Advances in Artificial Intelligence and Machine Learning; Research 5 (4) 4765-4796 Received 21-09-2025; Accepted 19-12-2025; Published online 26-12-2025

Detecting Wireless Relay Attacks in NFC Using Deep-Learning

Asia Othman Aljahdali aoaljahdali@uj.edu.sa
Cybersecurity Department
University of Jeddah
Saudi Arabia

Maria Jawah maryils661@gmail.com
Cybersecurity Department
University of Jeddah
Saudi Arabia

Jana Bakhalqi jan.saleh22@gmail.com
Cybersecurity Department
University of Jeddah
Saudi Arabia

Talah Fairaq talahfairaq@gmail.com
Cybersecurity Department
University of Jeddah
Saudi Arabia

Corresponding Author: Asia Othman Aljahdali

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 breaking 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. This study will improve detecting of NFC relay attacks using RF fingerprints and deep learning via Bluetooth. SDR++ software and a HackRF device were used to gather 2,400 NFC signal samples. 1,200 samples are allocated to the Normal category, while another 1,200 samples are assigned to the Relay Attack category. The dataset is trained, validated, and tested using the 2D-CNN model. The model's test set accuracy was 88%, with 0.85 precision and 0.91 recall for the "Normal" class and 0.90 precision and 0.84 recall for the "Attack" class. Both classes' F1-scores were approximately 0.88, showing performance that was balanced between sensitivity and precision.

Keywords: NFC, Relay attack, Machine learning, Wireless communication.

Citation: Asia Othman Aljahdali, et al. Detecting Wireless Relay Attacks in NFC Using Deep-Learning. **Advances in Artificial Intelligence and Machine Learning**. 2025;5(4):264. https://dx.doi.org/10.54364/AAIML.2025.54264

1. 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 simply 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 unauthorizedly. 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 where the data are encrypted, it is still possible to do this. Relay attacks involve the forwarding of communication signals by the attacker without any kind of analysis or modification. There are various solutions that have addressed this issue, such as 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 a 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. The aim of this project is to harden the security on NFC apps. This work is inspired by the increasing use of NFC technology in fields like access control systems and mobile payments, leading to a need for overly simple security approaches. Relay attacks are also a great threat which cannot be easily coped with by conventional countermeasures, because even the data is encrypted. This research effort aims to bridge this gap and proposes a novel approach for the detection of relay attacks based on automatic feature extraction from NFC signal waveforms employing deep learning methods in addition to enhance the security framework of NFC systems.

2. BACKGROUND

2.1 Near Field Communication (NFC).

Near Field Communication (NFC) is a short-range wireless communication technique that belongs to the Radio Frequency Identification family. It is used to send information between two nearby (up to 10cm) devices. NFC provides a means for halfduplex communication with less than 424 bps data rate, enabling contact-less data transfer which supports the user daily life interaction such as touch-and-go payments and automatic 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 an NFC tag can be an example of a passive device [3]. NFC devices usually work in three main ways. First, there's card emulation mode. Here, your phone or device acts just like a card—like when you tap to pay or use an electronic ticket. All your info stays secure on the device. Next, you've got peer-to-peer mode. This one's handy when you want to swap files or photos with someone else. Just bring the two devices close together, and they quickly share data—no cables,

no fuss. The third is reader-writer mode. In this case, your device can read or write information on special tags, kind of like how RFID works for tracking or ID purposes. In the end, NFC just makes it easier to connect, share, and pay without all the hassle. NFC's core applications involve connecting electronic devices, accessing digital content, and facilitating contactless transactions [4].

Near Field Communication (NFC) is changing the way of interaction between devices that requires no touch at all. This innovative technology is revolutionizing more than just the way we work, but our lives across industries. From contactless payments to smart advertising, NFC applications are becoming increasingly prevalent in our daily lives.

2.2 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 ways, including by connecting wirelessly over Bluetooth and Wi-Fi [1]. 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 [5]. It is a means of connecting and exchanging data and information across video games, laptops, digital cameras, and mobile phones. [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 [5]. Wireless communications are vulnerable to several types of attacks, such as replay attacks, eavesdropping, and man-in-the-middle attacks (MitM), including relay attacks. 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 forged tag at a distant location, which can then communicate with the legitimate reader as a copy of the original tag [6]. 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 [1].

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 [7].

3. CURRENT DETECTION METHODS OF RELAY ATTACKS

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 [8].

Ambient-based methods rely on environmental conditions such as temperature, humidity, and 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 [9].

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 [1].

The ISO/IEC 7816-4 standard is employed as a relay attack detection technique by applying time analysis in the application level to differentiate between legitimate and suspicious NFC transactions. It is based on small and constant-sized Application Protocol Data Units (APDUs) exchange, which enables to maintain steady round-trip times (RTTs) between the NFC device and reader 2. By introducing an up-bound on these RTT and applying a standard deviation threshold, the system is able to detect inordinate delays caused by relays of data which heavily penalize the response time. Working at the application layer has multiple advantages: it evades time issues, which are common in lower level protocols (eg. ISO/IEC 14443). It enables practicable relay attack detection using the

standardized communication frames of ISO/IEC 7816-4 to detect and mitigate the security threats [10].

4. RELATED WORK

Symon J. (2018) proposes A relay attack detection mechanism for Bluetooth communications on lines of Android devices. The authors also propose monitoring the variation between signal strength and response time of Bluetooth packets: discrepancies in these variations are waited to neighbors on a beacon tracking platform, where they are handled when anomalies that potentially represent relay devices are detected. This approach uses machine learning algorithm for anomaly detection that balances between need of security with increase in computational cost. Detection methods Detection of key consists in: Timing Analysis, based on RTT in order to detect delays introduced by relays devices; Signal Strength and Proximity analysis, based on the RSSI for detecting unexpected proximity changes. In addition, the authors mention Cryptographic Countermeasures (note: these include distance-bounding protocols that infer physical proximity between devices based on their response times to cryptographic challenges), which can be used to mitigate attacks effectively reducing device distances. The objective of the machine learning scheme is to identifiy unauthorized entry through sensors and radio signals on Android smartphones under constrained hardware resources. However, the effectiveness of this architecture is severely constrained by permissions of Android and can still be compromised by low-latency relay attacks for which delays are minimized and thus hard to be detected [11].

