Using Ambient Sensors for Proximity and Relay Attack Detection in NFC Transactions A Reproducibility Study

MARKANTONAKIS, Konstantinos; MEISTER, Julia A.; GURULIAN, Iakovos; SHEPHERD, Carlton; NAEEM AKRAM, Raja; HANI ABU GHAZALAH, Sarah; KASI, Mumraiz; SAUVERON, Damien; HANCKE, Gerhard

Published in:
IEEE Access

Published: 01/01/2024

Document Version:
Final Published version, also known as Publisher's PDF, Publisher's Final version or Version of Record

Published version (DOI):
10.1109/ACCESS.2024.3479729

Publication details:
MARKANTONAKIS, K., MEISTER, J. A., GURULIAN, I., SHEPHERD, C., NAEEM AKRAM, R., HANI ABU GHAZALAH, S., KASI, M., SAUVERON, D., & HANCKE, G. (2024). Using Ambient Sensors for Proximity and Relay Attack Detection in NFC Transactions: A Reproducibility Study. *IEEE Access*, *12*, 150372-150386. https://doi.org/10.1109/ACCESS.2024.3479729

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 August 2024, accepted 5 October 2024, date of publication 14 October 2024, date of current version 24 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3479729

RESEARCH ARTICLE

Using Ambient Sensors for Proximity and Relay Attack Detection in NFC Transactions: A Reproducibility Study

KONSTANTINOS MARKANTONAKIS[1], JULIA A. MEISTER[2], IAKOVOS GURULIAN[3], CARLTON SHEPHERD[4], RAJA NAEEM AKRAM[5], SARAH HANI ABU GHAZALAH[6], MUMRAIZ KASI[7], DAMIEN SAUVERON[8], AND GERHARD HANCKE[9], (Fellow, IEEE)

[1]Information Security Group, Smart Card and IoT Security Centre, Royal Holloway, University of London, TW20 0EX Egham, U.K.
[2]School of Computing, Engineering and Mathematics, University of Brighton, BN1 9PH Brighton, U.K.
[3]TEKA Systems S.A., 152 31 Athens, Greece
[4]School of Computing, Newcastle University, NE1 7RU Newcastle upon Tyne, U.K.
[5]Department of Computer Science, University of Aberdeen, AB24 3FX Aberdeen, U.K.
[6]Information Security and Cyber Security Unit, King Khalid University, Abha 62521, Saudi Arabia
[7]Department of Computer Science, FICT, BUITEMS, Quetta 87300, Pakistan
[8]Department of Computer Science, University of Limoges, 23204 Limoges, France
[9]Department of Computer Science, City University of Hong Kong, Hong Kong

Corresponding author: Konstantinos Markantonakis (k.markantonakis@rhul.ac.uk)

This work was supported by the Deanship of Research and Graduate Studies at King Khalid University through Small Group Research under Grant RGP1/349/45.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the College Research Ethics Committee (REC), Royal Holloway, University of London, U.K.

ABSTRACT Near-Field Communication (NFC) has enabled mobile devices to emulate contactless smart cards, which has also rendered them susceptible to relay attacks. Numerous countermeasures have been proposed that use ambient sensors as an anti-relay mechanism. However, there are concerns regarding their efficacy in time-critical scenarios, such as transport ticketing and contactless payments. This paper empirically and comprehensively evaluates whether ambient sensors are an effective anti-relay mechanism for such NFC-based contactless transactions. To this end, we examine 17 sensors available via the Android platform. Each sensor, where feasible, was used to record measurements in 1,000 contactless transactions with 252 users across four physical locations. We then conduct an extensive four-part evaluation using similarity metrics, traditional machine learning models, and deep learning methods used in existing work and beyond. We conclude that mobile ambient sensors are currently unsuitable for detecting relay attacks on NFC contactless transactions under realistic timing constraints, contrary to the suggestions and proposals made in existing work.

INDEX TERMS Near field communication (NFC), contactless transactions, relay attacks, ambient sensors, security, mobile payment.

I. INTRODUCTION

Contactless smart cards are susceptible to relay attacks [1], [2], [3], as are NFC-enabled mobile phones [4], [5], [6], [7]. A relay attack is a passive man-in-the-middle attack in which an attacker extends the distance between a genuine payment terminal (point-of-service) and genuine contactless smart card (or NFC-enabled mobile device). This attack can enable a malicious user to access services for which the genuine user is eligible, such as paying for goods or accessing a building with physical access controls. Quantifying the number of fraudulent activities where relay attacks are used

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

is a challenging task (on both smart card and NFC mobile phones). Evidence exists, however, that academic attacks on smart cards have been adopted by real-world criminals [8].

For contactless smart cards, a potentially effective countermeasure has been distance bounding protocols [9], [10]. For NFC-enabled phones, anti-relay mechanisms have focused on using ambient sensors for determining whether the terminal and payment device are in a shared location (Section II).

In this paper, we investigate the extent to which ambient sensors are actually effective under real-world timing constraints. We explore the sensors available through the Android platform and construct a test-bed environment (Section III) to evaluate their effectiveness as proximity detection mechanism for NFC-based contactless transactions (Section IV). The aim of this work is to provide empirical evidence of each ambient sensor's suitability as a proximity and relay attack detection mechanism (Section V).

A. OPERATIONAL ENVIRONMENT

We focus on NFC-based mobile applications that emulate traditional contactless smart cards, particularly for payments and transportation. In these domains, the use of ambient sensing must operate within strict proximity and transaction duration requirements. We describe these as follows:

1) *Proximity:* Two devices are considered to be in proximity of each other if they are physically present within a distance of 3-5cm [11], [12], [13].
2) *Transaction Duration:* The transaction must complete within 500ms. In accordance with the EMV specifications, the maximum permitted time in which a contactless payment transaction should complete is 500ms [14], [15], [16], [17]. For transport-related transactions, the performance requirements are similar, where transaction times can take approximately 300ms on average but must not exceed 500ms in total [18], [19], [20].

B. EVALUATION SCOPE

The effectiveness of a sensor-based proximity detection mechanism lies in the ability to discriminate between sensor measurements from genuine and illegitimate device pairs. The genuine pair is a terminal and mobile phone at <5cm from each other, while the illegitimate devices are not part of the intended transaction. Ultimately, the goal is to establish confidence that two devices are truly in close proximity intended by the NFC specifications (<5cm), rather than at a longer distance due to a relay attack. Moreover, the mechanism must also operate under a maximum transaction duration of 500ms.

In this paper, we broadly refer to NFC-based contactless transactions for payments due to their ubiquity and associated financial repercussions. However, our work is also relevant to other high-security NFC contactless services, e.g. physical access control systems. Our main contributions are as follows:

1) A test-bed implementation for evaluating various sensors on Android devices in time-critical NFC transactions.
2) A data analysis framework for evaluating ambient sensing as an anti-relay mechanism using traditional similarity metrics and machine and deep learning models.
3) An empirical evaluation of the effectiveness of ambient sensors as a proximity and relay attack detection mechanism. We show that ambient sensing is not suitable for high-value NFC transactions without compromising usability and security.

Our test-bed, data analysis and collected data sets are open-sourced in order to assist future research.[1]

II. USING AMBIENT SENSING FOR CONTACTLESS TRANSACTIONS

In this section, we briefly describe mobile contactless payments, relay attacks, and a generic architecture for deploying ambient sensing as a proximity detection mechanism for countering relay attacks.

A. CONTACTLESS MOBILE DEVICES AND RELAY ATTACKS

In an NFC-based mobile contactless transaction, a mobile handset is brought into the radio frequency range (<3-5cm) of a payment terminal through which a dialogue is initiated. Physical contact is not necessary during this process and, in most cases, a second factor of authentication is not required, e.g. biometrics or Personal Identification Number (PIN) [12].[2] From the terminal's perspective, this renders it difficult to ascertain whether a genuine or relay device is being used.

In a relay attack [5], [22], [23] (Fig. 1), an attacker presents a malicious payment terminal to the victim device and a separate masquerading payment device to the genuine payment terminal. The goal of the attacker is to extend the physical distance of the communication channel beyond that which it is intended. That is, beyond the <5cm intended for NFC-based transactions between a payment device and terminal.

If messages are successfully relayed without detection, an attacker can gain access to services using the victim's account; for example, the victim can remotely be charged for goods and services at the attacker's location. Distance-bounding protocols have been suggested as a potential countermeasure using tight challenge-response timings between communicating devices with near-identical execution platforms. The introduction of any significant latency suggests the presence of intermediate hops, i.e. the relay devices [9], [10], [24]. Unfortunately, distance-bounding protocols fail to generalise to mobile devices with different hardware-software configurations, such as system-on-chips (SoCs),

[1] Available at: https://github.com/AmbientSensorsEvaluation/Ambient-Sensors-Proximity-Evaluation.git

[2] Even the use of a PIN or biometric may not thwart relay attacks effectively, e.g. Mafia fraud attacks [21].

CPU frequencies, operating systems (OSs), and real-time clocks [25]. As a solution, numerous proposals—discussed in Section II-C—have argued that ambient sensors aboard mobile devices are an effective proximity detection method for thwarting relay attacks.

B. AMBIENT SENSORS FOR PROXIMITY DETECTION

An ambient sensor measures a physical attribute of its immediate surroundings, such as the temperature, light, or humidity. Modern smartphones and tablets are equipped with an array of such sensors (see Table 2 and Appendix A). The physical environment surrounding a smartphone, or a payment terminal, may provide a rich set of attributes that are unique to that location; for example, the sound and lighting of a quiet brightly lit room. This information is then used for assuring that only a genuine terminal and payment instrument pair are co-located. Relay attacks should then be detected since the ambient environment of the genuine terminal and payment instruments are different, which should be deducible from their sensing measurements. This paper casts doubt on the practical reality of this assumption.

