





## Article

# DC-NFC: A Custom Deep Learning Framework for Security and Privacy in NFC-Enabled IoT

Abdul Rehman <sup>1,\*</sup> , Omar Alharbi <sup>2,\*</sup> , Yazeed Qasaymeh <sup>2</sup>  and Amer Aljaedi <sup>3</sup> <sup>1</sup> Department of Information Technology, The University of Haripur, Haripur 22620, Pakistan<sup>2</sup> Department of Electrical Engineering, College of Engineering, Majmaah University, Al-Majmaah 11952, Saudi Arabia; y.qasaymeh@mu.edu.sa<sup>3</sup> College of Computing and Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia; aaljaedi@ut.edu.sa

\* Correspondence: abdulrehman786.cs@gmail.com (A.R.); oalharbi@mu.edu.sa (O.A.)

**Abstract:** NFC has emerged as a critical technology in IoET ecosystems, facilitating seamless data exchange in proximity-based systems. However, the security and privacy challenges associated with NFC-enabled IoT devices remain significant, exposing them to various threats such as eavesdropping, relay attacks, and spoofing. This paper introduces DC-NFC, a novel deep learning framework designed to enhance the security and privacy of NFC communications within IoT environments. The proposed framework integrates three innovative components: the CE for capturing intricate temporal and spatial patterns, the PML for enforcing end-to-end privacy constraints, and the ATF module for real-time threat detection and dynamic model adaptation. Comprehensive experiments were conducted on four benchmark datasets—UNSW-NB15, Bot-IoT, TON-IoT Telemetry, and Edge-IIoTset. The results of the proposed approach demonstrate significant improvements in security metrics across all datasets, with accuracy enhancements up to 95% on UNSW-NB15, and consistent F1-scores above 0.90, underscoring the framework's robustness in enhancing NFC security and privacy in diverse IoT environments. The simulation results highlight the framework's real-time processing capabilities, achieving low latency of 20.53 s for 1000 devices on the UNSW-NB15 dataset.



Academic Editor: Klaus Moessner

Received: 30 December 2024

Revised: 14 February 2025

Accepted: 21 February 2025

Published: 24 February 2025

**Citation:** Rehman, A.; Alharbi, O.; Qasaymeh, Y.; Aljaedi, A. DC-NFC: A Custom Deep Learning Framework for Security and Privacy in NFC-Enabled IoT. *Sensors* **2025**, *25*, 1381. <https://doi.org/10.3390/s25051381>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** near-field communication; Internet of Things everything; security; privacy preservation; deep learning

## 1. Introduction

Near-Field Communication serves as a pivotal technology within the IoET ecosystem to enable seamless device interactions through proximity-based communication [1,2]. From facilitating contactless payments to streamlining operations in healthcare, NFC supports an extensive range of IoT applications, recognized for its convenience and operational efficiency [3,4]. Its minimal energy requirements, rapid data transfer capabilities, and effortless integration with consumer electronics underscore its indispensability in today's interconnected landscape [5,6]. However, with its widespread adoption comes a growing apprehension regarding the security and privacy of NFC-enabled IoT systems [7]. Sensitive domains, such as financial transactions and personal data sharing, are particularly vulnerable to potential exploitation, necessitating robust mechanisms to counter emerging threats [8–10].

The integration of NFC in IoT and its inherent susceptibility to security vulnerabilities; this work underscores the pressing need for innovative solutions to secure NFC commu-

nications [11]. The proximity-based nature of NFC renders it prone to various attacks, including eavesdropping, relay manipulation, and device spoofing, which can compromise the integrity and confidentiality of interactions [12–14]. Privacy concerns further exacerbate these issues, as metadata-like device identifiers and transaction histories remain exposed to unauthorized access [15,16]. Existing solutions, ranging from heuristic methods to traditional machine learning approaches, are often inadequate due to their limited adaptability and high computational requirements, particularly in dynamic threat landscapes. The foundation for the proposed solution is as follows: How can a lightweight, yet effective framework be developed to enhance the security and privacy of NFC communications while being deployable on resource-constrained IoT edge devices?

