13]	Ambient-Based solutions Similarity Analysis and Evaluation Machine Learning Analysis Deep Learning Analysis Using Fully-Connected Artificial Neural Networks (ANNS) and Sensor Combinations Deep Learning with CNN and RNN and Sensor Combinations	Application layer	MAE and Correlation for anomaly detection Classification (Logistic Regression, SVM) Pattern recognition Spatial pattern detection Temporal pattern detection	**1,000** transactions was recorded per sensor, some sensors had failure rates over 99%, and humidity/temperature sensors showed valid data in only 6% of transactions Overall, 17 sensors were utilized	The first method showed limited effectiveness in distinguishing genuine transactions from unauthorized ones. In the second method, the best-performing models reached around 10% EER In the third method, models achieved moderate accuracy but was outperformed by traditional machine learning models, and the more complex deep learning models In the forth method, the CNN and RNN architectures achieved the highest performance with EER of 0.246 (CNN) and 0.273 (RNN)	High error rates Limited accuracy, unsuitable for high-security use High computation cost, modest improvement Resource-intensive, not accurate enough for real-time use High resource demand, insufficient for time-critical tasks
[1]	Deep Learning with CNN RF Fingerprinting Dataset Creation Classification	Physical Layer	Real-time detect Comprehensive dataset High accuracy	66,366 NFC signal samples, comprising four types of wired relay attack signals, four types of conventional NFC tag signals, and one type of wireless relay attack signal implemented via Wi-Fi, were gathered using an SDR-based testbed to create a freshly produced dataset.	66,366 NFC signal samples, comprising four types of wired relay attack signals, four types of conventional NFC tag signals, and one type of wireless relay attack signal implemented via Wi-Fi, were gathered using an SDR-based testbed to create a freshly produced dataset.	Signal Quality Dependence Relying on data search only from Wi-Fi network

Table 2: Compare Our Solution Proposal Features Over Other Related Work

Ref	Physical Layer	Application layer	Distance-bounding method	Ambient-Based solutions	RF Fingerprinting	Machine Learning	Through Wi-Fi	Through Bluetooth
Our Research	✓				✓	✓	✓	✓
[15]	✓					✓	✓	✓
[14]		✓	✓				✓	
[12]		✓	✓					
[13]		✓		✓			✓	✓
[1]	✓				✓	✓	✓	

The Papers mentioned Above together cover various facets of wireless communication anomaly detection. In the Symon J. (2018) [15], machine learning is used to improve timing accuracy and signal anomaly detection, while the system's effectiveness is limited by Android-specific permissions and can be bypassed by low-latency relay attacks, where delays are minimized and thus harder to detect are highlighted.

The Thorpe, C., Tobin, J., & Murphy, L. (2020) [14] offers a reliable approach with a low false positive rate and 100% detection accuracy, however it is limited by its dependency on Wi-Fi data, and it's limited to NFC technologies using the ISO/IEC 7816-4 protocol and may struggle in environments with high network interference.

While In Wang, Y., Zou, J., & Zhang, K. (2023) [1], employs CNN and other deep learning algorithms for real-time anomaly identification, it encounters issues with sensor reliability and temporal restrictions.

The C. H. Kim and G. Avoin [12] proposed a mixed challenge protocol to enhance security more than previous methods used in distance-bounding. However, while the paper discusses the protocol's performance in both noise-free and noisy environment, in real-world RFID communication is often subject to various forms of interference. Also, The analysis of the adversary's success probabilities is based on specific assumptions about their strategies. If an adversary employs different tactics or has more sophisticated capabilities, the security guarantees provided by the protocol could be compromised.

The Konstantinos Markantonakis et al. [13] evaluates the use of ambient sensors on mobile devices to detect relay attacks in NFC transactions, testing 17 sensors across 1,000 transactions with various machine learning and deep learning techniques to assess their effectiveness under a 500ms timing constraint. However, the limitations include high error rates, insufficient accuracy for high-security applications, and significant computational demands, with even the best-performing models (CNN and RNN) failing to meet the precision and real-time requirements essential for secure NFC transactions.

To fill the gap our research, we will use RF Fingerprinting to collect datasets with use deep learning to detect the relay attacks. that provides more adaptability to evolving environments, identify a wider range of anomalies, real-time analysis and detection and accommodate a broader range of devices and configurations, enhancing detection capabilities and low false positive rate and high detection accuracy. Furthermore, RF fingerprinting leverages the unique characteristics of radio frequency signals emitted by devices. Each device produces a distinct signal profile based on its hardware and environment, which can be captured and analyzed. wherefore, can be more effective in distinguishing legitimate devices from attackers.

VI. PROPOSED SOLUTION AND METHODOLOGY

Inspired by the work in [1], we proposed a model using RF fingerprints to detect NCF relay attacks by collecting a dataset of normal NFC signals and wired and wireless relay attack signals wirelessly via Bluetooth. This is done by building a testbed based on software-defined radio (SDR), then manually classifying these signals into two datasets (normal and relay attack signal) and feeding the classified dataset into a CNN deep-learning model. A neural network is used in signal processing and extracting RF fingerprints from the collected signals to distinguish between normal signals and transmitted signals. After training, the CNN can classify new signals based on the extracted features, which helps to detect any tampering or relay attacks with high accuracy. Since there is no publicly accessible dataset, we will create one. Specifically, we will install an SDR testbed to acquire data and collect normal and relay signal data for NFC tags. The data acquisition testbed will include an SDR hardware platform, a sniffer, a tag reader, and two types of NFC relay devices, and a wired device and a wireless device will be designed and built to simulate NFC relay attacks to collect data samples for training the deep neural network. Extracting fingerprints helps in identifying RF devices after collecting the transmitted and normal NFC signals from both devices. Then, we will identify the parts that contain ATQA instructions. These parts will be extracted from the original signals and used as data samples, which will be used to train the CNN architecture.

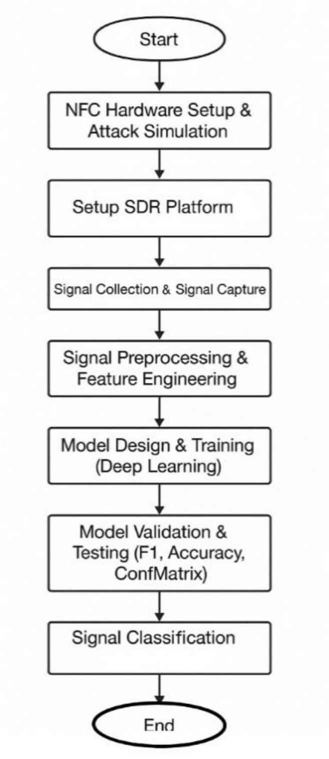

Start
NFC Hardware Setup & Attack Simulation
Setup SDR Platform
Signal Collection & Signal Capture
Signal Preprocessing & Feature Engineering
Model Design & Training (Deep Learning)
Model Validation & Testing (F1, Accuracy, ConfMatrix)
Signal Classification
End

Figure 1: Proposed Solution Workflow

NFC Hardware Setup & Attack Simulation

In our project, we created two environments. The first one is intended for a realistic NFC communication environment using the following devices: Near Field Communication (NFC) technology and a set of hardware that represent the true elements of the system. The setup includes a real (legitimate) NFC reader, a device capable of sending standard commands such as REQA and receiving responses such as ATQA from tags. We also used a real (legitimate) NFC tag, which is an actual tag that participates in traditional communication scenarios and responds to signals from the reader naturally without any external intervention. This setting is intended to simulate normal communication between the reader and the tag as a key stage before the relay attack is executed, allowing comparison between the original signals and those generated by the attack later in the project.