Three models exist in which sensing-based proximity detection may be deployed. The entities are illustrated in Fig. 2 and the models are described as follows:

FIGURE 1. Overview of a relay attack.

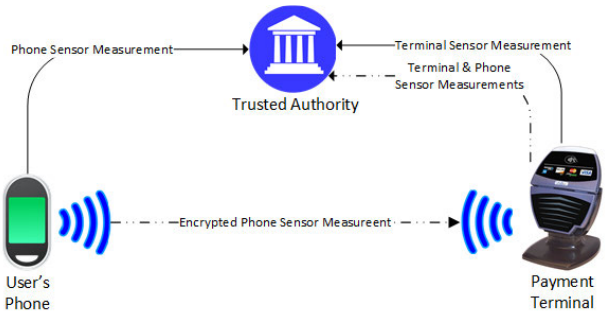


FIGURE 2. Generic deployments of ambient sensor-based proximity detection mechanisms.

1) **Independent Reporting**. Both the mobile device and payment terminal collect sensor measurements independently and transmit these to a trusted authority (depicted as solid lines in Fig. 2). The authority compares the sensor measurements, based on some predefined comparison algorithm with a set margin of error (threshold), and decides whether the devices are within sufficient proximity.
2) **Payment Terminal Dependent Reporting**. The mobile encrypts its sensor measurements with a shared key between itself and the trusted authority, and transmits the encrypted measurements to the payment terminal. Next, the terminal sends the mobile device's measurements and its own to the trusted authority for comparison (shown as a dot-dashed line in Fig. 2).
3) **Payment Terminal (Localised) Evaluation**. The mobile device securely transmits its own measurements to the payment terminal, which compares them with its own measurements to decide whether the phone is in proximity.

The deployment architecture falls under one of these scenarios irrespective of how the user interacts with the terminal.

C. RELATED WORK

Several proposals have been developed for sensing-based proximity and relay attack detection. These proposals are described forthwith.

Ma et al. [26] showed how location-related data, namely using GPS (Global Positioning System), can be used to determine the proximity of two NFC mobile phones. The authors used a ten-second window with location information collected every second, which was subsequently compared across various devices. The authors report a high success rate in identifying devices within close proximity.

Halevi et al. [27] demonstrated the suitability of ambient sound and light for proximity detection. Here, the authors analyse measurements collected for 2 and 30 seconds duration for light and audio respectively using a range of similarity comparison algorithms. Although the scenarios are identical, the transaction duration does not conform to industry requirements for NFC-based contactless mobile transactions (Section I-A). While the authors do not specify the number of transactions recorded at each location, the experiments show a high success rate of detecting co-located devices in various environments.

Varshavsky et al. [28] based their proximity detection mechanism on the shared radio environment of devices—the presence of WiFi access points and associated signal strengths—using the application of secure device pairing. This approach produced low error rates, recommending it as a proximity detection mechanism. While their paper did not focus on NFC-based mobile transactions, their techniques and methodology may still be applicable.

Urien and Piramuthu [29] use ambient temperature with an RFID/NFC authentication protocol for proximity detection. Using this method, they establish a secure channel by combining the timing channels in RFID, traditionally used in distance bounding protocols, in conjunction with ambient temperature. Their proposal, however, was neither implemented nor practically evaluated; there is no experimental evidence to judge its efficacy.

Mehrnezhad et al. [30] proposed the use of an accelerometer to provide assurance that the mobile phone is within

proximity of the payment terminal. Their proposal requires the user to tap the payment terminal twice in succession, after which the sensor streams of the device and the payment terminal are compared for similarity. It is difficult to deduce the exact time needed to complete a transaction in its entirety, but the authors use recording durations of 0.6–1.5 seconds.

TABLE 1. Sensing-based anti-relay mechanisms.

Work	Sensor Used	Sample Duration	Contactless Suitability
Ma et al. [26]	GPS	10 sec	Unlikely
Halevi et al. [27]	Audio	30 sec	Unlikely
	Light	2 sec	More Likely
Varshavsky et al. [28]	WiFi (Radio Waves)	1 sec	More Likely
Urien et al. [29]	Temperature	N/A	-
Mehrnezhad et al. [30]	Accelerometer	0.6 to 1.5 sec	More Likely
Truong et al. [31]	GPS Raw Data	120 sec	Unlikely
	Wifi	30 sec	Unlikely
	Ambient Audio	10 sec	Unlikely
	Bluetooth	12 sec	Unlikely
Shrestha et al. [32]	Temperature (T)	NA	Unlikely
	Precision Gas (G)	NA	Unlikely
	Humidity (H)	NA	Unlikely
	Altitude (A)	NA	Unlikely
	HA	NA	Unlikely
	HGA	NA	Unlikely
	THGA	NA	Unlikely
Choi et al. [33]	Audio	1–5 sec	More Likely

Truong et al. [31] evaluated four different sensors across recording durations of 10-120 seconds. Although the results were positive, such a long recording duration renders them unsuitable for realistic NFC-based mobile transactions. Moreover, the data collection set-up did not emulate a contactless transaction, either in the context of banking, transport or access control. However, the authors did discuss the impact of transaction duration on the real-world applicability of the results. For usability, the authors suggest that transaction durations should be minimised to the range of 5-15 seconds. They also conclud that measurements recorded beyond 10 seconds did not improve effectiveness.

Shrestha et al. [32] used bespoke hardware known as Sensordrone, with a number of ambient sensors, but did not evaluate the commodity ambient sensors available on commercial handsets, did not provide the sample duration, and only mentioned that data from each sensor was collected for a few seconds. It is difficult to evaluate the proposed technique in the context of NFC contactless mobile transactions in the banking and transport sector under their specified requirements. The results related to barometric air pressure were similar to what we have calculated. Sensors like Precision Gas and Altitude are not available on commodity off the shelf Android smart phones.

Choi et al. [33] propose an audio-based proximity detection for thwarting relay attacks on contactless car key fobs. The scheme measures the ambient sound surrounding an emulated key fob—using a Raspberry Pi with a microphone peripheral—and an automotive authentication module emulated by a laptop. Recordings of 1–5 seconds are evaluated in three static environments, which are compared for similarity using the Euclidean distance, correlation, and cosine similarity. The best case equal error rate (EER) was 0.0024 using correlation similarity. While the scheme is proposed as an anti-relay mechanism, the authors did not use a relay attack setup during its evaluation.

We summarise these proposals in Table 1, using measurement sampling durations to determine whether each approach is suitable for mobile contactless transactions. 'Unlikely' approaches require significant sampling times beyond reasonable limits for contactless transactions. Proposals with more reasonable durations are denoted 'More Likely' in Table 1. It is important to note that even these schemes may not be suitable for domains where strict transaction completion limits are imposed (see Section I-A). In these situations, the goal is to maximise customer throughput, e.g. at transport barriers and point-of-sales. An optimal transaction duration is, therefore, in the magnitude of milliseconds, not seconds.

We observe that no proposed relay detection scheme has evaluated sampling durations suitable for time-critical contactless transactions, i.e. $\leq$500ms. As a result, we do not repeat these proposals as the existing literature suggests, as their set-ups do not reflect conventional NFC-based transactions. This is due to the use of exceptionally long transaction durations [26], [27], [28], [31]; requiring the user to perform specific gestures, e.g. double-tapping a payment terminal [30]; or the use of non-standard hardware [32]. We evaluate whether ambient sensors in principal could be used for proximity and relay attack detection for NFC contactless transactions using off-the-shelf hardware and standard transaction behaviour. In this vein, Gurulian et al. [34] and Shepherd et al. [35] questioned the effectiveness of relay attack-based detection methods using ambient sensors. The authors utilise an extensive range of sensors available on Android devices in different environments, showing how sensor information between a legitimate device pair, i.e. a PoS terminal and payment handset, could not be differentiated from an illicit pair using traditional similarity learning and machine learning (i.e. data from a device pair in different locations). In this paper, we extend these results by applying deep learning via convolutional (CNNs) and recurrent neural networks (RNNs). The conclusions are further reinforced. Neither traditional machine learning nor deep learning in our study can discriminate between legitimate and illicit device pairs from sensor data captured within a realistic time period.

In wider literature, a large body of work has investigated using mutually acquired sensor measurements for secure device pairing and two-factor authentication. While these schemes use similar sensing modalities, e.g. magnetometers [36] and sound [37], these are separate problems with large device distances ($>$1m) and measurement times than NFC contactless transactions. We refer the reader to work

by Conti and Lal [38] for a comprehensive survey of these schemes.

III. FRAMEWORK FOR EVALUATING AMBIENT SENSORS

In this section, we describe the developed test-bed for testing, analysing, and evaluating the effectiveness of mobile sensors as a proximity detection mechanism. The results of the evaluation are presented in Section IV.

A. TEST-BED ARCHITECTURE

We developed an experimentation test-bed to collect empirical data for evaluating each sensor. Two applications were implemented and installed on a pair of Android devices: one emulating a payment terminal (PT) and the other acting as the payment instrument (PI), or a mobile phone. When the devices come sufficiently close, an NFC connection is established and both begin recording data using a specified sensor. After collecting measurements for 500ms, in line with the requirements specified in Section I-A, each device stores the recorded data in a local database. During field trials, one mobile phone was fixed as a terminal and the second mobile phone was free of any restrictions.

Fig. 3 shows this in greater detail. Bringing the two devices together (< 3cm) causes the PT application to send the first message to the PI over NFC, stating which sensor it uses in the transaction and a unique transaction ID. After this message is received by the PI, both applications initiate the process to record a sensor for 500ms.