Thorpe, C., Tobin, J. & Murphy, L. (2020) presented an application-layer countermeasure for detecting relay attacks in NFC communications using the ISO/IEC 7816-4 protocol. This countermeasure was based on using a round-trip time (RTT) measurement of APDU command-response pairs to measure delays caused by a relay attack. The countermeasure had a 100% detection rate and only a small false positive rate (0.38%–0.86%) through a range of different tests. Any transaction that had an RTT measurement that was above the upper threshold was stopped and flagged for significant delay (i.e. delay caused by interference resulting in a relay attack). The authors implemented a relay attack on the NFC devices using standard Android smartphones to provide a demonstration of variations of the relay attack while discussing attack feasibility and associated risks. The relay attack successfully completed a transaction 1-4 seconds after it was started, which was possible due to the additional time from the attack being relatively small. By using the ISO/IEC 7816-4 protocol that measures RTT of certain APDU command-response pairs, if the measurement is above established threshold levels the transaction was flagged for potentially being a relay attack. This technique is limited to NFC technologies using the ISO/IEC 7816-4 protocol and would also face challenges in environments with standard high network interference. Furthermore, the authors assume that there is some level of RTT delay where a relay attack is occurring, but this does not necessarily cover all potential scenarios where delays are optimized by good or sophisticated relay attackers [10].

Though the distance bounding technique is a countermeasure against relay attacks in wireless communications, like near field communication (NFC). Distance bounding works under the premise of measuring round trip time (RTT) of messages exchanged between two communicating parties allowing the initiator of the protocol to gain confidence between themselves and the responder. The distance bounding protocol typically operates via the initiator providing the responder with a series of challenges that the responder responds to with the appropriate response. The initiator

can analyze the RTT of each message and use the response time to determine if the responder was located a predetermined distance when the initiator provided the responder with the challenge. If the responding time exceeds that threshold then the responder is either too far to be valid, or the resolver is involved in a possible relay attack. Distance bounding provides a way to ensure that the two communicating devices are in proximity to each other enhancing the security of transactions or access control systems. In the study completed by Chong Hee Kim and Gildas Avoine [8], they critique existing distance bounding protocols as their primarily use binary challenges and do not sign the final challenge/response step leading to the adversary having a probability of success of $(3/4)^n$ where n represents the number of rounds. The authors provide a new protocol that uses mixed challenges (both random and pre-defined challenges) to bring the adversary probability of success down to $(1/2)^n$ as the optimal bound. In the reader and tag protocols, both parties exchange nonces and compute sequences that consist of both random and pre-defined challenges. The challenge is constructed so that if an adversary attempts to acquire responses to challenges in advance, the tag will be able to recognize the situation. This paper also explores cases where noise can affect the communication model to see how distances bounding protocols perform in these circumstances. The proposed protocol improves the distance bounding protocols, which employ a mixture of random and pre-defined challenges, which reduces the probability of success for adversary attacks or for the adversary to obtain multiple responses. Although this protocol has significant defense mechanisms, distance bounding protocols fundamentally rely on accurate time measurement. During noisy conditions or signals interference conditions, the time measurement defined may diminish or may even allow the possibility for relay adversary attacks to take place without detection. Mixed challenges and the mixture of random and predefined challenges improve security further, but for an advanced adversary, the predictable nature of some of the pre-defined challenges will still allow the adversary to prepare in advance [8].

Konstantinos Markantonakis et al. [9] examine the use of various ambient sensors (accelerometers, gyroscopes, and environmental sensors) to detect anomalies within the physical context underlying the NFC interaction process. By observing interactions relating to device movement and surrounding conditions, the aim is to identify when an attack is possibly happening. Researchers carry out experiments using ambient sensors to compare the efficacy of ambient sensor data to identify legitimate NFC transactions and unauthorized relay attacks. The researchers examined ambient sensor data to identify an anomaly from NFC transaction data and conducted experiments to observe ambient sensor data during NFC transactions. The authors then analyzed the ambient sensor data to identify anomalies suggesting an attack or relay. The authors applied three approaches to evaluate the benefit or misuse of the ambient sensors, the first approach was a 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 resolved the challenge of the variety of sensor data formats through the Haversine formula for location measures and mean absolute error (MAE) and correlation coefficient for other sensors after performing linear interpolation to correct for clock mismatch. For the three-dimensional sensors, vector magnitudes were calculated to enable simpler comparisons. A Python application was created to determine the false positive rate (FPR) and false negative rate (FNR) for the legitimate and the unauthorized transaction measurements. The objective was to find an appropriate threshold for the metrics of similarity that limited errors in detection while allowing as many legitimate transactions as possible, with the Equal Error Rate (EER) acting as a focal point of convergence. The method also assessed the transaction and sensor failure rates, which demonstrated potential usability problems in reliability and security when the quick device

movement more frequently resulted in missed data, indicating the difficulty in paired authentication of NFC transaction. Their second methodology was machine learning analysis, which acknowledged supervised machine learning algorithms by further enhancing the detection of relay attacks made through sensor data. The third methodology, deep learning methodology, is divided into two methods. Method 3 uses fully connected artificial neural networks (ANNs) to detect relay attacks through easier to manage feed-forward architectures consisting of input, hidden, and output layers.

The security risks associated with relay attacks in near-field communication (NFC) systems are discussed in research by Wang, Y., Zou, J., & Zhang, K. (2023) [1], with a focus on radio-frequency identification (RFID) technology. Using radio frequency fingerprinting, the research introduces a unique method for detecting NFC relay assaults. 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. 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 as time series signals or pictures. 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.

The papers mentioned above together cover various facets of wireless communication anomaly detection. In Symon J. (2018) [11], 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) [10] offer 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] employ CNN and other deep learning algorithms for real-time anomaly identification, they encounter issues with sensor reliability and temporal restrictions.

The C. H. Kim and G. Avoin [8] 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 environments, in the 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.

Konstantinos Markantonakis et al. [9] evaluate 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 in our research, we will use RF fingerprinting to collect datasets and use deep learning to detect the relay attacks. that provides more adaptability to evolving environments, identifies a wider range of anomalies, real-time analysis and detection, and accommodates 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, it can be more effective in distinguishing legitimate devices from attackers. TABLE 1 presents a comparative analysis of the current related works.