The second environment is intended for simulating a relay attack. The test environment is configured to replicate a realistic relay attack scenario using Near Field Communication (NFC). The hardware setup includes native NFC components along with wireless relay emulation devices. Specifically, the system consists of a real NFC reader, a real NFC tag, a reader emulator, a tag emulator, and a dedicated read-write NFC tag. Communication between the reader emulator and the tag emulator is handled wirelessly through ESP32 microcontrollers connected via Bluetooth. This setup enables the interception and relay of NFC communication, forming the basis for analyzing signal behaviors and vulnerabilities in relay attack scenarios.

Setup SDR Platform

The SDR platform captures NFC signals and inspects them using deep learning and RF fingerprinting to detect relay attacks and determine their identities. The SDR platform is a flexible and programmable communications system that operates and analyzes wireless signals using software. There is more than one SDR. Because of the large frequency range and high dynamic range, we used a HackRF One software defined radio (SDR) in this project. For allowing the SDR to be in resonance with ISO 14443 NFC systems, a well-constructed near-field loop antenna at 13.56 MHz frequency was used. To achieve optimal inductive coupling, the antenna was positioned close to the reader-tag system and was impedance matched. All captures were done in a low-noise condition to reduce EMI and maintain signal integrity.

The comparison of Software-Defined Radios (SDRs)—AirSPY, HackRF, and USRP (Universal Software Radio Peripheral)—reveals distinct strengths and applications. While AirSPY is recognized for its high sampling rate of 10 MSPS and its precision-focused 12-bit ADC, making it an excellent choice for narrowband tasks, such as NFC relay attack detection via direct RF monitoring, its narrow frequency range (24 MHz to 1.7 GHz) limits its applicability to broader or higher-bandwidth applications like Bluetooth [16].

In contrast, HackRF emerges as a versatile and lower budget ideal for detecting NFC relay attacks over Bluetooth due to its broader frequency range (1 - 6 GHz) and its capability to operate in the 2.4 GHz spectrum used by Bluetooth communication [17]. With a sampling rate of up to 20 MSPS and support for open-source tools like GNU Radio and gr-Bluetooth, HackRF can effectively capture, modify, and relay Bluetooth packets, which are often integral to emulating and manipulating NFC communications routed through Bluetooth relays [18]. This versatility makes HackRF a preferred choice, although its 8-bit offers slightly lower resolutions compared to AirSPY.

On the other hand, USRP is still the most flexible and advanced SDR platform. It supports higher sampling rates (up to 50 MSPS) and full transmission capabilities [18]. While it excels in complex, multi-domain, protocol-intensive applications, such as 5G or cognitive radio, its high cost and advanced configuration requirements may be excessive for detecting NFC relay attacks over Bluetooth, while HackRF offers a practical and cost-effective solution. Ultimately, HackRF bridges the gap between cost and functionality, making it a more versatile option for applications that require emulating Bluetooth-based NFC attacks. Together, these findings underscore that the choice of SDR depends heavily on the required frequency range, budget, and application needs.

Table 3: Compare SDR Devices

SDR Type	Sampling Rate	Flexibility	Accuracy	Cost-Effectiveness	Broad-Frequency Applications
AirSPY	Lower	Lower	High	Lower	Lower
HackRF	High	High	Lower	Lower	High
USRP	High	High	High	High	High

Signal Collection

We perform structured signal acquisition for both normal NFC communication and relay attack. The setup involves establishing two types of NFC interactions:
1. Normal Communication: Straightforward interaction between a legitimate NFC reader and genuine NFC tag without any interference.
2. Relay Attack Communication: A replay attack in which the NFC reader communicates to the tag indirectly through a relay channel to mimic a real-life relay attack.

To capture these interactions, we employ a Software Defined Radio (SDR) device, the HackRF One, and the SDR++ software platform. The HackRF One is configured to listen on the 13.56 MHz carrier frequency used by ISO/IEC 14443-A NFC systems. For a given test, the NFC tag is placed in the read area to offer consistent transmission, and the resulting electromagnetic signals are intercepted and recorded by the SDR.

Several recordings are captured for each category of communication (normal and relay) to ensure dataset diversity and RF fingerprint and environmental condition variations. These recordings form the basis of our raw dataset, which is then processed and analyzed for attack detection and classification.

Signal Preprocessing and Feature Engineering

Raw NFC signals are represented in spectrograms according to the Mel metric that is centered around presenting frequency energy over time with a focus on an exact time-frequency map as a physical property representation of the signal. This representation is used in feeding a deep learning model, with each spectrogram being viewed as an image that has a unique frequency signature (RF fingerprint) per signal.

Relay attacks normally render physical layer changes undetectable, such as power surges, temporal distortion, or signal delay; thus, the feature extraction process solely relies on learning these patterns from a well-engineered CNN model. These conventional surface features (e.g., standard deviation or mean) are not used; instead, the model is permitted to learn higher-level features such as local frequency structure, abrupt power transitions, and spectral transition patterns, which preserve the subtle differences between normal and attack-induced signals.
This is achieved through advanced signal processing, data expansion, and dimensionality normalization algorithms that provide the neural network with the best inputs, enabling an accurate frequency fingerprint (RF fingerprint) to be automatically and efficiently extracted, thereby improving classification accuracy and reducing overfitting and underfitting hazards.

Model Design and Training (Deep Learning)

In this stage, a convolutional neural network (CNN) is carefully constructed and trained to recognize NFC signals based on their spectrogram representations. The model architecture includes multiple convolutional layers and pooling and dense layers so that low-level and high-level RF fingerprinting features can be automatically extracted from the input spectrograms. They contain temporal energy patterns, frequency patterns of distribution, and fine-grained distortions caused by relay attacks that elude traditional methods of signal analysis.
These spectrogram images derived in the above step serve as the input to the network, which are resized and normalized for uniformity and improved convergence. Categorical cross-entropy loss and the Adam optimizer are used to train the CNN, with early stopping and dropout regularization included to prevent overfitting and ensure generalization.

Class balancing techniques are applied to balance any class imbalance between relay attack samples and normal samples. The model is trained on 70% of the dataset, and the remaining 30% is split equally into validation and test sets. During training, the model learns to recognize genuine NFC communications from relay attack manipulated communications using the learned RF fingerprinting patterns embedded in the spectrograms.

The final trained model is the core component of the classification pipeline that is extremely accurate and robust with domain-specific feature learning independent of manual feature engineering.

Model Validation and Testing

After we have trained a deep learning model, we subject it to a rigorous validation and testing process to check its generalization ability and classification accuracy. We split the dataset in a manner that 15% of the spectrograms are reserved for validation during training and the other 15% for ultimate testing. These two sets are completely different from the training data for the purpose of unbiased testing.
After training, the model is evaluated on the test set using some critical metrics such as accuracy, precision, recall, and F1 score, which give a good representation of the performance on both the classes (normal attack and sequence attack).
To further illustrate the model's behavior, we created a confusion matrix to graphically display the number of correct and incorrect classifications by category. The matrix illustrates how well the model separates legitimate NFC communications from relay attacks, reflecting any tendency toward

misclassification. Together, these numbers provide a general measure of performance and future deployment in real-world NFC security scenarios.

Signal Classification

After data collection and preprocessing, all the recorded NFC signals are classified into two classes: normal communication or relay attack. This depends on the origin and nature of the NFC transaction. Normal signals are true interactions between a legitimate NFC tag and a reader in secure environments, while relay attack signals are generated from sessions where an attacker has added a relay mechanism to intercept or divert the communication, usually resulting in slight alterations in the signal characteristics like increased power levels, timing distortions, or radio frequency distortions.

The classification process is essential to build a supervised deep learning model. By the allocation of an explicit label to each spectrogram image, the dataset provides the ground truth required to allow the CNN to learn the distinguishing features of benign and malicious NFC activity. By this supervised classification, the model is able not only to recognize patterns within the training set but also to generalize to novel, unseen signals and detect relay attacks with high reliability.