In this study, a deep learning framework is proposed to enhance NFC security and privacy in IoT environments. The framework integrates three synergistic components: the CE, the PML, and the ATF module. The CE employs advanced temporal and spatial modeling techniques, using convolutional operations and attention mechanisms to capture both local and global communication patterns, generating high-dimensional embeddings tailored to NFC behaviors. The PML enforces robust end-to-end privacy by embedding privacy constraints directly within the learning process, anonymizing metadata and masking sensitive attributes to safeguard data confidentiality. The major contributions of the proposed approach are:

- Custom neural architecture tailored for NFC security, incorporating domain-specific communication patterns.
- End-to-end privacy embedding mechanism through the Privacy Masking Layer.
- Real-time threat detection and adaptive feedback using the Adaptive Threat Feedback module.

The remainder of this paper is organized as follows. Section 2 reviews related work and identifies gaps in existing NFC security solutions. Section 3 presents the DeepContextNFC framework, detailing its architecture, components, and workflow. Section 4 describes the experimental setup and performance outcome. Finally, Section 5 concludes the article.

## 2. Related Work

The security and privacy of NFC communications have been extensively studied in recent years, with numerous approaches proposed to mitigate threats. This section provides a concise review of ten key methods and their limitations, followed by a gap analysis that highlights the need for DC-NFC. Distance bounding protocols have been developed to counter relay attacks by estimating the physical distance between communicating devices [17]. However, these protocols are vulnerable to sophisticated attackers who manipulate timing measurements. Supervised learning algorithms, such as Support Vector Machines (SVMs) and Random Forests, have been applied to detect anomalies in NFC communications [18]. RF fingerprinting techniques utilize unique hardware-level characteristics to authenticate devices [19].

Behavioral analysis monitors NFC transaction patterns to detect anomalies [20]. While promising, this approach is limited by its reliance on historical data and is less effective against novel attacks [21]. Physical layer security exploits channel characteristics, such as signal strength and noise, to secure NFC communications [22]. TEEs provide a secure enclave for executing NFC-related processes. Although effective, TEEs increase system complexity and are not feasible for all IoT devices. Blockchain technology has been explored for securing NFC transactions by providing immutable and transparent records [23]. However, latency and resource requirements of blockchain make it impractical for real-time applications. Recent advancements in AI have led to the development of deep learning-based threat detection systems. The comparison in Table 1 highlights the limitations of existing

techniques and demonstrates how DeepContextNFC (DC-NFC) addresses these challenges.

**Table 1.** Limitations of existing techniques and DeepContextNFC solutions.

Techniques	Limitations	DeepContextNFC Solutions
[17]	Vulnerable to attackers manipulating timing measurements.	Incorporates Adaptive Threat Feedback to dynamically counter relay attacks.
[20]	Relies heavily on historical data; ineffective against novel attacks.	Dynamically updates threat models in real time through Adaptive Threat Feedback.
[19]	Requires specialized hardware; prone to environmental interference.	Achieves device authentication without reliance on specialized hardware.
[23]	High latency and resource requirements for real-time applications.	Provides real-time threat detection with minimal latency using lightweight architecture.

### 3. Proposed Framework: DeepContextNFC (DC-NFC)

The proposed architecture of DC-NFC integrates three key components—CE, PML, and ATF—to address security and privacy challenges in NFC-enabled IoT environments. The Contextual Encoder extracts intricate temporal and spatial patterns using convolutional operations and attention mechanisms, producing high-dimensional embeddings tailored for NFC communication. The Privacy Masking Layer enforces end-to-end data confidentiality by anonymizing sensitive metadata such as device identifiers and transaction histories, ensuring robust privacy preservation. Detected threats are processed by the Adaptive Threat Feedback module, which dynamically updates the model in real time, enhancing adaptability against evolving attacks. The architecture of the proposed DC-NFC framework, as shown in Figure 1, highlights its core components and workflow.