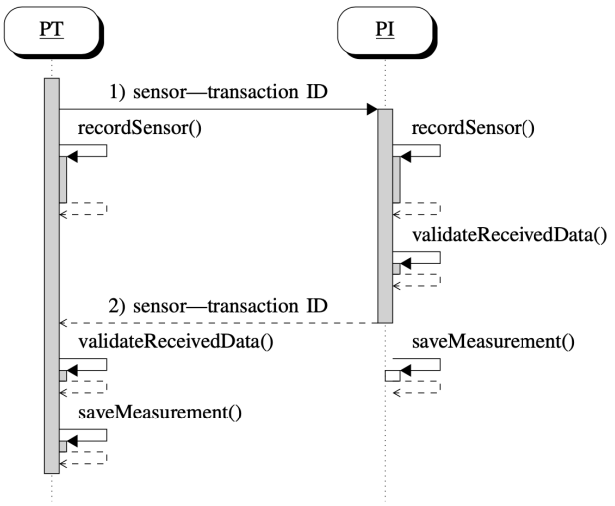


FIGURE 3. Measurement recording overview.

After collecting the measurements, the PI validates the data it received from the terminal—whether the transaction ID and chosen sensor match that of the terminal (shown in message one in Figure 3)—and returns an acceptance or rejection message accordingly. This validation process ensures that both devices were recording data from the same sensor. Finally, PT performs the same process, ensuring that both devices used the same transaction ID and recorded from the same sensor. The measurement is rejected in the event that devices recorded data for differing transaction IDs or sensors. Upon validation, the devices save the measurements in their local databases. The database is designed to hold measurements for each transaction, which are used in the off-line analysis of each sensor.

B. DATA COLLECTION FRAMEWORK

We tested each sensor in four different locations around the university—the lab, cafeteria, dining hall and library—to account for the influence of different physical locations on sensor measurements. A field trial was conducted in each location with 252 participants[3] who each conducted a varying number of transactions. Each participant used the PI provided by us and was given free reign with how they interacted with the PT for each transaction; they could tap it once, hold it extremely close without touching, tap and hold it to the device, etc. This was to closely replicate the conditions in which they would conduct a regular contactless transaction. The data collection at each of the locations was collected over an eight hours period (0900-1700hours) with irregular gaps between transactions over the course of four days.

TABLE 2. Sensor availability.

Sensors	Nexus 9	Nexus 5	SGS5 mini	SGS4
Accelerometer	✓	✓	✓	✓
Bluetooth	∗	∗	∗	∗
GRV†	✓	∗	✓	✗
GPS	∗	∗	∗	∗
Gyroscope	✓	✓	✓	✓
Magnetic Field	✓	✓	✓	✓
Network Location	✓	✓	✓	✓
Pressure	✓	∗	✗	∗
Rotation Vector	∗	∗	∗	✓
Sound	✓	✓	∗	∗
WiFi	∗	∗	∗	∗
Gravity	○	✓	✓	✓
Light	∗	✓	✓	✓
Linear Acceleration	○	✓	✓	✓
Proximity	✗	✓	✓	✓
Relative Humidity	✗	✗	∗	∗
Ambient Temperature	✗	✗	∗	∗

✓: Working properly. ✗: Not present on device. ∗: Technical limitations: no values returned in timeframe. ○: Returned only zero-values. †: Geomagnetic Rotation Vector.

Four devices were used in the experiments, forming two PT–PI pairs. The first pair consisted of two Nexus 9 tablets, while the second pair comprised two Android smartphones: a Nexus 5, assuming the role of the payment terminal, and a Samsung Galaxy S5 mini (SGS5 mini), which acted as the payment instrument. The availability of the sensors on each device is shown in Table 2.

[3]Informed Consent: No personal data was collected from individual participants. The mobile phone used by individual participants was provided by the researchers. Furthermore, the study's nature and the data collected during the study were explained to the participants beforehand. Based on this, they could continue to participate in the study if they consented.

A minimum of 1,000 transactions were recorded for each sensor, comprising measurement pairs for which both PT and PI have valid sensor data. Some sensors, such as Bluetooth, GPS, Rotation Vector and WiFi—although present on the devices—returned no or extremely few data points within the 500ms timeframe (>99% sensor failure[4]). Additionally, two sensors (humidity and temperature) are relatively uncommon among Android devices and none of our initial test devices possessed them. For completeness, we employed two Samsung Galaxy S4 (SGS4) smartphones that contained them; however, only 6% transactions using these sensors contained any measurements at all when recorded for 500ms. Consequently, we omitted these sensors from subsequent analysis.

The recorded sensor measurements were stored in XML form in each database. A new child element was created containing the sequence ID of the measurement, the timestamp (initialised to zero at the start of the transaction), along with the data for each returned measurement. The sequence ID consisted of the date and time the transaction occurred, the location in which it was captured, and a transaction ID. The transaction ID is a random, 7-byte string generated by the terminal used to link the measurements of each device to produce a PT–PI pair. Occasionally, the NFC connection was disrupted, primarily when the devices were moved apart before the transaction was completed. To address this, the transaction ID was used in conjunction with the sequence ID to detect and exclude these measurements prior to analysis.

IV. AMBIENT SENSOR EVALUATION

We now describe our data analysis and evaluation methodology using three approaches based on existing literature (see Section II-C) and beyond.

The first approach evaluates sensor data using simple, threshold-based similarity metrics used in existing literature. The second uses traditional supervised machine learning algorithms, e.g. logistic regression and support vector machines (SVMs), for classifying transaction pairs as (il-)legitimate. For the third approach, we examine the use of deep learning for classifying measurement transaction pairs.

After retrieving the databases from PT and PI, the set of all transactions, T, was produced using the shared IDs generated during data collection. Each transaction can be represented as a shared set of PT and PI measurements, PT_i and PI_i, with the same shared ID, i. We refer to this as $T_i = (PT_i, PI_i)$. Note that each device measures each sensor at potentially different time intervals (accounting for clock variances), which may produce an unknown total number of measurements for each device per transaction. As such, the number of measurements in PT_i is not necessarily that of PI_i. We discuss later how the effect of missing and inconsistent samples was mitigated using interpolation.

[4]Detailed in Section IV-A2 and Table 4.

A. METHOD 1: SIMILARITY ANALYSIS AND EVALUATION

This first evaluation method uses simple similarity analysis of (PT_i, PI_i). This is measured differently according to sensor type due to the differences in coordinate systems and dimensions used between sensors (see Appendix A). Measurements are recorded in three dimensions for the accelerometer, for example, while location returns a longitude-latitude pair on Earth. Due to this, we devised three methods of dealing with the diversity of reported measurements.

$$2r \arcsin\left(\sqrt{\sin^2\frac{\phi_2-\phi_1}{2} + \cos\phi_1 \cos\phi_2 \sin^2\frac{\lambda_2-\lambda_1}{2}}\right) \tag{1}$$

$$MAE(PT_i, PI_i) = \frac{1}{N}\sum_{j=0}^{N} |PT_{i,j} - PI_{i,j}| \tag{2}$$

$$corr(PT_i, PI_i) = \frac{covariance(PT_i, PI_i)}{\sigma_{PT_i} \cdot \sigma_{PI_i}} \tag{3}$$

$$M = \sqrt{x^2 + y^2 + z^2} \tag{4}$$

For the network location, we used the Haversine formula (Eq. 1), which measures the geographic distance between two latitude and longitude pairs, $\{(\phi_1, \lambda_1), (\phi_2, \lambda_2)\}$. In Eq. 1, r represents the radius of Earth. For the remaining sensors, the similarity of transaction measurements was measured using the *Mean Absolute Error* (MAE, Eq. 2) and *Correlation Coefficient* (Eq. 3), as used in [30], [33], between the signals of PT_i and PI_i. This was performed after linear interpolation to mitigate the effects of inconsistent clocks between devices.

To complicate matters, certain sensors—the accelerometer, gyroscope, magnetic field, rotation vector and GRV sensors—produce three-dimensional measurement vectors comprising x, y and z components. In these instances, the vector magnitude (Eq. 4) was used as a general-purpose method for producing a single, combined value prior to computing the MAE and correlation coefficient. Next, MAE was computed by applying Eq. 2 directly, while for correlation, this was found after measuring the covariance and computing the standard deviations, σ_{PT_i} and σ_{PI_i}, of the data points in PT_i and PI_i.

1) CALCULATING THE FPR, FNR AND EER

A Python application was developed for analysing the transaction measurements from the application databases; the NumPy and SciPy libraries [39] were used for implementing the similarity functions. Using this application, we calculated the $MAE(PT_i, PI_i)$ and $corr(PT_i, PI_i)$ for each successful transaction. Next, we calculated the False Positive Rate (FPR), False Negative Rate (FNR) and equal error rate (EER) of each sensor by the testing *MAE* and *corr* values of genuine pairs, (PT_i, PI_i), against the *MAE* and *corr* values of unauthorised pairs (PT_i, PI_j) with a threshold, t. An ideal similarity metric, V, would produce $V(PT_i, PI_i) < t$ and $V(PT_i, PI_j) > t$ for all possible pairs. We constructed these unauthorised pairs by exhaustively matching each PT_i

with every PI_j measurement belonging to another transaction ($i \neq j$). The FPR and FNR are calculated using Eq. 5, where FP, FN, TP and TN represent the number of False Positives, False Negatives, True Positives and True Negatives respectively for a given threshold.

$$FPR = \frac{FP}{FP + TN} \qquad FNR = \frac{FN}{FN + TP} \tag{5}$$

2) INDIVIDUAL SENSOR RESULTS

Our evaluation investigates to what extent legitimate and illegitimate transactions can be identified using these simple similarity metrics using a time-constrained transaction duration (500ms). For a transaction between two co-located devices, $MAE(PT_i, PI_i) \approx 0$ and $corr(PT_i, PI_i) \approx 1$, while for a PT and a PI device in differing locations, (PT_i, PI_j), the distance and correlation should be sufficiently distinct. This distinctiveness is determined by finding a suitable threshold, t, that permits all legitimate transactions while denying those which are illegitimate; that is, $V_i(PT_i, PI_i) < t$ and $V_{ij}(PT_i, PI_j) > t$. For each individual sensor, we aim to find an optimal value of t, its error rate and reliability, i.e. whether it collected measurements consistently and correctly across 1,000 transactions.