VII. Experimental Setup

1. Inventory of devices

Here, we describe the hardware components used to collect signals of normal NFC communication and simulate relay attacks. The setup includes three categories of devices: PCDs (readers), PICCs (tags/cards), and other components that support signal capture and transmission for simulate relay attacks.

A. PCDs (Readers)

Table 4 lists the different types of PCD devices used in the experiment. Reader 1, a smartphone (Hawaii – Android), used to read the card and tags in SDR capture, as shown in Figure 7. Also, reader 2, based on the RC522 module connected to an Arduino simulator, used to read the tags in the main electronic circuit and to ensure the success of the tag copying process in relay attack, as shown in Figure 3. An additional, the reader 3 using RC522 module, it acts as a transmission device read the main card data and then send it to the copier device, as shown in Figure 5.

Table 4: Inventory of PCDs devices

Name	Type	Model	Function
Reader 1	Smartphone	Hawaii - Android	To read the tags in SDR capture
Reader 2	Arduino Simulator	RC522	To read the tags in the main electronic circuit and to ensure the success of the tag copying process in relay attack
Reader 3	Arduino Simulator	RC522	Transmission device read the main card data and then send it to the copier device

Indeed, only one reader will be used to elaborate the final dataset it is Reader 1(Hawaii – Android). That because this decision is driven by the fact that Reader 2, which relies on the RC522 module connected to an Arduino board, continuously read signals as long as it is power on. This constant activity causes it to capture noise and unrelated signals. In other hand, the smartphone provides more controlled behavior; its NFC reader remains inactive by default and only activates when an NFC card is brought into close. This design reduces the noise and ensures more reliable capture of tag and card signals during SDR-based signal recording.

B. PICCs (Tags)

Table 5 presents the list of NFC tags and cards using. A total of four cards are used, all compliant with NFC-A (ISO 14443-A) standard. Cards 1 and 2 are read-only cards used to collect normal NFC signals. Whereas tags 1 and 2 are support write operations and they based on MIFARE Classic 1k card 1 and 2 chips for perform simulate relay attacks. Especially, used to copy UID and signal of card 1 and 2. These cards and tags are depicted in Figure 2.

C. Supporting Devices

Table 6 provides a detailed overview of the remaining hardware used in the experiment. The HackRF One software-defined radio (SDR) was employed to capture raw NFC signals for later analysis, these depicted in Figure 6. Two ESP32-based microcontrollers acted as Bluetooth transceivers to forward NFC data wirelessly, these are depicted in Figure 4 and 5. Reader RC522 is copying device, It takes the information it receives from the transmission device and then sends it to the tag that accepts writing on it, these depicted in Figure 4. Finally, an LCM1602 IIC LCD display module was used to display captured cards and tags IDs, which helps in visual confirmation so that we can be sure that the copy was successful, these depicted in Figure 3.

Each of these components played a critical role in either capturing, transmitting, or validating the NFC communication normal and simulate relay attacks.

Table 5: Inventory of PICC devices

Device No	Name	NFC type	Standard Chip	Chip	Writable
1	Card 1	NFC-A	ISO 14443-A	-	No
2	Card 2	NFC-A	ISO 14443-A	-	No
3	Tag 1	NFC-A	ISO 14443-A	Mifare Classic 1k	Yes
4	Tag 2	NFC-A	ISO 14443-A	Mifare Classic 1k	Yes

Table 6: Inventory of other devices

Device No.	Name	Type	Model	Function
1	SDR	HackRF One	-	Capture the NFC signals
2	Microcontroller	Arduino Simulator	ESP32	Bluetooth device to transmit signals between the transmitter and the copying device via Bluetooth and perform relay attack
3	Microcontroller	Arduino Simulator	ESP32	Bluetooth device to transmit signals between the transmitter and the copying device via Bluetooth and perform relay attack
4	Reader	Arduino Simulator	RC522	Copying device, It takes the information it receives from the transmission device and then sends it to the tag that accepts writing on it.
5	Display Screen	LCM1602 IIC	-	Shows the card ID and displays it on the screen

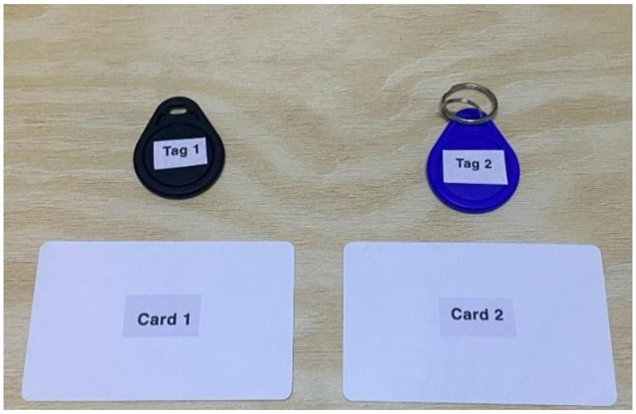


Figure 2: NFC tags

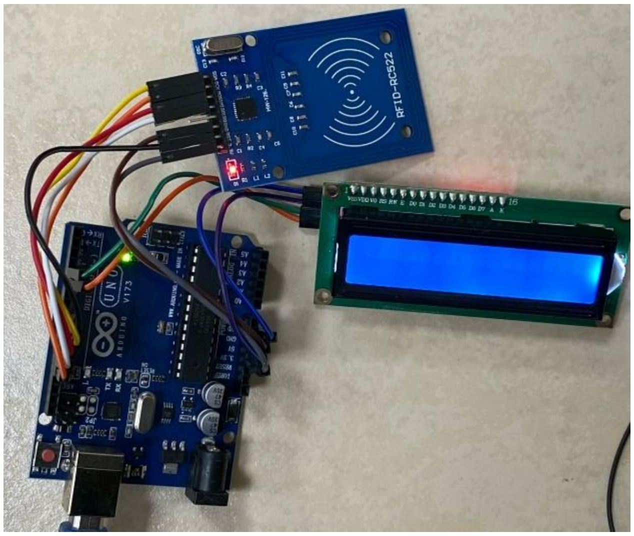


Figure 3: Reader in the main electronic circuit, Arduino Uno board and Display Screen.