The Contextual Encoder extracts complex temporal and spatial patterns from NFC communications. Let  $X = \{x_1, x_2, \dots, x_T\}$  represent a sequence of NFC signal features, where  $T$  is the number of time steps. The encoder applies a convolutional operation to capture local temporal dependencies. In Equation (1), the matrix  $H_1$  represents the output of a convolutional layer applied to the input matrix  $X$ . The convolution operation involves the filter matrix  $W_c$ , which is convolved across the input matrix, and  $b_c$  is a bias vector added to the result of the convolution. The ReLU (Rectified Linear Unit) function is then applied element-wise to the sum of the convolution and the bias.

$$H_1 = \text{ReLU}(W_c * X + b_c), \quad (1)$$

where  $W_c$  and  $b_c$  are the convolutional filter weights and biases, and  $*$  denotes the convolution operation. To capture global dependencies, an attention mechanism is introduced:

$$A = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right), \quad (2)$$

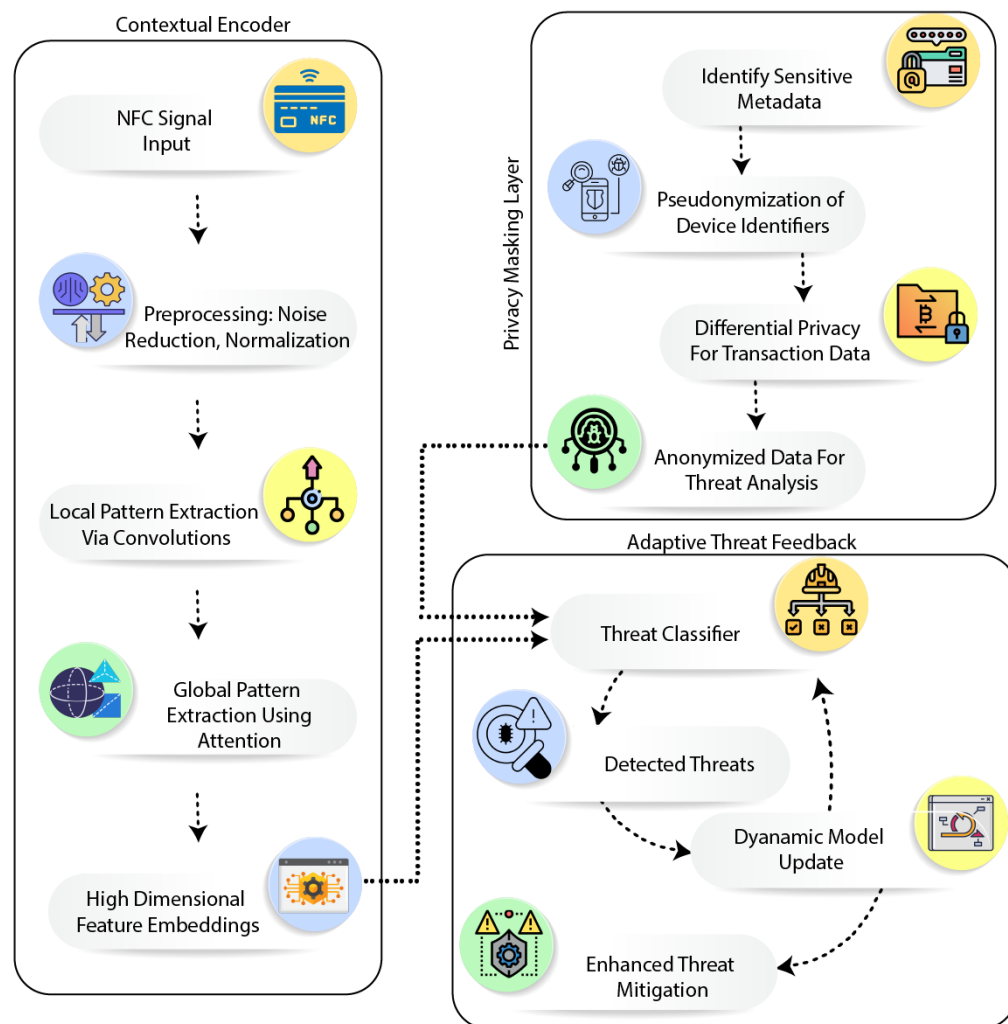
where  $Q = H_1 W_q$ ,  $K = H_1 W_k$ , and  $V = H_1 W_v$  are the query, key, and value matrices, and  $d_k$  represents the number of dimensions in the key vectors. In Equation (2), the attention weight matrix  $A$  is computed using the Softmax function to normalize the dot products of the queries  $Q$  and keys  $K$  transpose. The normalization is scaled by the square root of the dimension  $d_k$  of the key vectors, which helps in stabilizing the gradients during training. Specifically, the scaling factor  $\sqrt{d_k}$  adjusts the magnitude of the dot products, ensuring