TABLE 3. Threshold-based analysis: EERs and thresholds.

Sensor (units)	t_{MAE}	EER_{MAE}	t_{corr}	EER_{corr}
Accelerometer (ms^{-2})	0.913	0.494	0.526	0.480
Gyroscope ($rads^{-1}$)	0.329	0.521	0.336	0.455
Magnetic Field (μT)	153.9	0.444	0.399	0.473
Rotation Vector (N/A)	0.493	0.330	0.470	0.472
Gravity (ms^{-2})	0.290	0.521	0.586	0.490
Light (lux)	26.54	0.367	0.714	0.488
Linear Accel. (ms^{-2})	0.569	0.482	0.064	0.536

We generate FPR and FNR curves for MAE and $corr$ for every sensor for which we were able to collect data. The point of intersection for these curves provides an EER threshold for MAE and $corr$: the rate at which the acceptance and rejection errors are equal.

Practically speaking, a single threshold would be used in a wide-scale deployment of a sensing-based proximity detection mechanism. The terminal (or third party) would store this threshold (Section II-B). If the similarity of the terminal's and device's sensor measurements was within this, then the transaction is assumed to be legitimate. However, setting a threshold of this nature invariably incurs some rate of false positives and false negatives. The intersection of FPR and FNR, or equal error rate (EER), is the point where equal consideration is given to illicit transactions being classed as genuine (false positives) and the rate at which genuine transactions are rejected (false negatives). A threshold with a higher FPR provides a large working space to the attacker, whereas a higher FNR will reduce the usability of the scheme, potentially frustrating consumers by rejecting legitimate transactions. Table 3 lists the optimum thresholds and associated EERs for each tested sensor.

TABLE 4. Sensor- and transaction-level reliability analysis.

Sensors	Total Transactions	Transaction Failures	Sensor Failures
Accelerometer	1025	13 (1.26%)	0 (0%)
Bluetooth	101	1 (0.99%)	99 (99.1%)
GRV	1019	8 (0.78%)	0 (0%)
GPS	101	1 (0.99%)	100 (99.10%)
Gyroscope	1022	11 (1.07%)	0 (0%)
Magnetic Field	1027	17 (1.65%)	0 (0%)
Network Location	1053	15 (1.42%)	960 (91.17%)
Pressure	1018	10 (0.98%)	0 (0%)
Rotation Vector	1023	14 (1.36%)	0 (0%)
Sound	1047	4 (0.38%)	0 (0%)
Gravity	1165	143 (12.27%)	0 (0%)
Light	1057	37 (3.50%)	0 (0%)
Linear Acceleration	1175	159 (13.53%)	3 (0.26%)
Proximity	1071	58 (5.41%)	0 (0%)
Ambient Temperature	50	0 (0%)	47 (94%)
Relative Humidity	50	0 (0%)	47 (94%)

As a further investigation, we also evaluated the reliability of the selected sensors. At times, transactions during field trials were not registered during this process, usually due to the user moving the handset away too quickly. This was the primary cause of transaction failures, i.e. no shared measurements between the PT and PI.[5] The sensor failure rate represents the situation when the transaction was successfully completed on both the PT and PI, but where one or both devices failed to record any data in the 500ms timeframe. The percentage of transaction failures relates to the total transactions, while sensor failures are measured with respect to the number of successful transactions. Table 4 presents our findings regarding the proportion of failed transactions and sensor failures. In general, the transaction failure rate implies potential usability issues when using each sensor for NFC-based contactless transactions, while the sensor failure rate reflects their reliability.

B. METHOD 2: MACHINE LEARNING ANALYSIS

The distance and correlation metrics used in the previous section give each pair of individual measurements $PT_{i,j}$ and $PI_{i,j}$ the same weighting when PT_i and PI_I are compared. However, it is conceivable that not all time slots in $PT_{i,j}$ and $PI_{i,j}$ are equally important when the task is to discriminate between genuine and unauthorised transaction pairs. Moreover, it is possible that discrimination becomes possible by modelling complex non-linear interactions between their individual differences, $|PT_{i,j} - PI_{i,j}|$, which cannot be captured by simple similarity measures used in existing work.

To investigate this, we applied several supervised machine learning algorithms to the problem, including algorithms that are able to model (in an approximate manner) arbitrary non-linear interactions given enough training data. The data for learning was created by treating each pair (PT_i, PI_i) for a

[5]Failed transactions were not included in the data analysis.

TABLE 5. Estimated EER for machine learning algorithms, obtained by repeating 10-fold cross-validation 10 times.

Sensor	Random Forest	Naive Bayes	Decision Tree	Logistic Regression	Support Vector Machine
Accelerometer	**0.277**±0.052	0.474±0.047	0.358±0.059	0.483±0.050	0.454±0.126
Gyroscope	**0.179**±0.041	0.354±0.059	0.228±0.049	0.356±0.055	0.288±0.045
Magnetic Field	**0.361**±0.055	0.400±0.053	0.389±0.063	0.421±0.061	0.385±0.053
Rotation Vector	**0.285**±0.052	0.327±0.055	0.317±0.073	0.353±0.050	0.325±0.050
Gravity	0.499±0.046	0.488±0.043	0.494±0.057	**0.484**±0.043	0.486±0.156
Light	0.361±0.059	0.369±0.058	**0.293**±0.149	0.407±0.054	0.351±0.054
Linear Acceleration	**0.307**±0.050	0.484±0.048	0.392±0.057	0.502±0.049	0.397±0.058

particular sensor as a labelled observation $(\vec{x}, y)$, where the label y is either *genuine* or *unauthorised* and the feature vector $\vec{x}$ consists of the individual differences $|PT_{i,j} - PI_{i,j}|$ for the pair (PT_i, PI_i). We use equal error rate to measure performance, using the confidence scores associated with each model's classifications to rank observations according to their estimated likelihood of being genuine transactions.

When applying machine learning to a classification problem like this, it is important to test the discriminative ability of the model inferred by the learning algorithm to a set of observations that have not been used. Given the number of observations available in our datasets, a single train-test experiment is not sufficient to establish a reliable estimate of equal error rate. A standard procedure is to perform 10-fold stratified cross-validation, where the data is shuffled and split into 10 disjoint test sets each containing the same number of observations. The data is also stratified so that the proportion of genuine and unauthorised transactions is the same in each set. Then the algorithm is run 10 times, once for each test set, where the observations not in the corresponding test set are used for training the model, and the observations in the test set are used to measure its equal error rate. This yields 10 estimates of equal error rate, which are averaged to obtain the final performance estimate. To reduce the variance of the performance estimate even further, we repeat 10-fold cross-validation 10 times, each time shuffling the data before it is split into 10 test sets. This yields 100 estimates of equal error rate and we report the mean and standard deviation of these estimates for each learning algorithm and sensor.

Table 5 shows results for the six learning algorithms we evaluated, including both parametric and non-parametric approaches, as implemented in the WEKA machine learning software [40]. We used default parameter settings for the learning algorithms unless otherwise specified. The random forest method [41] learns an ensemble classifier consisting of 100 semi-random decision trees from bootstrap replicates of the training data. This classification method is able to model arbitrarily complex interactions and is known to be a general-purpose approach that performs well without parameter tuning. The Naïve Bayes classifier fits a multivariate Gaussian distribution with a diagonal covariance matrix to the data for each classification (genuine *vs.* unauthorised), thus assuming conditional independence of the features in the data, and uses Bayes' rule to obtain class probability estimates. Logistic regression fits a linear model using maximum conditional likelihood. The widely used C4.5 [42] algorithm is used to grow decision tree classifiers. We also include linear classification using support vector machines, which are trained using the SMO [43] algorithm. A logistic regression model is fit to the output of the support vector machine to obtain class probability estimates. The last learning method in our collection is a multilayer perceptron, an artificial neural network (ANN) variant, with one hidden layer containing 10 units, which is trained using the *MLPClassifier* method in WEKA.

The results in Table 5 are largely in line with those observed earlier; the lowest equal error rate for each sensor is shown in bold. No useful discriminative signal appears to be present in the accelerometer, geometric rotation vector (GRV), gyroscope, light, linear acceleration, gravity, and proximity data. Decision tree-based methods give the best results for the remaining sensors. Magnetic field, rotation vector, and sound data provide some discriminative ability, but the EER remains close to 30%. The best result is obtained on the pressure data, with an EER of approximately 10%. Pressure was also the most informative sensor in the earlier experiments, with 27% EER for the *MAE* distance metric. Although the result obtained using tree-based learning is substantially better, the discrimination is still significantly too inaccurate to be used as a practical proximity detection mechanism.

C. METHOD 3: DEEP LEARNING ANALYSIS USING FULLY-CONNECTED ARTIFICIAL NEURAL NETWORKS (ANNS) AND SENSOR COMBINATIONS

While the use of traditional machine learning methods showed some level of discriminative power (Section IV-B), none were accurate enough to be used in practice. Our third approach explored the application of deep learning in the hope that a more abstract and complex representation of the data would lead to more promising results. As with Method 2, the problem of identifying transactions as either genuine or relayed based on sensor readings is regarded as a binary (supervised) classification problem. There are many successful examples of deep learning techniques being applied to classification problems in academic literature [44]. We evaluate the applicability of such techniques in the following sub-sections.

1) OVERVIEW

Both machine and deep learning exploit non-linear interactions present in the data; however, their methodologies and results differ due to different approaches during feature extraction and classification. Deep learning uses representation learning to translate observations about the data into an internal representation by changing the weights of neurons in the model, which is used for inferring the class label of input vectors [45]. This is opposed to traditional machine learning techniques, which require manual feature extraction before the training phase. The models' basic architecture also differs as a result. Traditional ML models can be described as a 'shallow' network, e.g. MLP ANNs, or no network at all (like Bayesian or linear classifiers such as SVMs). In comparison, deep learning networks are architected using interconnected intermediate layers, allowing them to encode data as a hierarchy of representations, or abstracted concepts [46]. In recent years, well-designed and well-trained deep learning models have outperformed traditional ML counterparts in various domains, including natural language processing (NLP) [47], [48] and computer vision tasks [49], [50], [51].