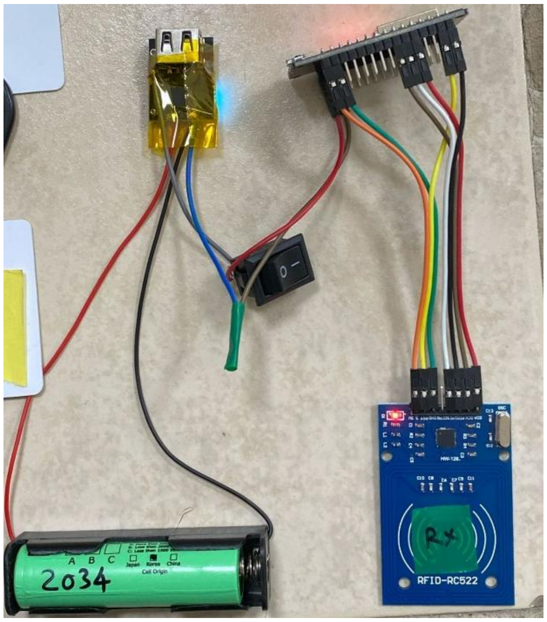


Figure 4: The reader is a transmission device

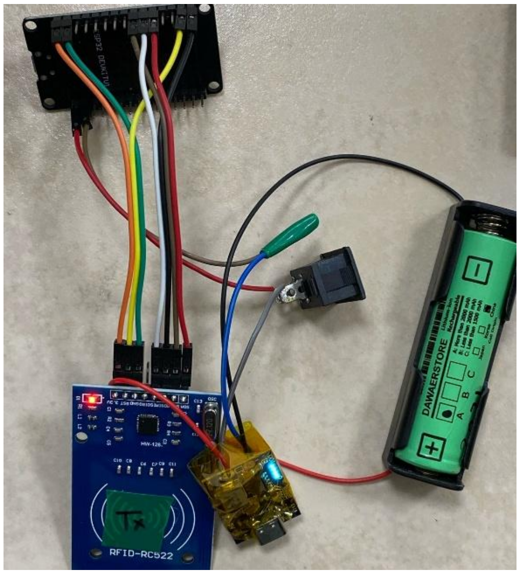


Figure 5: The reader is a Copying device

Figure 6: The SDR HackRF

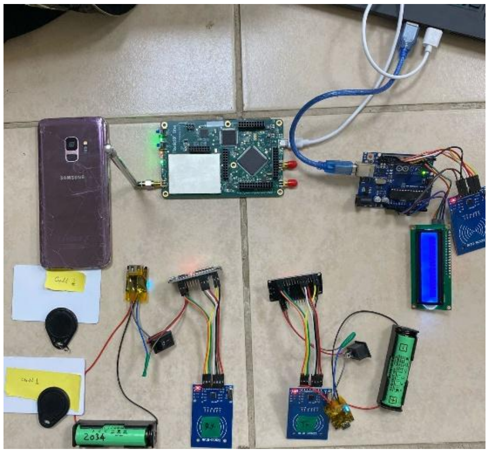


Figure 7: All component to implement detect Relay Attack

2. BUILDING SETUP

2.1 Normal Setup

For normal data collection of real NFC tag interactions, a simple setup consisting of an NFC reader, an NFC tag, an Arduino Uno board, and an LCD display module was established.

The NFC reader (RC522) was connected to the Arduino Uno board through the SPI interface. The Arduino was programmed to initialize the NFC reader, detect nearby NFC tags, and retrieve the unique identifier (UID) from that tag. Once the detection of the tag was complete, the UID would be printed to the serial monitor through the Arduino IDE as well as displayed onto the LCD screen interfaced to the board through I2C.
With the help of this setup, tag detection events were monitored and verified in real-time.

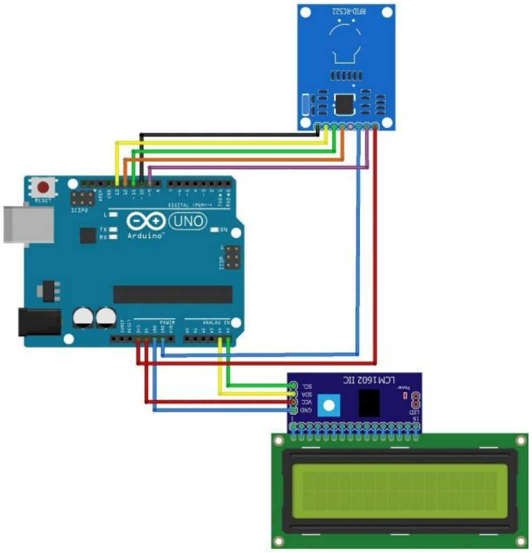


Figure 8: Diagram of Normal setup

2.2 Relay Attack Setup

This experimental environment has been set up to mimic a real-world scenario of an NFC relay attack. The relay system consisted of a combination of real NFC equipment and wireless relay emulation devices that comprise a real NFC reader, a real NFC tag, a reader emulator, a tag emulator, and a specialized read-write tag.

Under a Bluetooth wireless connection using ESP32 microcontrollers on both ends, the reader emulator was joined to the tag emulator. The reader emulator was near the real reader, allowing interception of the REQA command sent out by the real reader. The tag emulator was also placed near a read-write NFC tag, which was used to allow communication between the real tag and the emulator.

During the time the real tag tries to communicate with the real reader, the reader emulator captures that interaction and transmits the resulting signal wirelessly via Bluetooth to the tag emulator. Then, a read-write tag, which is located next to the tag emulator, plays double duty; that is, it captures the relayed ATQA response that was sent wirelessly by the reader emulator from the original tag and writes that into the tag emulator memory. In effect, this takes the tag emulator to the state of a real tag and responds to the reader like it was actually there.

One major differentiating trait between a real tag and a tag emulator is power emission. Real NFC tags operate passively and raise minimal energy from the electromagnetic field of the reader, whereas the tag emulator actively utilizes electronics that tend to yield a much higher emission-level signal, making the RF fingerprint of the tag identifiable.

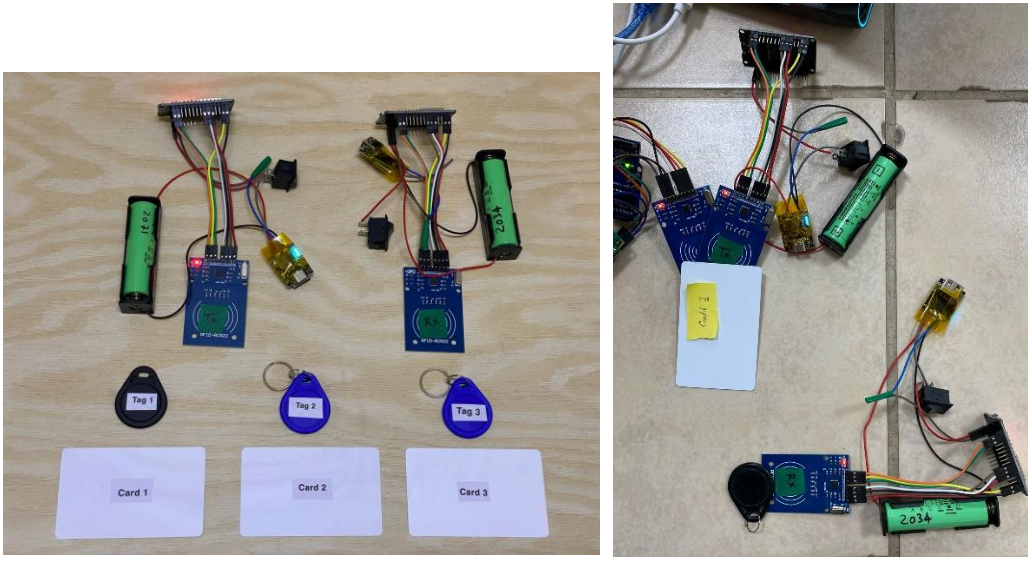


Figure 9: Relay attack setup

VIII. Dataset creation

1. Radio setup

The SDR and Antennae

For acquiring NFC signals, we use the HackRF One Software Defined Radio (SDR) due to its broader frequency range, its capability to operate in the 2.4 GHz spectrum used by Bluetooth communication, and a sampling rate of up to 20 MSPS[17]. The SDR is paired with a custom-fabricated near-field loop antenna optimized for 13.56 MHz to ensure resonance with ISO 14443 NFC systems. The antenna is impedance-matched and placed close to the reader-tag system to maximize inductive coupling

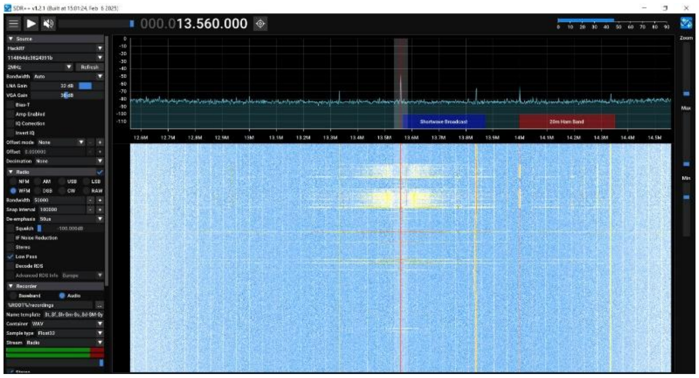
2. Software use

A. SDR++

The first part of the analysis phase is conducted using the SDR++ application. We use it to learn about SDR in general and to make our first recordings and real-time signal monitoring at 13.56 MHz. In that regard, it is really very useful; this software provides an easy-to-use GUI for visually aligning the NFC tag and reader. With a configured sampling rate of 2 MSPS, SDR++ enables precise tuning and calibration of the radio before starting data acquisition. However, our configuration needs have a set of shortcomings that we cannot accommodate. Specifically, we struggle to obtain a clear ATQA signal. This limitation stems from the application's lack of support for more advanced options, preventing us from setting a higher sample rate. As a result, we are unable to properly identify the ATQA, which is quite frustrating.