Table 1: Related Work Comparison

Ref	Methods and Layer	Features	Datasets	Accuracy	Limitations
[12]	Timing Analysis and Signal Strength Analysis Machine Learning (ML) for Anomaly Detection. On physical Layer.	RSSI and Timing Anomaly Detection ML Integration enhances detection optimizing accuracy over time.	Detecting Relay Attacks Against Bluetooth on Android (1,200 instances). Subsets: 10% (121) and 1% (12) used for smaller sample testing.	Wi-Fi: 98.3%.Combined (Bluetooth, Wi-Fi, Cell): 98.3%.Bluetooth: 86.7%.	Limited to Android OS. Restricted by permissions. Timing detection insufficient for low-latency attacks. RSSI affected by environment. Bluetooth unreliable due to variability.
[11]	Distance-bounding. On application Layer	100% detection accuracy.Low false positive rate (0.38–0.86%).	10,000 NFC transactions (contactless cards and terminals).Measured round-trip times (RTTs) for uninterrupted and relayed transactions.	Detection rate: 100%. False positive rate: 0.38–0.86%.Delay per transaction: 0.22s.	Effective mainly for ISO/IEC 7816-4 systems. Less effective for minimal RTT relay methods. Adds slight delay to transactions.
[9]	Distance-bounding Mixed Challenge Protocol. On application Layer.	Mixed challenges enhance security. Measures RTT for proximity. Avoids final signature to reduce computational cost.	N/A	Reduces adversary success probability to $(1/2)^n$.	Adversary still has 50% success probability.
[10]	Ambient-Based Solutions Machine Learning and Deep Learning (ANN, CNN, RNN). On application Layer.	MAE, Correlation, Classification (Logistic Regression, SVM), Pattern recognition, Spatial and Temporal detection.	1,000 transactions per sensor. 17 sensors total. Some sensors failed 99% of the time; humidity/temperature sensors valid in only 6%.	CNN EER: 0.246. RNN EER: 0.273. Traditional ML models outperform deep models.	High error rates. Limited accuracy. High computation cost. Resource-intensive. Not suitable for real-time use.
[1]	Deep Learning with CNN RF Fingerprinting Dataset Creation and Classification. On physical Layer	Real-time detection. Comprehensive dataset. High accuracy.	66,366 NFC samples: 4 wired relay attacks, 4 conventional NFC tags, 1 wireless relay via Wi-Fi.	High accuracy on new dataset.	Dependent on signal quality. Relies only on Wi-Fi data sources.

5. 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. FIGURE 1 depicts the proposed solution workflow.

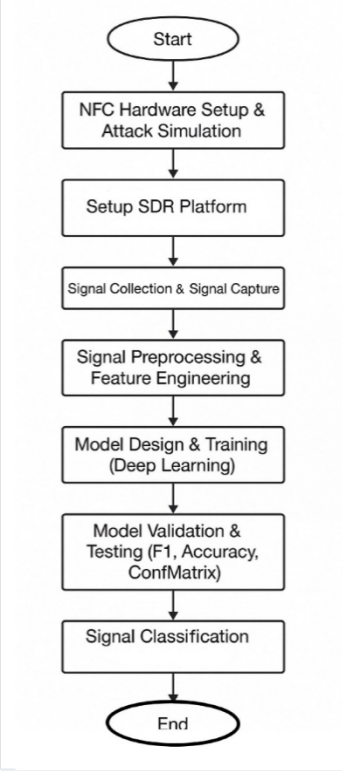

Figure 1: Proposed Solution Workflow

5.1 NFC Hardware Setup and 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 represents 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. The purpose of this setting is to establish normal interaction between the reader and the tag as an important preamble to initiating the relay attack, and to facilitate the later comparison of normative signals to those from the relay attack as part of the overall project.

The second one is a scenario simulating a relay attack. The testbed is designed to resemble a real NFC relay attack. On the hardware side, there are embedded NFC and wireless relay emulators. More specifically, the system comprises a real NFC-reader, a real NFC tag, reader emulated device (RED) and tag emulated device (TED), read-write specific 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.

5.2 Set Up 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. TABLE 2 presents a comparison between different SDR devices.

The Software Defined Radios (SDRs)—AirSPY, HackRF, and USRP (Universal Software Radio Peripheral)—all have their respective strengths and applications. The AirSPY is a high-speed SDR capable of sampling at a rate of 10 MSPS, with a 12-bit ADC intended for precision work, and thus is typically a good option for narrowband applications such as detecting a NFC relay attack via onboard radio frequency analytics, however due to its narrow band of frequencies (from 24 MHz to 1.7 GHz) while the AirSPY is an excellent SDR for narrowband detection it is still considered narrow band which limits its use for a broader application or for work requiring larger bandwidth, as compared to SDRs such as a HackRF or a USRP [13]. In contrast, HackRF is the practical and lower budget option for NFC relay attack detection over Bluetooth due to its wider frequency range (1-6 GHz), as well as its ability to function in the 2.4 GHz frequency range used by Bluetooth [14]. HackRF has a sampling rate of up to 20 MSPS and supports open-source software like GNU Radio and gr-Bluetooth to capture, alter, and relay Bluetooth packets, which make up a large portion

of emulating and spoofing NFC communications over Bluetooth relays [15]. The versatility of the HackRF is encouraging, though it is slightly lower resolution with its 8-bit offerings when compared to AirSPY.

On the other hand, USRP is still the most flexibly oriented and advanced SDR platform. It supports higher sampling rates (up to 50 MSPS) and its full transmission functions [15], while also being useful in very complicated and multi-domain, protocol-intensive applications, like 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 resolves the cost vs. capabilities issue and demonstrates to be a more adaptable solution for use cases that require to backup NFC attacks based on Bluetooth. Both ideas highlight that the SDR selection is very dependent on the necessary frequency range, budget, and application requirements.