2) DATA PREPARATION

As with Method 2, supervised learning methodologies require all records used for training to have a label $(\vec{x}, y)$, where $\vec{x}$ is the observation and y is the label. To that end, all legitimate PT and PI transactions were labelled as *genuine*, while illicit pairs were labelled as *unauthorised*.

Instead of using the original dataset, a factorial combination of all sensor readings per transaction record was generated for two reasons. Firstly, deep learning models generally perform better when trained on more data [52]. By generating the factorial combination of readings, the total number of records was largely increased without modifying the information encoded in the original dataset. Secondly, it is very likely that the most discriminating factor in identifying a record as either genuine or unauthorised is not a single sensor, but a combination of sensors. The factorial combination ensures that the (unknown) most discriminating combinations of sensors were included in the training dataset.

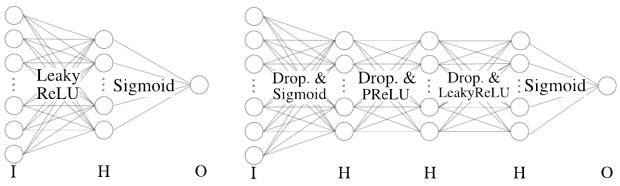
To train and test the model described in the next section, the collected data was split in a random, non-contiguous fashion. Initially, 60% of the data was allocated to the training dataset, 20% to the validation dataset, and the final 20% forms the testing dataset. The final models were trained on a 80% training dataset and tested on the remaining 20%.

3) MODEL CONFIGURATION

The following results and descriptions are based on PyTorch [53] implementations of DL models. Two models were developed, tested and evaluated on the available datasets. Each of the models was put through three training iterations designed to test, evaluate and select the best performing sensor combinations for further analysis, as described in the following.

- I_1: For the first iteration, the models are trained on every dataset and their accuracy is stored for analysis. Datasets that reach an accuracy of 70% or above proceeded to the next iteration.
- I_2: The second iteration tunes the model's hyperparameters (specifically the number of hidden units and the learning rate) via trial and error on the best-performing sensor combinations and, again, stores their results. The accuracy threshold is higher than previously, models trained on the datasets now need to reach an accuracy of 75% or above to continued to the next iteration.
- I_3: Finally, the highest-performing model configurations are trained and tested on the selected datasets ten times to obtain the final performance estimate by averaging their respective results.

The developed models are both feed-forward networks, comprising an input layer, a number of hidden layers, and an output layer that feeds into a sigmoid function to generate a number between 0 (illegitimate) and 1 (genuine). This allows us to not only acquire the model's classification of a record, but also the certainty of the model's classification, i.e. accuracy and loss. The difference between the models lies in the number of hidden layers and their activation functions (see Fig. 4). The first model, M_1, is somewhat simple with only one hidden layer connected through a non-linear function. The second model, M_2, incorporates principles that are known to generally increase model performance. This includes regularisation through dropout layers to avoid overfitting [52], more hidden layers to increase the model's complexity and reduce underfitting [54], and disparate activation functions to reduce the drawbacks of any single activation function [52].



(a) M_1 (b) M_2

FIGURE 4. DL feed-forward model architectures. $\boldsymbol{I} \in \mathbb{R}^{50}$, $\boldsymbol{H} \in \mathbb{R}^{[10,50]}$, and $\boldsymbol{O} \in \mathbb{R}^{1}$ are the input, hidden, and output layers.

4) RESULTS DISCUSSION

The models are evaluated based on the training accuracy achieved when trained on different sensor configurations. Both models slightly outperformed the machine learning techniques applied in Section IV-B. While model M_1 had marginally more accurate results, as shown in Table 6, model M_2 was generally more consistent in its high accuracy results. In iteration I_1, model M_1 had 303 datasets over the 70% threshold while model M_2 had 475. Model M_1 also outperformed model M_2 on average in iteration I_2, where it

had 72 datasets *vs.* 59 over 70% accuracy. However, only one trained M_2 model succeeded in reaching the threshold of 75%.

Using six combined sensors was the most prevalent in the high-performing datasets, with five and seven combined sensors tied in second place for both models. It is worth noting that the highest performing datasets for model M_1 are, for the most part, the same three sensors in varying orders and with slightly different model configurations, namely the linear acceleration (*Lin*), gyroscope (*Gyr*), and rotation vector (*Rot*) sensors. In contrast, the highest accuracy datasets for M_2 are larger (six to seven sensors) and more varied. This could be ascribed to the larger architecture of model M_2, which enables the formation of more complex data representations from a greater number of features.

Furthermore, there is a trend towards more accurate results based on the models' configurations. Interestingly, while 50 hidden nodes and a learning rate of 0.01 is the most common configuration for high-performing datasets with M_2 models, the top five all have a configuration of 20 hidden nodes with a 0.01 learning rate (see Table 6). For M_1 models, the most common configuration closely aligns with the top results. A learning rate of 0.01 and 20 or 50 hidden nodes are almost tied, with 0.1 learning rate and 50 hidden nodes close behind. Despite this, the results are still not accurate enough for practical use. Notwithstanding, there are several sensor combinations that consistently result in >65% accuracy, which signals some discriminative power for determining whether a contactless transaction is genuine or illegitimate.

TABLE 6. Best-performing ANN EER results.

Model	Sensors	Config	Iteration		
		hn^*	I_1	I_2	I_3
M_1	**LinGyrRot**	20	0.281 ± 0.016	0.283 ± 0.085	$\mathbf{0.317} \pm 0.083$
	LinGyrRot	50	0.284 ± 0.119	0.280 ± 0.027	$\mathbf{0.313} \pm 0.038$
	LinRotGyr	50	0.291 ± 0.024	0.309 ± 0.000	$\mathbf{0.311} \pm 0.015$
	RotLinGraAccGyr	50	0.300 ± 0.011	0.315 ± 0.104	$\mathbf{0.310} \pm 0.021$
	LinRotGyr	20	0.288 ± 0.074	0.291 ± 0.050	$\mathbf{0.302} \pm 0.048$
M_2	**GyrLinGraAccLigRot**	20	0.290 ± 0.034	0.289 ± 0.018	$\mathbf{0.317} \pm 0.062$
	MagLigRotGyrLinAccGra	20	0.289 ± 0.013	0.317 ± 0.030	$\mathbf{0.316} \pm 0.011$
	RotGyrAccLinLigMag	20	0.277 ± 0.038	0.291 ± 0.029	$\mathbf{0.299} \pm 0.081$
	MagGyrLigAccLinRotGra	20	0.300 ± 0.000	0.298 ± 0.000	$\mathbf{0.313} \pm 0.081$
	GyrGraAccLinRotMagLig	20	0.305 ± 0.091	0.301 ± 0.084	$\mathbf{0.305} \pm 0.059$

Acc: Accelerometer, Gra: Gravity, Gyr: Gyroscope, Lig: Light, Lin: Linear Acceleration, Mag: Magnetic Field, Rot: Rotation Vector.
*The number of hidden nodes, $hn \in [10, 50]$.

D. METHOD 4: A DEEP LEARNING ANALYSIS USING CONVOLUTIONAL AND RECURRENT NEURAL NETWORKS AND SENSOR COMBINATIONS

A major challenge of traditional, fully connected MLP ANNs is the explosion in the number of parameters that require training, which can become computational infeasible for networks with multiple hidden layers and input sizes [52]. Alternative architectures have been explored in recent years with enormous success in the classification of time-series sensor data, particularly using convolutional and recurrent neural networks (CNNs and RNNs) [55], [56], [57]. CNNs solve the optimisation problem of MLP ANNs by reducing the number of potential parameters through the use of partial connections (i.e. layers using convolutional operations), downsampling (max. pooling), and weight sharing between layers in addition to one or more fully connected layers. RNNs, in contrast, consider temporal dependencies in the input using directed cycles, where the current time-step, t, depends on the network state at the previous step, $t-1$. The output of a recurrent neuron state, s, is trained as a function of the inputs of the previous step, $s(t) = f(s(t-1), x(t))$. RNNs have found particular utility in the modelling of time-series sequences for which our problem is well suited.

1) CNN APPROACH

In this work, we explore results using three sequential CNN architectures. After the initial input layer, the following network architectures were evaluated: ① a convolution layer followed by a max pooling layer and a fully connected layer; ② a network comprising a convolution, max pooling, convolution, max pooling, and fully connected layers; and ③ applying dropout layers in ② after each max pooling layer. The architectures are completed with a fully connected binary output layer, indicating whether a sample is legitimate (i.e. taken from devices within true proximity) or illegitimate (taken from the relay attack pair).

2) RNN APPROACH

We similarly explore three different (sequential) architectures as a first step in evaluating the use of RNNs for sensor-based relay attack detection. We opted for widely used long short-term memory (LSTM) units to address the vanishing gradients problem with vanilla RNN architectures that prevents the back-propagation of errors [58]. Similarly, after the initial input layer, we evaluated the following architectures: ① a recurrent/LSTM layer and a fully connected output layer; ② an LSTM layer, followed by a dropout layer, and a fully connected output layer; and ③ LSTM, dropout, second LSTM, dropout, and a final fully connected layer. comprising a convolution, max pooling, convolution, max pooling, and fully connected layers; and ③ applying dropout layers in ② after each max pooling layer.