We do not realize these constraints until later in the process, after several attempts to test signals and confirm a genuine connection. This delay in discovery hinders our progress and adds to our challenges in effectively utilizing the SDR++ application.

Despite these setbacks, we are able to record communications well enough to capture the minimum ATQA from the reader's transmissions and the card response, as described in section figure 10. These are the reasons why we replace the SDR++ app with the GNU Radio application.



SDR++ v1.2.1 (Built at 15:01:24, Feb 6 2025)

000.013.560.000

Source

HackRF

114864dc3624391b

2MHz Refresh

Bandwidth Auto

LNA Gain 32 dB

VGA Gain 30 dB

Bias-T

Amp Enabled

IQ Correction

Invert IQ

Offset mode None

Offset 0.000000

Decimation None

Radio

○ NFM ○ AM ○ USB ○ LSB

● WFM ○ DSB ○ CW ○ RAW

Bandwidth 50000

Snap Interval 100000

De-emphasis 50us

Squelch -100.000dB

IF Noise Reduction

Stereo

✓ Low Pass

Decode RDS

Advanced RDS Info Europe

Recorder

○ Baseband ● Audio

%ROOT%/recordings

Name template $t_$f_$h-$m-$s_$d-$M-$y

Container WAV

Sample type Float32

Stream Radio

Figure 10: Capture signals from SDR++ platform

B. GNU Radio

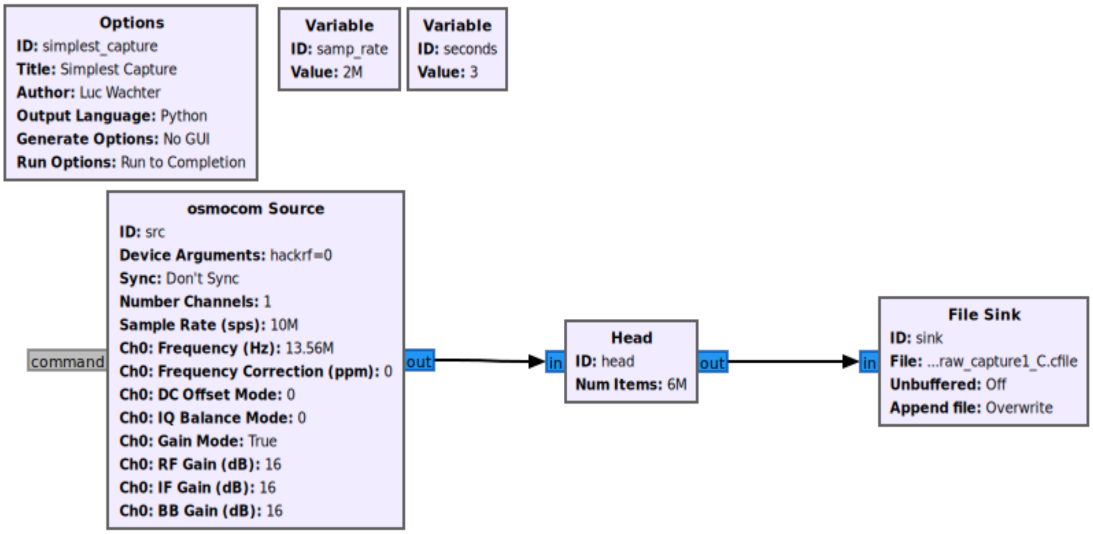
GNU Radio is an open-source toolkit that provides signal processing blocks to implement software-defined radios (SDRs) and signal processing systems. It is particularly useful for working with various radio protocols, including Near Field Communication (NFC). Although GRC (the GNU Radio Companion) uses the word radio, it is actually a graphical tool for building DSP applications by dragging blocks around on a computer screen. Furthermore, GNU Radio provides tools for visualizing the signal's spectrum, constellations, and other characteristics. This is crucial for debugging and ensuring proper signal reception and processing [20]. The main acquisition flowgraph is built with the Osmocom source block configured for HackRF One. It captures NFC exchanges in complex (IQ) baseband format (.cfile), preserving raw waveform characteristics for post-processing.

Acquisition

The acquisition script is used to facilitate the acquisition process. We create a very simple acquisition script based on a script generated by GNU Radio Companion. The goal is to make it as easy as possible to start recording with a selection of parameters and to record for a set amount of time (a set number of samples to be precise). To do this, we use the Osmosdr source block to connect to our HackRF One device and the Head block to set a fixed number of samples until the script stops. The script is written for the HackRF One. The sampling rate, the center frequency, and the capture length are configured. A path for the output file can also be specified.

We use "NFC-simplest-capture" for all the captured data. It is a very versatile tool, allowing us to define software pipelines using a block interface to create flow graphs. As it compiles to Python, the idea is to use it as a base for acquisition and processing scripts.

Figure 11 shows a GNU Radio Companion (GRC) flowgraph designed to capture raw NFC signals using a HackRF One software-defined radio (SDR). The flowgraph begins with an osmocom Source block configured to interface with the HackRF device (hackrf=0), capturing signals centered at 13.56 MHz, the standard frequency for NFC communication. The sampling rate is set to (10 MSPS), and the gain settings for RF, IF, and BB are each configured to 16 dB to ensure adequate signal amplification. A Head block follows, limiting the number of collected samples, which corresponds to a 3-second capture window. Finally, the captured samples are written to a binary .cfile using the File Sink block. This file can later be analyzed or processed for further signal analysis. The flowgraph is simple but effective, ensuring raw I/Q data is reliably captured and stored for processing. NFC interactions between tag and reader are captured, after that it enabling analysis of responses like ATQA.



Options
ID: simplest_capture
Title: Simplest Capture
Author: Luc Wachter
Output Language: Python
Generate Options: No GUI
Run Options: Run to Completion

Variable
ID: samp_rate
Value: 2M

Variable
ID: seconds
Value: 3

osmocom Source
ID: src
Device Arguments: hackrf=0
Sync: Don't Sync
Number Channels: 1
Sample Rate (sps): 10M
Ch0: Frequency (Hz): 13.56M
Ch0: Frequency Correction (ppm): 0
Ch0: DC Offset Mode: 0
Ch0: IQ Balance Mode: 0
Ch0: Gain Mode: True
Ch0: RF Gain (dB): 16
Ch0: IF Gain (dB): 16
Ch0: BB Gain (dB): 16

Head
ID: head
Num Items: 6M

File Sink
ID: sink
File: ...raw_capture1_C.cfile
Unbuffered: Off
Append file: Overwrite

Figure 11: GNU Radio Companion (GRC) flowgraph used for capturing raw radio signals with a HackRF One device.

C. Features extraction

1) ATQA

The Answer to Request Type A (ATQA) is a standardized NFC response signal defined in the ISO/IEC 14443-A protocol. It is the first message a passive NFC tag sends to an NFC reader after receiving a REQA (Request Type A) command. This message serves to indicate the tag's presence and readiness to communicate. It contains key information about the tag, such as its compliance type, configuration, and supported communication speed [21].