Table 2: Comparison of SDR Devices

Criteria	AirSPY	HackRF	USRP
Sampling Rate	Lower	High	High
Flexibility	Lower	High	High
Accuracy	High	Lower	High
Cost Effectiveness	Lower	Lower	High
Broad Frequency Applications	Lower	High	High

5.3 Signal Collection.

We perform structured signal capture for normal NFC communication and relay attack. The signal capture requires establishing two types of NFC communication: 1. Normal Communication: Walk up and read NFC reader to NFC tag communication in a straightforward manner that does not require additional overhead to perform the communication. 2. Relay Attack Communication: A replay attack in which the NFC reader communicates to the tag through an indirect relay channel to imitate a real-world relay attack. To capture these schemes, we use the SDR device HackRF One, with the SDR++ software platform. The HackRF One is tuned to listen on the 13.56 MHz carrier frequency associated with ISO/IEC 14443-A NFC systems. In all cases for this test, the NFC tag is available in the read area to maintain consistent transmission of the supply voltage draining the tags battery, and to allow for the SDR subsequent capturing of electromagnetic signals. Mutable NFC tags allow for defining standard output conditions while adding a repeated pUSE feature. To guarantee diversity in the dataset and separation between RF (Radio Frequency) fingerprint and environmental condition types, several recordings are taken for each category of communication (normal and relay). These recordings form the basis of our raw dataset, which is then processed and analyzed for attack detection and classification.

2,400 NFC signal samples were collected using a HackRF device and SDR++ software. The samples were evenly distributed into two categories: 1,200 samples for the Normal category and 1,200 samples for the Relay Attack category. For each category, we used a training, tuning, and testing

time split as follows: 820 samples for the training (train), 180 samples for the validation (validation), and 200 samples for the final (test).

5.4 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.

5.5 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.

5.6 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 the quality of the model in separating legitimate NFC communications from relay attacks while documenting any tendency towards misclassifying. In conjunction, the numbers produce a summary score for performance, which can be proposed for future real-world applications of NFC security.

5.7 Signal Classification.

After data collection and preprocessing, all recorded NFC signals, were classified into either (1) normal communication or (2) relay attacks. 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.

6. EXPERIMENTAL SETUP

6.1 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 simulated relay attacks.

6.1.1 PCDs (Readers).

TABLE 3 lists the different types of PCD devices used in the experiment. Reader 1, a smartphone (Hawaii—Android), was 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, is used to read the tags in the main electronic circuit and to ensure the success of the tag copying process in a relay attack, as shown in FIGURE 2. Additionally, the reader 3 using the RC522 module acts as a transmission device to read the main card data and then send it to the copier device, as shown in FIGURE 5.

Table 3: Inventory of PCDs Devices

Name	**Type**	**Model**	**Function**
Reader 1	Smartphone	Huawei - Android	To read the tags in SDR (Software-Defined Radio) capture
Reader 2	Arduino Simulator	RC522	Reads tags in the main electronic circuit and ensures success of tag copying in relay attack
Reader 3	Arduino Simulator	RC522	Acts as a transmission device: reads main card data and sends it to the copier device

Indeed, only one reader will be used to elaborate the final dataset; it is Reader 1 (Hawaii – Android). That's because this decision is driven by the fact that Reader 2, which relies on the RC522 module connected to an Arduino board, continuously reads signals as long as it is powered on. This constant activity causes it to capture noise and unrelated signals. On the 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 proximity. This design reduces the noise and ensures more reliable capture of tag and card signals during SDR-based signal recording.

6.1.2 PICCs (Tags).

TABLE 4 presents the list of NFC tags and cards used. A total of four cards are used, all compliant with the 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 support write operations, and they are based on MIFARE Classic 1k cards 1 and 2 chips to perform simulated relay attacks. Especially used to copy the UID and signal of cards 1 and 2. These cards and tags are depicted in FIGURE 3.

Table 4: Inventory of PICC Devices

No.	Name	NFC Type	Standard	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

6.1.3 Supporting devices.

TABLE 5 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 are depicted in FIGURE 6. Two ESP32-based microcontrollers acted as Bluetooth transceivers to forward NFC data wirelessly; these are depicted in FIGURE 4 and FIGURE 5. Reader RC522 is a 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 are depicted in FIGURE 4. Finally, an LCM1602 IIC LCD display module was used to display captured cards and tag IDs, which helps in visual confirmation so that we can be sure that the copy was successful; these are depicted in FIGURE 2. Each of these components played a critical role in either capturing, transmitting, or validating the NFC communication normal and simulated relay attacks.

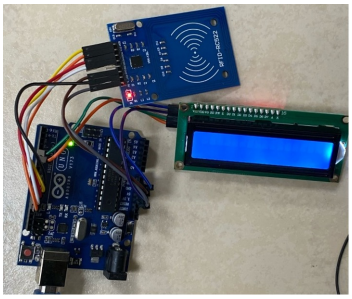

Figure 2: Reader in the main electronic circuit, Arduino Uno board and Display Screen

Figure 3: NFC Tags

Table 5: Inventory of Other Devices

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: receives data from the transmission device and sends it to a writable tag
5	Display Screen	LCM1602 IIC	-	Shows the card ID and displays it on the screen

6.2 Building Setup

6.2.1 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 presents a diagram of a normal data collection setup.

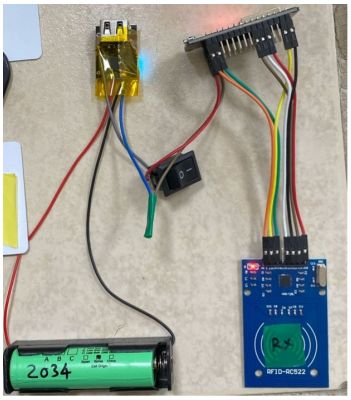

Figure 4: The Reader is a Transmission Device

6.2.2 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).

7. DATASET CREATION

7.1 Radio Setup

The SDR and Antenna Setup: 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 [13]. The SDR is combined with a custom-made near-field loop antenna tuned for 13.56 MHz so that the antenna will resonate with ISO 14443 protocols for NFC systems. The antenna has been impedance-matched and placed close to the reader-tag system to provide the best inductive coupling.