3) EVALUATION AND RESULTS

From an implementation perspective, we employed the Keras framework by Chollet [59] using the PyTorch backend. We utilise the same data set as the previous sections, divided into a 60:20:20 train, validation, and test set ratio; the final accuracy scores were made against the test set. To scope our study, we used the datasets conforming to the best-performing sensors identified in §IV-C. For training each architecture, we used a workstation with an Intel i7-6700k CPU (quad-core, 8M cache, 4.0GHz base clock frequency), 32GB RAM, and an NVIDIA 970 GTX on 64-bit Ubuntu 22.10, taking approximately 2.5hrs to train all architectures until convergence (minimising binary cross-entropy using Keras' implementation of the standard Adam [60] optimiser).

TABLE 7. CNN and RNN architecture EER results.

Model	Sensors	Architecture 1	Architecture 2	Architecture 3
CNN	**LinGyrRot**	0.299 ± 0.039	0.256 ± 0.028	0.285 ± 0.101
	RotLinGraAccGyr	0.291 ± 0.113	0.273 ± 0.046	0.316 ± 0.050
	GyrLinGraAccLigRot	0.302 ± 0.147	0.287 ± 0.084	0.314 ± 0.055
	MagLigRotGyrLinAccGra	**0.284** ± 0.011	0.264 ± 0.035	**0.246** ± 0.020
	RotGyrAccLinLigMag	0.329 ± 0.145	0.277 ± 0.131	0.266 ± 0.088
	MagGyrLigAccLinRotGra	0.313 ± 0.096	**0.253** ± 0.050	0.730 ± 0.102
	GyrGraAccLinRotMagLig	0.335 ± 0.086	0.274 ± 0.145	0.301 ± 0.036
RNN	**LinGyrRot**	0.360 ± 0.064	0.287 ± 0.124	**0.275** ± 0.013
	RotLinGraAccGyr	**0.298** ± 0.038	**0.273** ± 0.015	0.278 ± 0.000
	GyrLinGraAccLigRot	0.339 ± 0.125	0.331 ± 0.086	0.285 ± 0.142
	MagLigRotGyrLinAccGra	0.312 ± 0.034	0.302 ± 0.042	0.334 ± 0.025
	RotGyrAccLinLigMag	0.325 ± 0.137	0.308 ± 0.145	0.282 ± 0.138
	MagGyrLigAccLinRotGra	0.344 ± 0.091	0.347 ± 0.088	0.336 ± 0.090
	GyrGraAccLinRotMagLig	0.328 ± 0.046	0.294 ± 0.037	0.286 ± 0.042

Acc: Accelerometer, Gra: Gravity, Gyr: Gyroscope, Lig: Light, Lin: Linear Acceleration, Mag: Magnetic Field, Rot: Rotation Vector.

The accuracy results for the best-performing sensor combination datasets are presented in Table 7. In general, the highest accuracy was achieved using the second CNN and third RNN architectures. In both cases, the classification scores are marginally improved over the ANN analysis in §IV-C. Nevertheless, the best-performing sensor combinations yielded an EER of only 0.246 (CNN; Architecture 3; MagLigRotGyrLinAccGra) and 0.273 (RNN; Architecture 2; RotLinGraAccGyr). It is important to note that no work has directly addressed ambient sensor-based relay attack detection using CNNs or RNNs. As such, the analysed architectures represent an exploratory study, as opposed to concrete proposals. Indeed, it is conceivable that the development of a more sophisticated, bespoke deep learning architecture may lead to more promising results. We pose the challenge of creating an effective architecture as an open challenge to researchers in the field.

Open Challenge: *With enough data, can a sufficiently complex learning model solve sensor-based relay attack detection using commercial mobile devices?*

V. OUTCOME AND FUTURE DIRECTIONS

On conventional mobile devices, user authentication mechanisms are set with very high thresholds with respect to FPR, FNR, and EER. For example, Android's biometric security requirements stipulate that biometric system, at worst, has a false acceptance rate (FAR) of 1 in 50,000 illicit samples, and rejects legitimate samples at a rate of, at most, 1 in 10 times [61]. Under these definitions, we recognise that every sensor and algorithm combination failed to reach these levels+ of security. Even under a liberal setting of EER = 0.5, i.e. accepting almost every 1 in 20 illicit transactions (or rejecting legitimate ones), this threshold is not met. Based on our analysis, it is difficult to recommend any of the sensors individually for a high security deployment application, such as banking and transport. These sensors, however, might be appropriate for low-security access control, but we recommend that a thorough analysis of the sensors and their performance is performed prior to deployment. For further research, we suggest that an acceptable EER is >0.98 in accordance with existing authentication systems for security and usability.

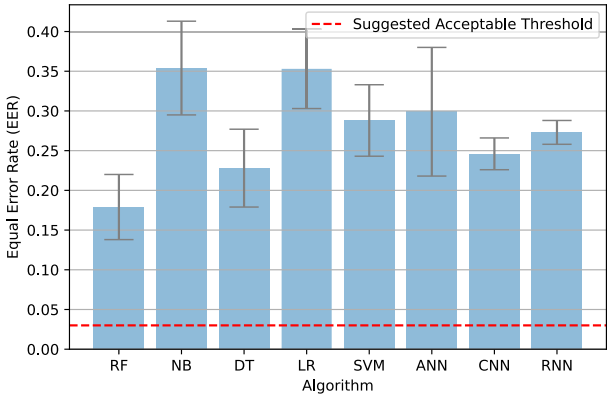


FIGURE 5. Best-case EER results across all sensors for each algorithm (lower EERs indicate greater effectiveness).

There are two potential reasons why previous research has achieved more promising results. The first is due to their relatively limited field trials—a small user base and a limited number of locations—and, secondly, the widespread use of significantly larger and unrealistic transaction durations. In our experiments, the total sampling duration was in line with standard performance requirements: 500ms for transport ticketing and EMV-based payments, as described in Section I-A. Along with banking, transportation is one of the biggest application areas of contactless smart cards; in this domain, the average duration for a transaction is even lower (300–400ms [20]). Imposing a 500ms limit in our experiments is thus an upper-bound of the requirements for two major areas where NFC-based contactless transactions may be used.

A potential future research direction is to investigate the extent to which measurements can be recorded *before* the actual transaction occurs. However, we have some reservations about this proposal. Firstly, it requires users to preempt transactions, which may an additional task to be performed before the mobile device can be used for NFC-based transactions. This could significantly detract from usability, which is a major driver behind the use of contactless transactions. Secondly, it may not provide adequate proximity detection assurances if the user is several meters away from the terminal when the recording is initiated. This opens the possibility for close-quarters relay attacks to be executed successfully; for example, by an attacker in the same store line. As a final challenge, it is well-known that long sensor polling has a significantly detrimental impact on device battery life [62]. This excess battery use, accounting for the millions of NFC contactless transactions that occur each day, may cause major energy wastage on aggregate in a large-scale deployment.

Researchers may argue that sensor-based proximity detection is redundant or unnecessary if a PIN or biometric is required to use a payment application. We do not agree with

the argument. In the relay attack variant known as a Mafia Attack [30], the attacker deploys a malicious terminal in order to deceive victim devices. In this scenario, a PIN or biometric cannot protect against relay attacks if the victim believes they are interacting with a genuine terminal while their credentials are being relayed to another terminal.

During this work, we realised that mobile sensing platforms are unlikely to be suitable for time-critical proximity detection mechanisms. Variations in sensor measurements, e.g. jitter; sensor availability; the effect of competing applications using the same sensors; and differences in minimum sampling rates may vary significantly across devices. We posit that mobile sensors have a considerable way to go before meeting the performance requirements needed to underpin time-critical proximity detection mechanisms.

As such, relay attacks are still a threat to NFC-based transactions, despite the recent spate of sensing-based countermeasures. These proposals might be suitable for low-security applications with long transaction durations. This is certainly not the case for mobile payments, transport ticketing, and high-security access control. We note that, after the evaluation described in this paper, we conducted preliminary experiments using a Samsung Galaxy S4 (Model: GT-I9505) and an Apple iPhone 6S. In both tests, the outcomes were even less promising, providing further evidence for our results. If sensing-based countermeasures are to be used, then deployment authorities ought to consider the risks highlighted in this paper, including the reported EERs and reliability rates.

VI. CONCLUSION

The use of sensor-based proximity detection mechanisms for NFC-based mobile services is expanding. This paper aimed to evaluate a range of sensors present on modern mobile devices and determining which sensors, if any, would be suitable under realistic time-critical NFC transactions.

To this end, we evaluated 17 sensors in total, including those proposed in existing literature, and many sensors that have not yet been examined. We developed a test-bed that was used to record sensor measurements from 1,000 NFC contactless transactions with 252 users across four locations. After this, a comprehensive multi-faceted evaluation was conducted using similarSecureCommity metrics, traditional machine learning, and deep learning models. These methods employed techniques used in existing work and beyond.

At present, we cannot recommend mobile sensors for proximity and relay attack detection in time-critical NFC transactions. Based on the experimental evidence presented in Section IV, the use of sensors for this purpose is likely to result in serious security and usability issues. However, we did identify potential discriminative power for certain sensors and combinations thereof. We pose an open challenge to the research community to develop a tailored statistical framework for effectively capturing this.

It is important to note that this work concentrates on NFC-enabled mobile devices that emulate traditional smart card services, such as transportation ticketing and mobile payments. Moreover, the transaction time limit and operating distance are not set arbitrarily, but rather in compliance with industry requirements stipulated by EMV and transportation bodies. It is neither evaluated nor claimed that existing proposals (Section II-C) cannot be useful for alternative domains where tight transaction durations and distance-bounding requirements are not used. Finally, we have publicly released the source code of our test-bed, along with our collected data sets, for open scrutiny and further analysis.

APPENDIX A
SENSOR DESCRIPTIONS

A. ACCELEROMETER

Deployed in most modern smartphones, the accelerometer measures the acceleration applied to the device about its x, y and z axes; its units are metres per second per second (ms^{-2}).