In our research, ATQA serves as the critical point of identification for each NFC signal. Because each tag, based on its hardware characteristics and manufacturing variability, exhibits subtle signal differences even when transmitting the same ATQA content, these responses are suitable for use in RF fingerprinting. By isolating and analyzing ATQA segments, we extract highly unique features that differentiate between tags and between normal communication and relay attacks. To extract ATQA:

- We use high-sample-rate SDR recordings (10 MSPS) of NFC communications.
- We apply energy-based windowing to locate high-energy short bursts (~50us to 1ms), which typically correspond to ATQA.
- Each detected ATQA segment is saved as an independent WAV file with normalized duration (512 samples).
- RF fingerprinting features such as energy, RMS, peak, standard deviation, and duration are computed for each segment.

This approach provides the foundation for building a dataset rich in physical-layer device characteristics, making ATQA a reliable signature.

2) Validating the dataset - Measure library (urh)

In this section, we explain how segments potentially containing ATQA responses are extract, process, and validate through decode and correlation against known ATQA patterns (0x0400) based on ISO/IEC 14443A. Dataset validation is a multi-step process designed to ensure the integrity, balance, and discriminability of the data before training deep learning models. Our validation process includes both automated techniques and manual inspections:

A. Manual Label Consistency Check-Byte-level Signal Validation with URH.

All extracted segments are traced back to their original source WAV files (e.g., C1.wav = Normal, T1.wav = Attack). We import the original NFC recordings into Universal Radio Hacker (URH), a tool designed for signal analysis and protocol reverse engineering. Within URH, we manually mark the suspected ATQA segments using the waveform view and the decode timeline. Using the demodulation feature On-Off Keying (OOK), we convert the baseband waveforms into bit-level symbols. These bits are converted into byte streams and displayed in hexadecimal format. We manually verify the hex values against ATQA responses (0x0400 for MIFARE Classic 1K) to ensure the segments correspond to actual ATQA fields. This method provides signal-level confidence that our extract segments are correct ATQA messages, not RF noise or unrelated commands.

B) Segment Extraction Using Energy Envelope - Signature Matching

We plot waveforms for randomly selected ATQA segments. Segment Extraction from Raw IQ Signal. To begin the process, a raw IQ capture file (NFC_raw_capture1_C.cfile) is loaded. The IQ data is analyzed to compute the signal amplitude envelope. That is a representation of the instantaneous energy in the signal. A peak detection algorithm is then applied to the envelope to locate regions of high energy, which typically correspond to bursts of NFC communication precisely ATQA responses. We use the five strongest peaks that extract individual signal segments, each potentially containing an ATQA response or another form of NFC card communication. These are saved as separate WAV files, which typically correspond to bursts of NFC communication — including ATQA responses. The graphs shown in Figure 12 (Real + Imag parts) represent the extracted signal segments, where sudden increases in signal strength indicate potential NFC tag responses.

C) Byte Recovery

With use Measure library (urh) we recovered byte then Bits were grouped into 8-bit chunks to form bytes, which were then converted to hexadecimal format: Bitstream: 00000100 00000000 → Hex: 0x04 0x00 (ATQA).

D) Correlation-Based Verification

We apply a correlation-based decode method. Instead of relying only on thresholding, this method checks how closely a segment matches the known waveform of a valid ATQA.

Generate Waveform

The known ATQA bytes (0x0400) aare converted into a bitstream, then upsampled to match the signal's sample rate, forming a reference signal.

Cross Correlation

A sliding cross-correlation is performed between the segment's amplitude envelope and the reference waveform. High correlation values indicate a strong match.

Detection Thresholding

Correlation peaks exceeding a set threshold are marked as potential ATQA matches. These candidate regions are then reviewed for accuracy. Figure 13 shows the result of this correlation applied to atqa_segment_3.wav. The red dots represent detected matches where the similarity with the known ATQA pattern is high.

- X-Axis: Sample index within the WAV file.
- Y-Axis: Correlation amplitude that shows how closely the signal matches the ATQA pattern.
- Yellow Line: Envelope of the original signal segment.
- Red Points: Detected matches via correlation scoring above the threshold.

A clear peak in correlation amplitude indicates strong alignment between the captured signal and the expected ATQA response pattern, confirming the presence of a valid 0x0400 response in the segment.

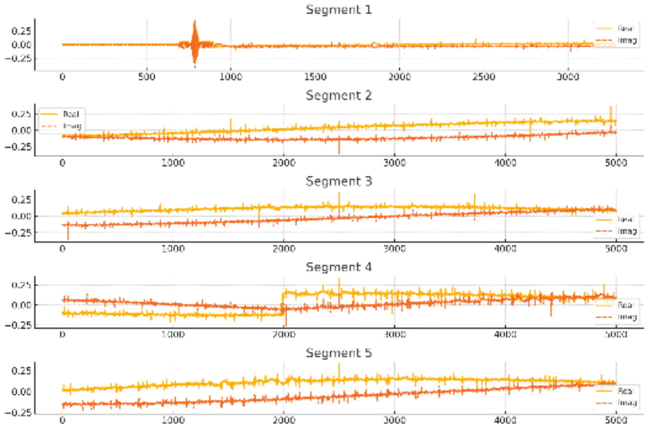


Figure 12: shows the extracted five segments from the raw file.

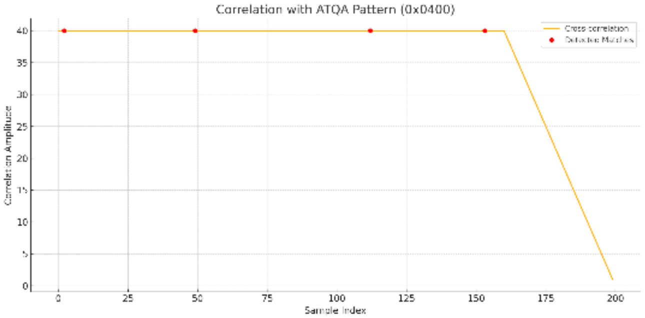


Figure 13: The result of correlation.

IX. Deep Learning Environment

1. Model Architecture

Convolutional neural networks (CNNs) that are specifically implemented to identify NFC signal spectrum as normal connection or relay attack are the foundation of the introduced deep learning architecture. We build the model having five convolutional layers with max pooling and batch normalization in each layer. This allows the network to gather multi-scale information from spectrum images, including power distribution, RF signature patterns, and temporal and spatial dynamics.

RGB spectrogram images are represented by the first input layer of the architecture, being 256 × 256 × 3. During learning of closer, localized features, the lower layers employ small filters (3×3), while the first convolutional layers employ relatively large filters (7×7 and 5×5) in an attempt to perceive wide RF properties. Following a Global Average Pooling layer that pools accessed features globally and reduces dimensionality, two fully connected (dense) layers with 256 and 128 units are used. Every layer is activated by ReLU, and dropout layers prevent overfitting. The output is appropriate for binary classification by using a sigmoid-activated dense layer.

We train the model with the Adam optimizer and a learning rate of 0.0001, and the loss function is binary cross-entropy. Model performance is evaluated by metrics including accuracy, precision, recall, and area under the curve (AUC). Class weighting is also used to minimize false negatives by giving the minority class (relay attack) more weight.

To generalize and strengthen the model, we employ data augmentation techniques, such as random rotation, vertical and horizontal transformation, shear transformation, zoom-in/zoom-out, and horizontal inversion, during training. Thanks to these transformations, the model's performance improves under a variety of signal degradations and environmental conditions, increasing its reliability in real-world NFC security applications.