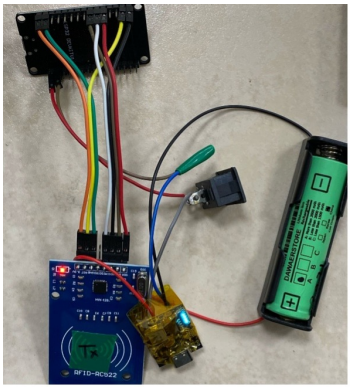

Figure 5: The Reader is a Copying Device

Figure 6: The SDR HackRF

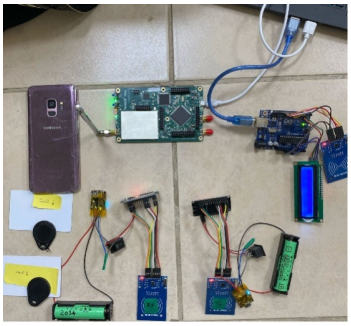

Figure 7: Components Used in Detecting Relay Attack

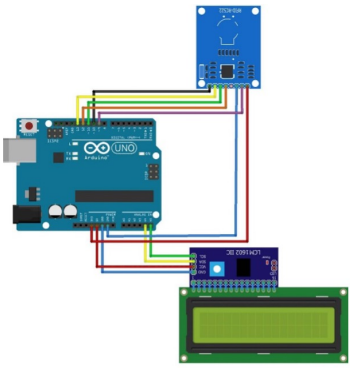

Figure 8: Diagram of Normal Setup

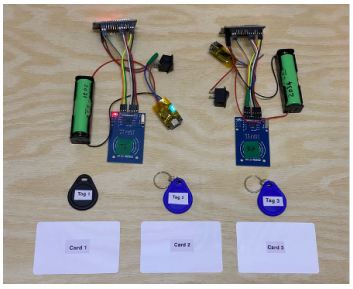

Figure 9: Relay Attack Setup

7.2 Used Software

7.2.1 SDR++

The first part of the analysis phase involves the SDR++ application, which we utilize to familiarize ourselves with SDR in general, and to conduct our first recordings and real-time signal monitoring at 13.56 MHz. In this regard, it is very helpful; the SDR++ application has a nice and simple to use GUI that permits the NFC tag and reader to be visually aligned. Once the sampling rate of 2 MSPS is available in the configuration options, the SDR++ program can be used to precisely tune and calibrate the radio prior to data acquisition. Nevertheless, our configuration needs has some limitations which are beyond our control. We are unable to obtain a recognizable ATQA signal due to the application's limitation of features which does not permit the configuration of a higher sample rate. This leads to the inability to identify the ATQA properly, which is understandably frustrating. We do not realize the limitations until it is too late, after we check signals multiple times and confirm a legitimate connection. The fact that we are distorted and discovering this for the first time means we are not making as much progress as we could in operating the SDR++ application more effectively.

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, see FIGURE 10. These are the reasons why we replaced the SDR++ app with the GNU Radio application.

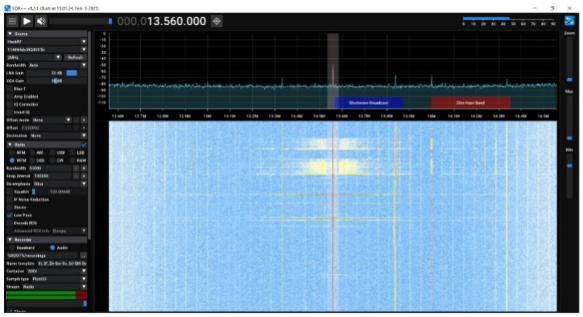

Figure 10: Capturing Signals from the SDR++ Platform

7.2.2 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 [16]. 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. Subsequently, this file can be analyzed or further processed for different signal analysis purposes. The flowgraph is very simple and works well to ensure the raw I/Q data is

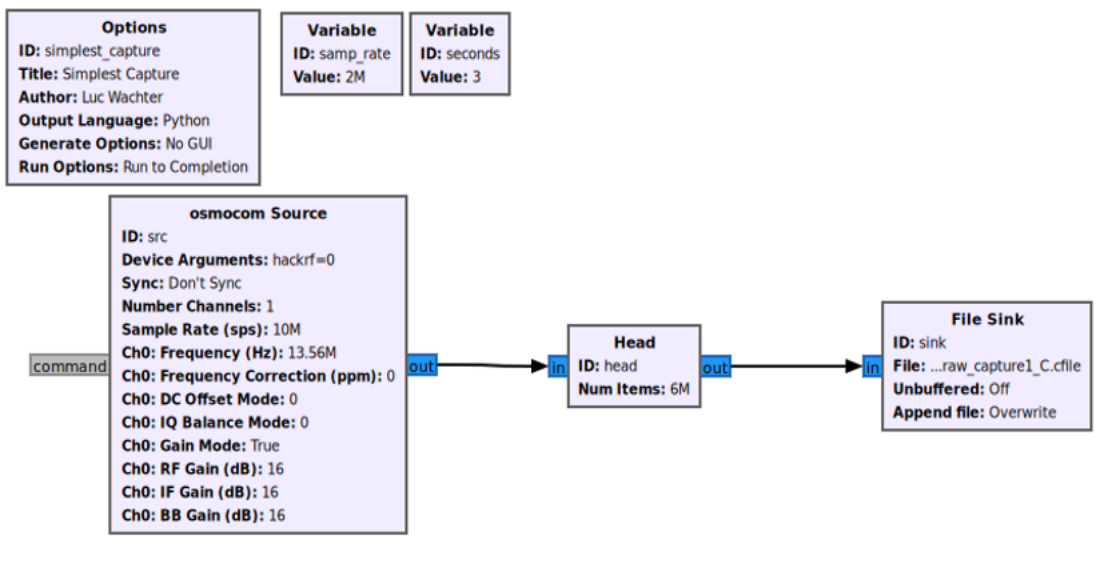

Figure 11: GNU Radio Companion (GRC) flowgraph used for capturing raw radio signals with a HackRF One device.

properly captured and saved for further processing. NFC interactions between the tag and reader are recorded, and then a subsequent process will analyze, for example, the answer-to-request type A (ATQA) in the interaction.

7.3 Features Extraction

7.3.1 ATQA

The answer-to-request type A (ATQA) is a well-known NFC response signal standardized and 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 [16].