B. BLUETOOTH

A technology that facilitates wireless communication and operates in the ISM band centred at 2.4GHz. As a proximity sensor, we measure the Bluetooth devices in the vicinity (SSIDs and MAC addresses).

C. GEOMAGNETIC ROTATION VECTOR (GRV)

Measures the rotation of the device using the device's magnetometer and accelerometer; it returns a vector containing the angles that the device is rotated about the x, y and z axes.

D. GLOBAL POSITIONING SYSTEM (GPS)

A satellite-based global positioning mechanism. A latitude and longitude pair is returned, representing a geographical location on Earth.

E. GRAVITY

Measures the effect of Earth's gravity on the device in metres per second per second (ms^{-2}).

F. GYROSCOPE

Measures the rate of rotation of the device about the x, y and z axes; Android's standard units are radians per second ($rads^{-1}$).

G. LIGHT

Gauges the illumination of the device's surroundings, measured in lux on the Android platform.

H. LINEAR ACCELERATION

Calculates the vector magnitude of the device's acceleration in all directions; its units are metres per second per second (ms^{-2}).

I. MAGNETIC FIELD

Detects the effect of nearby magnetic emanations along three perpendicular axes x, y and z. Android measures these values in micro-teslas (μT).

J. NETWORK LOCATION

The location of the connected network point. A latitude and longitude pair is returned, representing a geographical location on Earth.

K. PRESSURE

Measures the atmospheric pressure surrounding the mobile handset in hectopascals (hPa).

L. PROXIMITY

The proximity sensors detects the distance of the device to a closely located object in the sensor's line of sight, measured in centimetres. On many devices, the sensor returns only a boolean value denoting whether or not an object is in close proximity to the device.

M. ROTATION VECTOR

Rotation vector is a software sensor, similar to the GRV, but also incorporates the gyroscope. The returned values represent the angles which the device has rotated through the x, y and z axes.

N. SOUND

We used the device's microphone to record the ambient noise in the vicinity of the device and calculated the maximum observed amplitude.

O. WIFI

We detected nearby WiFi access points in the device's vicinity using their MAC addresses and ESSIDs.

ACKNOWLEDGMENT

The author sincerely thank the students and users who participated in their study, providing valuable sample measurements. Additionally, they are grateful to the Ph.D. students from the Smart Card Centre and Information Security Group for their time and contributions during the data collection phase.

REFERENCES

[1] G. P. Hancke, K. E. Mayes, and K. Markantonakis, "Confidence in smart token proximity: Relay attacks revisited," *Comput. Secur.*, vol. 28, no. 7, pp. 615–627, Oct. 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404809000595

[2] G. P. Hancke, "Practical attacks on proximity identification systems," in *Proc. IEEE Symp. Secur. Privacy*, May 2006, pp. 328–333. [Online]. Available: http://dblp.uni-trier.de/db/conf/sp/sp2006.html#Hancke06

[3] Z. Kfir and A. Wool, "Picking virtual pockets using relay attacks on contactless smartcard," in *Proc. 1st Int. Conf. Secur. Privacy Emerg. Areas Commun. Netw. (SECURECOMM)*, vol. 2523, 2005, pp. 47–58, doi: 10.1109/securecomm.2005.32.

[4] L. Francis, G. P. Hancke, K. Mayes, and K. Markantonakis, "Practical NFC peer-to-peer relay attack using mobile phones," in *Proc. RFIDSec*, S. B. O. Yalcin, Ed., 2010, pp. 35–49. [Online]. Available: http://dblp.uni-trier.de/db/conf/rfidsec/rfidsec2010.html#FrancisHMM10

[5] L. Francis, G. P. Hancke, K. Mayes, and K. Markantonakis, "Practical relay attack on contactless transactions by using NFC mobile phones," *IACR Cryptol. ePrint Arch.*, vol. 2011, p. 618, Dec. 2011. [Online]. Available: http://dblp.uni-trier.de/db/journals/iacr/iacr2011.html#FrancisHMM11

[6] G. Madlmayr, J. Langer, C. Kantner, and J. Scharinger, "NFC devices: Security and privacy," in *Proc. Int. Conf. Availability, Rel. Secur.*, 2008, pp. 642–647.

[7] M. Roland, J. Langer, and J. Scharinger, "Applying relay attacks to Google wallet," in *Proc. 5th Int. Workshop Near Field Commun. (NFC)*, Feb. 2013, pp. 1–6, doi: 10.1109/NFC.2013.6482441.

[8] H. Ferradi, R. Geraud, D. Naccache, and A. Tria, "When organized crime applies academic results," in *Proc. IACR Cryptol. ePrint Archive*, 2015, p. 20.

[9] G. P. Hancke and M. G. Kuhn, "An RFID distance bounding protocol," in *Security and Privacy for Emerging Areas in Communications Networks* (SecureComm). Washington, DC, USA: IEEE, 2005, pp. 67–73. [Online]. Available: http://dx.doi.org/10.1109/SECURECOMM.2005.56

[10] R. Trujillo-Rasua, B. Martin, and G. Avoine, "The Poulidor distance-bounding protocol," in *Radio Frequency Identification: Security and Privacy Issues*. Cham, Switzerland: Springer, 2010, pp. 239–257.

[11] *How to Optimize the Consumer Contactless Experience? The Perfect Tab*, MasterCard, Harrison, NY, USA, Mar. 2014.

[12] *EMV Contactless Specifications for Payment Systems: Book A—Architecture and General Requirements*, EMVCo, Foster City, CA, USA, Specification Version 2.5, Mar. 2015.

[13] *EMV and NFC: Complementary Technologies That Deliver Secure Payments and Value-Added Functionality*, Smart Card Alliance, West Windsor Township, NJ, USA, Oct. 2012.

[14] *The Future of Ticketing: Paying for Public Transport Journeys Using Visa Cards in the 21st Century*, VISA, Fresno, CA, USA, Jan. 2013.

[15] *MasterCard Contactless Performance Requirement*, MasterCard, Harrison, NY, USA, Mar. 2014.

[16] *EMV Contactless Specifications for Payment Systems: Book D—EMV Contactless Communication Protocol Specification*, EMVCo, Foster City, CA, USA, Specification Version 2.6, Mar. 2016.

[17] *Transactions Acceptance Device Guide (TADG)*, VISA, Fresno, CA, USA, Specification Version 3.0, May 2015.

[18] *Transit and Contactless Open Payments: An Emerging Approach for Fare Collection*, Smart Card Alliance Transportation Council, NJ, USA, White Paper TC-11002, Nov. 2011.

[19] M. Emms, B. Arief, L. Freitas, J. Hannon, and A. van Moorsel, "Harvesting high value foreign currency transactions from EMV contactless credit cards without the PIN," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, vol. 8, Nov. 2014, pp. 716–726.

[20] MasterCard. (2018). *Contactless Payments Travel Well in London*. [Online]. Available: https://www.mastercard.us/content/dam/mccom/en-us/documents/ContactlessTFLLondonCaseStudy.pdf

[21] C. Cremers, K. B. Rasmussen, B. Schmidt, and S. Capkun, "Distance hijacking attacks on distance bounding protocols," in *Proc. IEEE Symp. Secur. Privacy*, May 2012, pp. 113–127, doi: 10.1109/SP.2012.17.

[22] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, "Practical NFC peer-to-peer relay attack using mobile phones," in *Proc. Int. Workshop Radio Freq. Identificat. Security Privacy Issues*. Berlin, Germany: Springer, 2010, pp. 35–49. [Online]. Available: http://dl.acm.org/citation.cfm?id=1926325.1926331

[23] R. Verdult and F. Kooman, "Practical attacks on NFC enabled cell phones," in *Proc. 3rd Int. Workshop Near Field Commun.*, Feb. 2011, pp. 77–82, doi: 10.1109/NFC.2011.16.

[24] S. Brands and D. Chaum, "Distance-bounding protocols," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 765, T. Helleseth, Ed. Cham, Switzerland: Springer, 1994, pp. 344–359, doi: 10.1007/3-540-48285-7_30.

[25] I. Gurulian, "On enhancing the security of time constrained mobile contactless transactions," Ph.D. dissertation, Roy. Holloway, Univ. London, London, U.K., 2019.

[26] D. Ma, N. Saxena, T. Xiang, and Y. Zhu, "Location-aware and safer cards: Enhancing RFID security and privacy via location sensing," *IEEE Trans. Dependable Secure Comput.*, vol. 10, no. 2, pp. 57–69, Mar. 2013, doi: 10.1109/TDSC.2012.89.

[27] T. Halevi, D. Ma, N. Saxena, and T. Xiang, "Secure proximity detection for NFC devices based on ambient sensor data," in *Proc. Eur. Symp. Res. Comput. Secur.*, in Lecture Notes in Computer Science, vol. 7459, S. Foresti, M. Yung, and F. Martinelli, Eds. Cham, Switzerland: Springer, 2012, pp. 379–396, doi: 10.1007/978-3-642-33167-1_22.

[28] A. Varshavsky, A. Scannell, A. LaMarca, and E. de Lara, "Amigo: Proximity-based authentication of mobile devices," in *Ubiquitous Computing* (Lecture Notes in Computer Science), vol. 4717, J. Krumm, G. Abowd, A. Seneviratne, and T. Strang, Eds. Cham, Switzerland: Springer, 2007, pp. 253–270, doi: 10.1007/978-3-540-74853-3_15.

[29] P. Urien and S. Piramuthu, "Elliptic curve-based RFID/NFC authentication with temperature sensor input for relay attacks," *Decis. Support Syst.*, vol. 59, pp. 28–36, Mar. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167923613002509

[30] M. Mehrnezhad, F. Hao, and S. F. Shahandashti, "Tap-tap and pay (TTP): Preventing man-in-the-middle attacks in NFC payment using mobile sensors," in *Security Standardisation Research*, L. Chen and S. Matsuo, Eds. Cham, Switzerland: Springer, 2015, pp. 21–39.