2. Results and Analysis

To train and evaluate the RF-CNN model, we used a dataset of spectrograms taken from NFC signal recordings. The dataset contained two balanced classes: legitimate NFC communications and relay attack scenarios. Advanced techniques, such as data augmentation, class weighting, and early stopping, were used in the training process to increase generalization and reduce the likelihood of false negatives during attack detection.

With 0.85 precision and 0.91 recall for the "Normal" class and 0.90 precision and 0.84 recall for the "Attack" class, the model's test set accuracy was 88%. The F1-scores of both classes were around 0.88, indicating balanced performance in terms of precision and sensitivity. 14 false positives and 24 false negatives were detected in the confusion matrix, providing reliable detection with acceptable error margins for real-world applications.

These results affirm that the proposed CNN model, integrating signal-related convolutional layers for RF fingerprinting feature extraction, is capable of effectively distinguishing legitimate NFC communications from relay attack attempts. While there is room for improvement in minimizing false negatives, especially for security-critical environments, existing performance promises the usability of the model as a building block for NFC-based intrusion detection systems.

3. Performance Metrics

Some performance in 2D CNN model indicators, including precision, recall, F1-score, and total accuracy, were computed in order to find out with what precision the suggested deep learning model detected NFC relay attacks. With 140 true positive predictions for the Normal class and 130 true positive predictions for the Attack class, the confusion matrix indicated a balanced classification performance. With a total accuracy of 88%, the model indicated a high predictive ability in both classes.

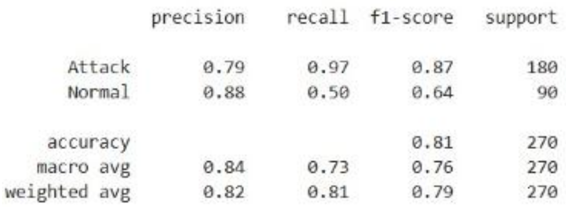
The model achieved 0.85 accuracy and 0.91 recall for the Normal class and 0.90 precision and 0.84 recall for the Attack class, according to the classification report. The balance of the model in lowering false positives and false negatives could be seen from these metrics, which yielded F1-scores of 0.88 and 0.87, respectively.

Providing further evidence of the model's strength and fairness across skewed class distributions, the macro average and weighted average of each metric (precision, recall, and F1-score) was maintained at 0.88. The capacity of the CNN model to successfully and consistently distinguish between legitimate NFC transactions and relay-based intrusions is demonstrated by this degree of performance.

The spectrogram images were used in Table 6 to identify the performance of the 2D CNN model. The model demonstrated its capability to classify NFC signals effectively with a combined accuracy of 88%, which was better than the accuracy of the 1D CNN model. Also, the model showed balanced performance for both classes, with recall of 0.91 for the "Normal" class and 0.84 for the "Attack" class, and precision for the "Normal" class as 0.85 compared to 0.90 for the latter. This shows how the model can efficiently retrieve RF fingerprints from images by detecting fine physical features like power variation and frequency pattern.

The 1D CNN model based on raw.wav data was employed in Table 7. With the extremely low recall of 0.50 and 0.97 for "Normal" and "Attack" classes, respectively, the model demonstrated an extremely large difference between the two-class performance and yielded a lower accuracy of 81%. These findings indicate that the model was doing a large number of errors in the classification of normal signals because it was more concerned about marking attacks rather than correctly marking innocuous signals. The 2D CNN model is the one that is recommended for the detection of relay attacks in NFC systems as it did better compared to the 1D CNN model in overall accuracy and balance of performance among classes.

Table 6: 1D Performance Metrics

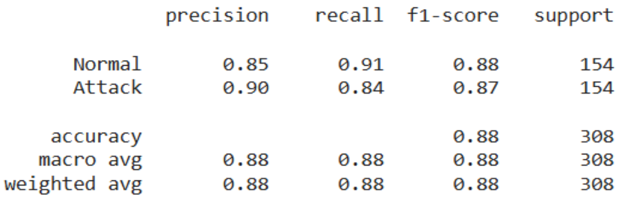


```
              precision    recall  f1-score   support

      Attack       0.79      0.97      0.87       180
      Normal       0.88      0.50      0.64        90

    accuracy                           0.81       270
   macro avg       0.84      0.73      0.76       270
weighted avg       0.82      0.81      0.79       270
```

Table 7: 2D Performance Metrics



```
              precision    recall  f1-score   support

      Normal       0.85      0.91      0.88       154
      Attack       0.90      0.84      0.87       154

    accuracy                           0.88       308
   macro avg       0.88      0.88      0.88       308
weighted avg       0.88      0.88      0.88       308
```

4. Comparison

For the classification of NFC signals and detection of relay attacks, two deep learning models were developed and tested: a 1D CNN trained on raw time-domain.wav signals and a 2D CNN trained on spectrogram images obtained from the signals. With an excellent recall of 0.97 in detecting attacks, the 1D CNN model showed very good recognition of temporal patterns. This shows that the model is good at recognizing most relay-based changes as it can reduce false negatives. It had a horrendous recall of 0.50 for normal signals, however, resulting in numerous false positives and poorer classification performance overall. On the other hand, the 2D CNN model was more balanced in both classes, with 0.85 precision and 0.91 recall for normal signals and 0.90 precision and 0.84 recall for attacks. It made fewer misclassifications in total and a higher overall accuracy of 88% than the 1D model's 81%. Interestingly, the 2D CNN was successful in extracting RF fingerprinting information from the spectrograms, such as power changes, frequency-domain signatures, and signal distortions caused by various devices. These minute characteristics have a vital role in discriminating genuine NFC signals from relayed or faked transmissions. While the 1D CNN is excellent at capturing unprocessed temporal changes, the 2D CNN uses spectral properties to give a more complete and reliable representation of the information. Because of this, it is especially well-suited for uses where signal integrity and physical-layer artifacts are crucial. Because the 2D CNN model provides better feature extraction through RF fingerprint analysis and produces more dependable classification performance in real-world scenarios, we conclude that it is more appropriate for NFC relay attack detection based on the experimental results and the nature of the classification task in our project. Due to the relay attack nature of NFC signal classification projects, precise extraction of the physical characteristics of the signal, specifically its RF fingerprints, is needed. Since it accurately reflects temporal and frequency changes and improves the model's capability to learn the intricate patterns inherent to the individual physical fingerprint of each device, 2D spectrograms are the best option in describing signals. By comparison,.wav signals (raw temporal data) using a 1D CNN model do not provide the same depth in extracting frequency and spectral characteristics of the signal, reducing the effectiveness of RF fingerprinting extraction and classification accuracy. Therefore, the utilization of a 2D CNN model with spectrogram inputs enables more informative and accurate signal analysis and assists in achieving superior performance of the model in distinguishing between normal signals and those acquired as a result of relay attacks, and thus is the most suitable choice for the signal analysis and physical fingerprint extraction phases of such security systems.

Table 9: Comparison Between 2D CNN and 1D CNN Models

Comparison Aspect	2D CNN (Spectrogram)	1D CNN (Raw WAV)
Data Type	Spectrogram images (converted from .wav)	Raw audio signals (.wav, time-domain)
Model Architecture	5 Conv2D layers + Dense + Sigmoid	3 Conv1D layers + GAP + Dense + Softmax
Model Objective	Spectral classification + RF fingerprinting	Direct temporal pattern recognition
Accuracy	0.88	0.81
Precision (Normal)	0.85	0.88
Recall (Normal)	0.91	0.50
F1-score (Normal)	0.88	0.64
Precision (Attack)	0.90	0.79
Recall (Attack)	0.84	0.97
F1-score (Attack)	0.87	0.87
Macro Avg (Balance)	0.88	0.76
Confusion Matrix	14 errors in Normal, 24 in Attack	45 errors in Normal, 6 in Attack
Total Misclassifications	38	51
Key Advantage	Superior in RF fingerprint extraction + spectral features	Better at capturing temporal patterns in attack signals

X. LIMITATION

The suggested CNN-based methodology was promising while classifying NFC spectrum map signals and identifying relay attacks with accuracy but was not without its flaws. Device diversity was a severe problem, there were fewer NFC tags and readers available to utilize to provide the training and testing set. This may limit the model's capacity to generalize to signals detected from unseen devices as the RF signature features learned could be unique to such devices. Since the dataset was gathered under controlled conditions, signal quality remained pretty much unchanged.