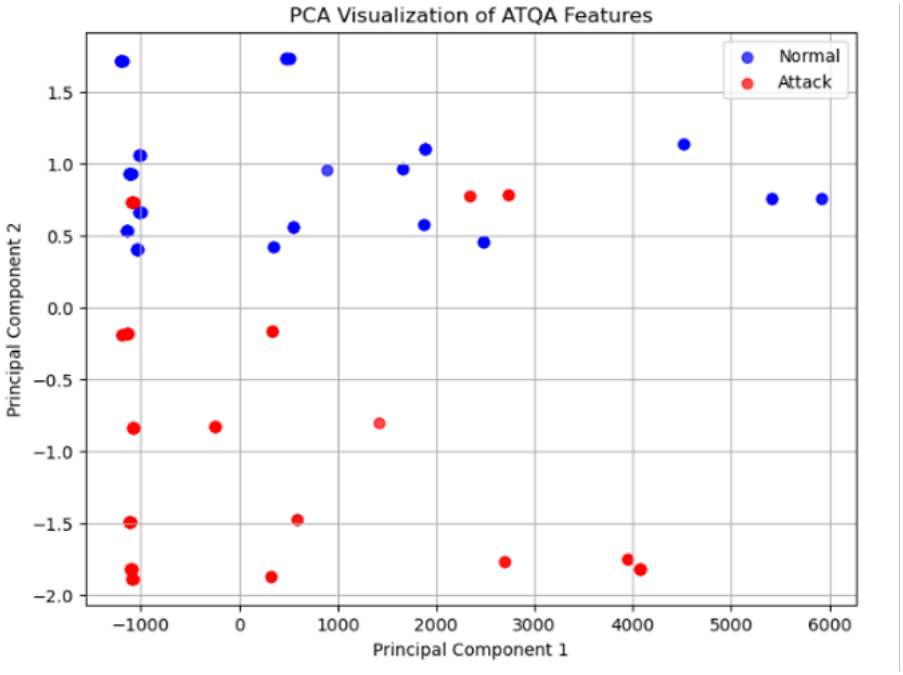

Figure 12: PCA projection of the extracted RF features.

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 (50 us to 1 ms), which typically correspond to ATQA. Each identified ATQA segment is recorded as a separate WAV file, with the duration normalized to equal (512 samples). RF fingerprinting features, including energy, RMS (root mean square), peak, standard deviation, and duration, are generated for each segment. This allows us to create a dataset composed of physical-layer device characteristics and makes ATQA a reliable signature.

To demonstrate the distinctiveness of ATQA responses between normal and relay attack signals, a feature-level comparison was conducted. FIGURE 12 presents a PCA projection of the extracted RF features (energy, RMS, peak, duration, and standard deviation), showing a noticeable separation trend between normal and attack samples. FIGURE 13 illustrates the energy distribution histogram for both classes, where relay attack signals exhibit higher and more variable energy levels compared to normal signals. These visualizations clearly confirm that ATQA-based RF fingerprints contain distinctive patterns that can be effectively utilized for reliable classification.

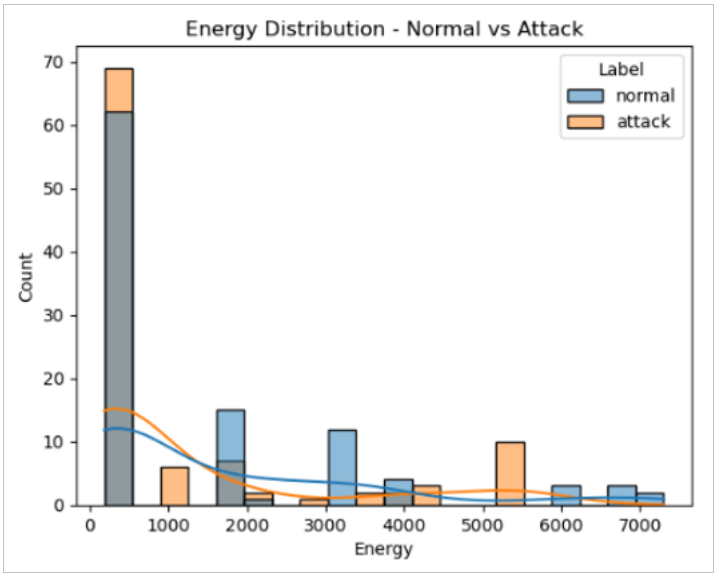

Figure 13: The energy distribution histogram.

7.3.2 Validating the dataset—Measure library (urh)

In this section, we describe how we extract, process, and validate segments that may contain ATQA responses, and decode and correlate these towards an experimental decoding known ATQA patterns (0x0400) as defined in ISO/IEC 14443A. While fully validating the dataset is impractical, we construct a multi-step validating process to provide confidence in the integrity, balance, and discriminability in the dataset before training models on the deep learning platform. Our validation process involves both automatic and manual validation checks.

Manual Label Consistency Check—Byte-Level Signal All extracted segments are traced back to their original WAV files (e.g., C1.wav = Normal, T1.wav = Attack). The original NFC recordings are brought into Universal Radio Hacker (URH), a signal analysis tool for reverse engineering protocols. We manually mark the suspected ATQA segments in URH using the waveform view and decode timeline. We demodulate the baseband waveforms into bits utilizing the On-Off Keying (OOK) feature. The bits are converted into byte streams and displayed in hexadecimal. We manually confirm the hex values correspond to ATQA responses (0x0400 for MIFARE Classic 1K), which confirms that the segments represented ATQA. This method provides signal-level confidence that our extract segments are correct ATQA messages, not RF noise or unrelated commands.

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 at 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 14 (Real + Imag parts) represent the extracted signal segments, where sudden increases in signal strength indicate potential NFC tag responses.

Byte Recovery. With the use of the Measure library (urh) we recovered bytes. 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).

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) are 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 15 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.

8. DEEP LEARNING ENVIRONMENT

8.1 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.