[31] H. T. T. Truong, X. Gao, B. Shrestha, N. Saxena, N. Asokan, and P. Nurmi, "Comparing and fusing different sensor modalities for relay attack resistance in zero-interaction authentication," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2014, pp. 163–171, doi: 10.1109/PERCOM.2014.6813957.

[32] B. Shrestha, N. Saxena, H. T. T. Truong, and N. Asokan, "Drone to the rescue: Relay-resilient authentication using ambient multi-sensing," in *Financial Cryptography and Data Security*. Cham, Switzerland: Springer, 2014, pp. 349–364, doi: 10.1007/978-3-662-45472-5_23.

[33] W. Choi, M. Seo, and D. H. Lee, "Sound-proximity: 2-Factor authentication against relay attack on passive keyless entry and start system," *J. Adv. Transp.*, vol. 2018, no. 1, 2018, Art. no. 1935974, doi: 10.1155/2018/1935974.

[34] I. Gurulian, C. Shepherd, E. Frank, K. Markantonakis, R. N. Akram, and K. Mayes, "On the effectiveness of ambient sensing for detecting NFC relay attacks," in *Proc. IEEE Trustcom/BigDataSE/ICESS*, Aug. 2017, pp. 41–49, doi: 10.1109/Trustcom/BigDataSE/ICESS.2017.218.

[35] C. Shepherd, I. Gurulian, E. Frank, K. Markantonakis, R. N. Akram, E. Panaousis, and K. Mayes, "The applicability of ambient sensors as proximity evidence for NFC transactions," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2017, pp. 179–188. [Online]. Available: https://www.ieee-security.org/TC/SPW2017/MoST/proceedings/Shepherd_MoST17.pdf

[36] R. Jin, L. Shi, K. Zeng, A. Pande, and P. Mohapatra, "MagPairing: Pairing smartphones in close proximity using magnetometers," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1306–1320, Jun. 2016, doi: 10.1109/TIFS.2015.2505626.

[37] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun, "Sound-proof: Usable two-factor authentication based on ambient sound," in *Proc. 24th USENIX Secur. Symp.*, 2015, pp. 483–498.

[38] M. Conti and C. Lal, "Context-based co-presence detection techniques: A survey," *Comput. Secur.*, vol. 88, Jan. 2020, Art. no. 101652.

[39] E. Jones, T. Oliphant, and P. Peterson. (2001). *SciPy: Open Source Scientific Tools for Python*. Accessed: Nov. 20, 2015. [Online]. Available: http://www.scipy.org/

[40] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2011.

[41] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[42] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.

[43] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Microsoft, Redmond, WA, USA, Tech. Rep. MSR-TR-98-14, Apr. 1998.

[44] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.* New York, NY, USA: Association for Computing Machinery, Jun. 2009, pp. 41–48, doi: 10.1145/1553374.1553380.

[45] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013, doi: 10.1109/TPAMI.2013.50.

[46] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[47] B. Alshemali and J. Kalita, "Improving the reliability of deep neural networks in NLP: A review," *Knowl.-Based Syst.*, vol. 191, Mar. 2020, Art. no. 105210, doi: 10.1016/j.knosys.2019.105210.

[48] D. Baishya and R. Baruah, "Recent trends in deep learning for natural language processing and scope for Asian languages," in *Proc. Int. Conf. Augmented Intell. Sustain. Syst. (ICAISS)*, Nov. 2022, pp. 408–411, doi: 10.1109/ICAISS55157.2022.10010807.

[49] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3523–3542, Jul. 2022, doi: 10.1109/TPAMI.2021.3059968.

[50] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A comprehensive survey of deep learning for image captioning," *ACM Comput. Surveys*, vol. 51, no. 6, pp. 1–36, Feb. 2019, doi: 10.1145/3295748.

[51] Z. Hu, J. Tang, Z. Wang, K. Zhang, L. Zhang, and Q. Sun, "Deep learning for image-based cancer detection and diagnosis—A survey," *Pattern Recognit.*, vol. 83, pp. 134–149, Nov. 2018.

[52] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[53] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. NIPS*, 2017, pp. 1–4.

[54] M. Czajkowski and M. Kretowski, "Decision tree underfitting in mining of gene expression data. An evolutionary multi-test tree approach," *Expert Syst. Appl.*, vol. 137, pp. 392–404, Dec. 2019.

[55] K. Xia, J. Huang, and H. Wang, "LSTM-CNN architecture for human activity recognition," *IEEE Access*, vol. 8, pp. 56855–56866, 2020.

[56] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *Proc. 6th Int. Conf. Mobile Comput., Appl. Services*, Nov. 2014, pp. 197–205.

[57] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1533–1540.

[58] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM—A tutorial into long short-term memory recurrent neural networks," 2019, *arXiv:1909.09586*.

[59] F. Chollet. (2015). *Keras*. [Online]. Available: https://keras.io

[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[61] Android Open Source Project. (Aug. 2023). *Measuring Biometric Unlock Security*. [Online]. Available: https://source.android.com/docs/security/features/biometric/measure

[62] K. Katevas, H. Haddadi, and L. Tokarchuk, "SensingKit: Evaluating the sensor power consumption in iOS devices," in *Proc. 12th Int. Conf. Intell. Environ. (IE)*, Sep. 2016, pp. 222–225.

KONSTANTINOS MARKANTONAKIS received the B.Sc. degree from Lancaster University, the M.Sc. and Ph.D. degrees from London, and the M.B.A. degree in international management from Royal Holloway, University of London. He is a Professor of information security with Royal Holloway, University of London, where he is also the Director of the Information Security Group Smart Card and IoT Security Centre (SCC). His research interests include smart card security and applications, secure cryptographic protocol design, embedded system security, the IoT, autonomous systems, and trusted execution environments.

JULIA A. MEISTER received the B.Sc. degree in computer science and information security and the M.Sc. degree in data science and analytics from Royal Holloway, University of London. She is currently pursuing the Ph.D. degree with the University of Brighton. Her research interests include machine learning with confidence, conformal prediction, and event detection and classification.

IAKOVOS GURULIAN received the B.Sc. degree (Hons) in computer science from the University of Surrey, in 2011, the M.Sc. degree in information security from University College London, in 2012, and the Ph.D. degree from Royal Holloway, University of London, in 2018. Currently, he is the Chief Technology Officer with TEKA Systems. His main research interests, developed during his time as an Information Security Researcher with Royal Holloway, include smart-device security, network security, and user-centric security.

CARLTON SHEPHERD received the B.Sc. degree in computer science from Newcastle University, and the Ph.D. degree in information security from the Information Security Group, Royal Holloway, University of London. He is currently a Lecturer in computer science with Newcastle University, based within the Secure and Resilient Systems Centre. Previously, he was a Senior Research Fellow with the Information Security Group, Royal Holloway, University of London. His research interests include the security of trusted execution environments (TEEs) and their applications, system-on-chip security, embedded systems, and hardware security.

RAJA NAEEM AKRAM received the B.Sc. degree, the M.Sc. degree in CS, and the M.Sc. and Ph. D. degrees in information security from Royal Holloway, University of London, in 2007 and 2012, respectively. He is a Senior Lecturer in computer science with the University of Aberdeen, U.K. He has held research positions with Edinburgh Napier University; the University of Waikato; and the Information Security Group, Royal Holloway. His expertise spans cybersecurity, focusing on secure software design, threat analysis, and digital infrastructure protection. He is particularly interested in the security of emerging technologies such as the IoT, cloud computing, and blockchain, with a strong emphasis on bridging theoretical research with real-world cybersecurity solutions.

SARAH HANI ABU GHAZALAH received the B.Sc. degree in information systems from King Khalid University, Saudi Arabia, the M.Sc. degree in computer science from the University of Glasgow, and the Ph.D. degree in information security from Royal Holloway, University of London. She is an Assistant Professor with King Khalid University. Her research interests include network security, secure cryptographic protocol design, and RFID security and privacy.

MUMRAIZ KASI received the bachelor's degree from the Government College University, Lahore, Pakistan, the master's degree from the University of Adelaide, Australia, and the Ph.D. degree in computer science from the University of Waikato, New Zealand. He is an Assistant Professor of computer science with the Balochistan University of IT, Engineering and Management Science (BUITEMS), Pakistan. His research interests include wireless sensor networks, complex event processing, and context-aware systems.

DAMIEN SAUVERON received the B.Sc. degree and the M.Sc. and Ph.D. degrees in computer science from the University of Bordeaux, France. He is currently a Professor at the University of Limoges, and he is the Dean of the Faculty of Science and Technology. He was Head of the Computer Science Department, Faculty of Science and Technology, University of Limoges, from 2016 to 2020. Between 2011 and 2019, he has been a member of the CNU 27, the National Council of Universities (for France). He has been chair of IFIP WG 11.2 Pervasive Systems Security between 2014 and 2022, having previously been appointed Vice-Chair of the working group. His research interests are related to smart card applications and security (at hardware and software level), RFID/NFC applications and security, mobile network applications and security (particularly UAV), sensor network applications and security, the Internet of Things (IoT) security, cyber–physical systems security, and security certification processes. He has been involved in more than 100 research events in a range of capacities. In December 2013, the General Assembly of IFIP (International Federation for Information Processing) awarded D. Sauveron the IFIP Silver Core award for his work.

GERHARD HANCKE (Fellow, IEEE) received the B.Eng. and M.Eng. degrees in computer engineering from the University of Pretoria, South Africa, in 2002 and 2003, respectively, and the Ph.D. degree in computer science from the University of Cambridge, U.K., in 2009. He is a Professor with the Department of Computer Science, City University of Hong Kong. Previously, he was a Researcher with the Smart Card and IoT Security Centre and a Teaching Fellow with the Department of Information Security, Royal Holloway, University of London. His research interests include security, reliable communication, and distributed sensing for the Industrial Internet of Things.