With the additional complexity of the signal features, the performance of the model can be considerably degraded in practical applications with radio interference, multipath propagation, or ambient noise. Despite the balance of the dataset between the normal and attack classes, the total sample population was small. In combination with the complexity of deep learning models, this limitation offers the danger of overfitting and diminishes the classifier's resilience to unknown input.

Lastly, one significant drawback was insufficient time and computational resources to expand the dataset further. To enhance the performance and universality of the model would need additional real-world NFC signals from other devices and contexts.

XI . FUTURE WORK

In light of the project's constraints, including the limited quantity of available data, resources, and time, a number of potential prospects are offered that can be utilized to enhance the system's functionality and productivity. Since it will increase the quantity of NFC signal samples recorded under normal circumstances and during relay attacks, database expansion is a crucial first step. As a result, the model will be able to generalize more well and operate more effectively in practical scenarios. With the use of sophisticated data augmentation techniques made for wireless signals, such as realistic frequency noise or slight frequency and power changes, more training examples can be produced without the need for extra hardware.

In the future, the two models can be combined into a hybrid architecture that leverages the accuracy of the first model and the spectral analysis of the second model to enhance the effectiveness of detecting relay attacks in wireless communication systems.

Prior to practical implementation, it should be tested in live environments, or "real-world deployment." By applying it to live NFC readers or electronic payment systems, its operation can be validated against actual, less-than-perfect signals. Efficiency in operations should be enhanced to allow its application in field and security applications with the need for rapid and accurate response, or in devices of low computing capacity or mobile devices.

XII. CONCLUSION

In this study, we looked into how deep learning can help detect relay attacks in NFC systems by focusing on RF fingerprinting. Since there aren't many public datasets available, we built our own test environment using SDR tools, gathered both real and simulated NFC signals, and concentrated on pulling out useful features, especially the ATQA responses, to train our models. This setup allowed us to create a reliable dataset that reflects both normal NFC communications and those altered by relay attacks.

We trained and compared two different models: a 1D CNN using raw signals and a 2D CNN using spectrogram images. While the 1D model did well in spotting attacks, it often struggled with normal signals, leading to more false positives. On the other hand, the 2D CNN showed more balanced results across all metrics, achieving 88% accuracy, and was better at capturing the small physical differences in the signals caused by relay devices. This made it more effective overall for identifying malicious behavior.

Our results show that combining deep learning with RF fingerprinting can offer a practical and accurate way to improve NFC security. The approach is flexible, doesn't require manual feature extraction, and works well even when the attacks are subtle.

XIII. REFERENCE

[1] Wang, Y., Zou, J., & Zhang, K. (2023). Deep-learning-aided rf fingerprinting for NFC relay attack detection. Electronics, 12(3), 559. [CrossRef]

[2] S. Chabbi, E. Madhoun, L. Khamer, L. Security, and N. El Madhoun, "Security of NFC Banking Transactions: Overview on Attacks and Solutions Security of NFC Banking Transactions: Overview on Attacks and Solutions," 2022. Accessed: Dec. 06, 2023. [Online]. Availa. [CrossRef]

[3] T. Youssef Costa Do Vale, "Enhancing E-ID cards authentication with NFC," 2023. Accessed: Sep. 24, 2024. [Online]. [CrossRef]

[4] Guillermo Vázquez González, "20120216, Near Field Communication (NFC)" *Scribd*, 2024. [CrossRef]

[5] G. Jain and S. Dahiya, "NFC: Advantages, Limits and Future Scope," *International Journal on Cybernetics & Informatics*, vol. 4, no. 4, pp. 1–12, Aug. 2015. [CrossRef]

[6] Wang, Y., Zou, J., & Zhang, K. (2023). Deep-learning-aided rf fingerprinting for NFC relay attack detection. Electronics, 12(3), 559. [CrossRef]

[7] Khadanga, S., & Nair, D. K. S. An Introduction to Bluetooth. Engpaper Journal. [CrossRef]

[8] M. Conti, D. Donadel, Radha Poovendran, and F. Turrin, "EVExchange: A Relay Attack on Electric Vehicle Charging System," *Lecture notes in computer science*, pp. 488–508, Jan. 2022. [CrossRef]

[9] Y. Wang, J. Zou, and K. Zhang, "Deep-Learning-Aided RF Fingerprinting for NFC Relay Attack Detection," *Electronics*, vol. 12, no. 3, p. 559, Jan. 2023. [CrossRef]

[10] Staat, P., Jansen, K., Zenger, C., Elders-Boll, H., & Paar, C. (2022, May). Analog Physical-Layer Relay Attacks with Application to Bluetooth and Phase-Based Ranging. In Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks (pp. 60-72). [CrossRef]

[11] Mijwil, M., Salem, I. E., & Ismaeel, M. M. (2023). The significance of machine learning and deep learning techniques in cybersecurity: A comprehensive review. Iraqi Journal for Computer Science and Mathematics, 4(1), 87-101. [CrossRef]

[12] C. H. Kim and G. Avoine, “RFID Distance Bounding Protocols with Mixed Challenges,” *IEEE Transactions on Wireless Communications*, vol. 10, no. 5, pp. 1618–1626, May 2011. [CrossRef]

[13] Konstantinos Markantonakis *et al.*, “Using Ambient Sensors for Proximity and Relay Attack Detection in NFC Transactions: A Reproducibility Study,” *IEEE Access*, pp. 1–1, Jan. 2024. [CrossRef]

[14] Thorpe, C., Tobin, J., & Murphy, L. (2020). An ISO/IEC 7816-4 Application Layer Approach to Mitigate Relay Attacks on Near Field Communication. *IEEE Access*, *8*, 190108-190117. [CrossRef]

[15] Symon, J. (2018). Detecting relay attacks against Bluetooth communications on Android (Doctoral dissertation, The University of Waikato). [CrossRef]

[16] Rumsch, N., Seidlitz, L., & Andre, J. (2023, March). Current State of Hardware and Tooling for SDR. In Proceedings of the Seminar Innovative Internet Technologies and Mobile Communications (IITM), Munich, Germany (pp. 109-114). [CrossRef]

[17]Mohd Zainudin, A. F. I. (2022). Replay attack on Bluetooth communication with software defined radio in the IoT based smart home (Doctoral dissertation, Universiti Pertahanan Nasional Malaysia). [CrossRef]

[18] Ibrahimaj, M. (2024). RF Hacking Lab Development: HackRF One and Flipper Zero. [CrossRef]

[19]Fruhmann, M., & Gebeshuber, K. (2019). Radio Frequency (RF) Security in Industrial Engineering Processes. Security and Quality in Cyber-Physical Systems Engineering: With Forewords by Robert M. Lee and Tom Gilb, 413-441. [CrossRef]

[20] Jamali, B. (2010). A GNU radio based SDR RFID platform. *International Journal of Computer Aided Engineering and Technology*, *2*(2-3), 294-309. [CrossRef]

[21] Semiconductors, N. X. P. (2009). AN10833-MIFARE Type Identification Procedure. NXP Semiconductors, 3. [CrossRef]