The 2D-CNN model consists of five convolutional layers followed by fully connected layers for binary classification. It contains approximately 2.3 million parameters and was trained for 50 epochs with a batch size of 32 using the Adam optimizer (learning rate = 0.001). Training was performed

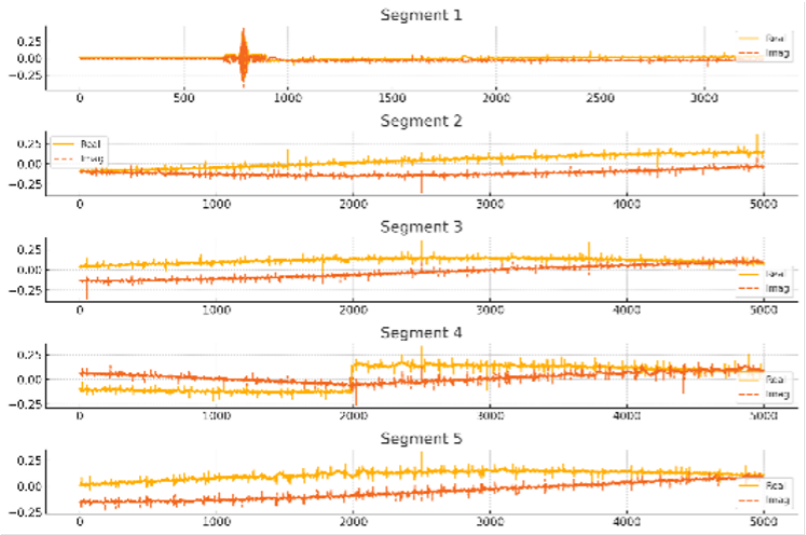

Figure 14: The extracted five segments from the raw file.

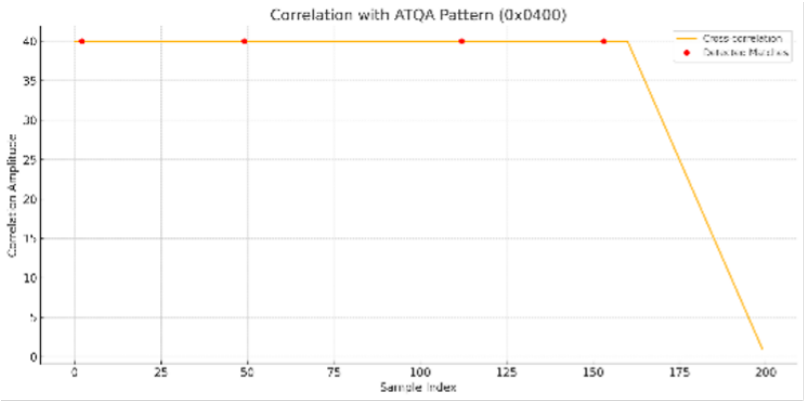

Figure 15: Correlation Results

on Google Colab (Tesla T4 GPU, 16 GB VRAM) and took about 25–30 minutes in total (35 seconds per epoch).

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. These transformations improved the performance of the model across a wide range of signal distortions and environmental conditions, increasing reliability in practical NFC security contexts (FIGURE 16).

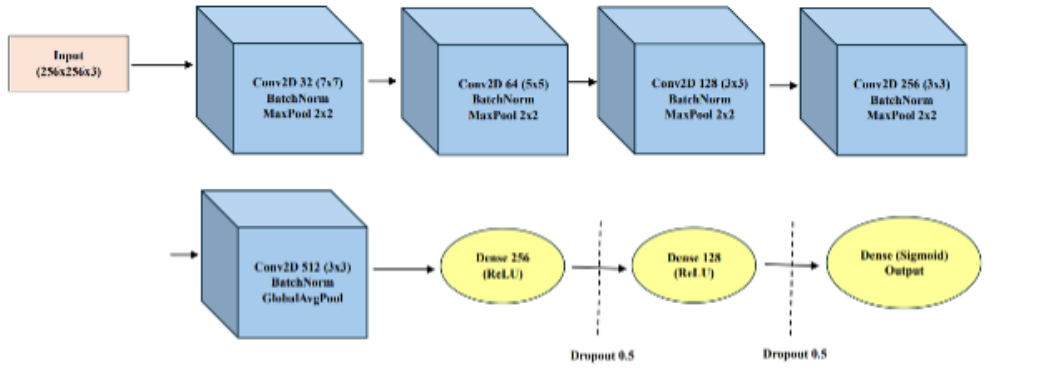

Figure 16: Compact CNN architecture for NFC signal classification.

8.2 Results and Analysis

To train and assess the RF-CNN model, we can leverage a dataset of spectrograms extracted from recordings of NFC signals. The dataset featured two balanced classes: legitimate NFC communications and relay attacks. In the training procedure, we employed best practices such as data augmentation, class weighting, and early stopping to promote generalization and decrease deviation from accurate attack detection, or false negatives. The model achieved an overall accuracy of 88% on the test set, providing 0.85 precision and 0.91 recall for the ”Normal” class and 0.90 precision and 0.84 recall for the ”Attack” class. The F1-scores for both classes were close to 0.88, showing adequate performance between precision and recaller performance. The confusion matrix noted 14

false positives and 24 false negatives, which signifies robust detection with reasonable error rates in the real world.

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.

A 3-fold cross-validation was conducted to evaluate the model's stability. The results show an average accuracy of 60.78% ± 15.25%, precision of 59.32% ± 13.18%, recall of 96.73% ± 4.62%, and F1-score of 72.32% ± 8.00% across folds. These results indicate that the model's performance was evaluated robustly across multiple data partitions, reducing the likelihood of results being due to a single test split, although the relatively high variance suggests some instability across folds.

8.3 Performance Metrics

Some performance indicators in the 2D CNN model, 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. Furthermore, the model demonstrated an ability to classify NFC signals with an accuracy of 88%, which proved to be better than that 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 making a large number of errors in the classification of normal signals because it was more concerned about marking attacks 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.

	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 6: 1D 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

Table 7: 2D Performance Metrics

8.4 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

Table 8: Comparison of 2D 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

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 8).

In comparison with previous studies, our work presents a deep learning-based RF fingerprinting approach operating at the physical layer, achieving 88% accuracy with a 2D CNN and 81% with a 1D CNN. The research in [11] concentrated on the physical layer, utilizing timing and signal strength analysis in conjunction with machine learning for anomaly detection. It reported up to 98.3% accuracy across multiple wireless technologies. However, that approach primarily relied on handcrafted signal features rather than deep neural representations. At the application layer, studies [10] and [8] utilized distance-bounding protocols, achieving near-perfect detection and theoretically minimizing adversarial success probabilities, though at the cost of additional processing delays. In contrast, [9] explored ambient-based and deep learning solutions incorporating CNN, RNN, and fully connected ANN architectures, achieving equal error rates (EER) between 0.246 and 0.273, reflecting moderate performance. Overall, our proposed method distinguishes itself by focusing on physical-layer RF fingerprinting with custom dataset creation and CNN-based classification, offering a balance between high accuracy and system-level simplicity without relying on higher-layer contextual or timing information.

9. CONCLUSIONS AND FUTURE WORK

In this study, we investigated the use of deep learning for detecting relay attacks in NFC systems through RF fingerprinting. Due to the limited availability of public datasets, we developed our own experimental environment using software-defined radio (SDR) tools to collect both real and simulated NFC signals. Particular emphasis was placed on extracting discriminative features—especially the ATQA responses—to train and evaluate the proposed models. This approach enabled the creation of a reliable dataset that represents both normal NFC communications and those affected by relay attacks. Two deep learning architectures were explored: a 1D CNN trained on raw signal data and a 2D CNN trained on spectrogram images. While the 1D CNN demonstrated good performance in detecting attacks, it exhibited a higher false positive rate for normal communications. In contrast, the 2D CNN achieved a more balanced performance across all metrics, reaching an accuracy of 88% and effectively capturing subtle physical differences in signals caused by relay devices. These results confirm that combining deep learning with RF fingerprinting provides a practical and accurate means to strengthen NFC system security. The approach eliminates the need for manual feature engineering and maintains high detection capability even when attacks exhibit minimal signal distortions. However, the study also revealed limitations related to device diversity and controlled data collection conditions. The limited number of NFC readers and tags used may restrict the model's ability to generalize to signals from unseen devices, as the learned RF signatures may be device-specific. Furthermore, since the dataset was captured in a controlled environment, real-world signal variability was not fully represented.

Future research will address these limitations by expanding the experimental setup to include a larger and more diverse range of NFC readers, tags, and environmental conditions. Additionally, we aim to evaluate the proposed framework under real-time operational settings to assess latency and deployment feasibility in practical NFC systems as well as the model's resilience under adversarial noise and adaptive relay attacks. Further exploration of hybrid architectures that combine the strengths of both 1D and 2D CNN models may also improve performance in complex or dynamic environments. These enhancements are expected to strengthen the robustness, generalizability, and real-world applicability of the proposed approach.

References

[1] Wang Y, Zou J, Zhang K. Deep-Learning-Aided RF Fingerprinting for NFC Relay Attack Detection. Electronics. 2023;12:559.

[2] Chabbi S, El Madhoun N, Khamer L. Security of NFC Banking Transactions: Overview on Attacks and Solutions. In 2022 6th Cyber Security in Networking Conference (CSNet). IEEE. 2022:1-5.

[3] do Vale TY. Enhancing E-ID Cards Authentication with NFC. Publication number: 31517138. [Master's thesis, ISCTE-Instituto Universitario de Lisboa (Portugal)]. 2023. Available at: `https://www.proquest.com/openview/45dda9362404cf23b916bb22cb72d890/1?pq-origsite=gscholar&cbl=2026366&diss=y`

[4] Al-Ofeishat HA, Al Rababah MA. Near Field Communication (NFC). Int J Comp Sci Netw Sec. 2012;12:93-99.

[5] https://www.engpaper.com/download/an-introduction-to-bluetooth.pdf

[6] Conti M, Donadel D, Poovendran R, Turrin F. EVExchange: A Relay Attack on Electric Vehicle Charging System. Atluri V, Di Pietro R, Jensen CD, Meng W, editors. Computer security – ESORICS 2022: 27th European symposium on research in computer Security, Copenhagen, Denmark, September 26-30, 2022, proceedings, part I. Cham: Springer International Publishing. 2022:488-508.

[7] Staat P, Jansen K, Zenger C, Elders-Boll H, Paar C. 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. New York, USA: ACM. 2022:60-72.

[8] Kim CH, Avoine G. RFID Distance Bounding Protocols with Mixed Challenges. IEEE Trans Wirel Commun. 2011;10:1618-1626.

[9] Markantonakis K, Meister JA, Gurulian I, Shepherd C, Naeem Akram R, et al. Using Ambient Sensors for Proximity and Relay Attack Detection in NFC Transactions: A Reproducibility Study. IEEE Access. 2024;12:150372-150386.

[10] Thorpe C, Tobin J, Murphy L. An ISO/IEC 7816-4 Application Layer Approach to Mitigate Relay Attacks on Near Field Communication. IEEE Access. 2020;8:190108-190117.

[11] Symon J. Detecting Relay Attacks Against Bluetooth Communications on Android. [Thesis, Master of Cyber Security (MCS)]. The University of Waikato, Hamilton, New Zealand. 2018. Available at: https://hdl.handle.net/10289/12077

[12] Rumsch N, Seidlitz L, Andre J. Current State of Hardware and Tooling for SDR. In: Proceedings of the seminar innovative internet technologies and mobile communications (IITM). 2023:109-114.

[13] Mohd Zainudin AF. Replay Attack on Bluetooth Communication With Software Defined Radio in the IoT-Based Smart Home. [Masters thesis, Universiti Pertahanan Nasional Malaysia.] 2022. Available at: http://repo.upnm.edu.my/id/eprint/244

[14] Ibrahimaj M. RF Hacking Lab Development: HackRF One and Flipper Zero. [Bachelor's Thesis, Metropolia University of Applied Sciences]. 2024. Available at: https://www.theseus.fi/handle/10024/869150

[15] Fruhmann M, Gebeshuber K. Radio Frequency (RF) Security in Industrial Engineering Processes. In: Biffl S, Eckhart M, Lüder A, Weippl E., editors. Security and quality in cyber-physical systems engineering. Cham: Springer International Publishing. 2019:413-441.

[16] https://www.newark.com/pdfs/techarticles/nxp/AN10833.pdf