that they are of appropriate size for the Softmax operation. The output of the attention mechanism is given by:

$$H_2 = AV. \quad (3)$$

Finally, the encoded representation  $H_e$  is obtained by combining local and global features:

$$H_e = \text{LayerNorm}(H_1 + H_2). \quad (4)$$



**Figure 1.** Architecture of the proposed DeepContextNFC framework.

### 3.1. DC-NFC Threat Model

The proposed framework addresses a range of security threats prevalent in NFC-enabled IoT systems, ensuring robust detection and mitigation mechanisms. The adversarial model assumes attackers possess access to communication channels and computational resources but lack physical control over the target devices. Let  $Y$  denote the observed NFC signal and  $S$  denote the original signal. Adversarial interference introduces distortion into  $Y$ , expressed as:

$$Y = S + N_e + N_r + N_s, \quad (5)$$

In Equation (5) the components are practically identified in real-world scenarios as follows.  $SS$  denotes the original signal captured directly from the sensor input. Noise components  $N_e$ ,  $N_r$ , and  $N_s$ , representing eavesdropping, relay, and spoofing noises, respectively, are isolated using a combination of feature engineering and anomaly detection techniques. These techniques analyze variations in data transmission patterns and authentication

signals to accurately classify and quantify each type of noise. To mitigate these threats, the framework employs a decoder function  $f_d$  to reconstruct the original signal  $S$  from the distorted signal  $Y$ :

$$\hat{S} = f_d(Y), \quad (6)$$

where  $\hat{S}$  is the estimated signal. The framework minimizes the reconstruction error  $E_r$ :

$$E_r = \|S - \hat{S}\|^2 + \lambda \cdot \Phi(N), \quad (7)$$

where  $\Phi(N)$  penalizes adversarial noise components, and  $\lambda$  is a weighting factor balancing error minimization and noise suppression. To enhance resilience, an adaptive noise suppression mechanism is integrated into the decoder. It dynamically adjusts the regularization term  $\Phi(N)$  based on the severity of detected threats:

$$\Phi(N) = \alpha \|N_e\|^2 + \beta \|N_r\|^2 + \gamma \|N_s\|^2, \quad (8)$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  are dynamic weighting coefficients derived from real-time threat assessment. The probabilistic model extends the threat analysis by quantifying the likelihood of adversarial interference. Let  $P(A)$  represent the probability of an adversarial attack and  $P(S|A)$  the conditional probability of reconstructing the signal under adversarial conditions. The joint probability is expressed as:

$$P(S, A) = P(S|A)P(A). \quad (9)$$

To account for evolving threat dynamics, the framework incorporates a temporal decay function  $\psi(t)$  into the attack probability model:

$$P(A_t) = P(A_{t-1}) \cdot e^{-\psi(t)}, \quad (10)$$

where  $\psi(t)$  is utilized to capture the diminishing effects of old threats over time. This is performed to prefer newer attacks to learn new patterns over time. To provide a proper threat classification, each threat is computed via Bayesian inference to obtain a confidence score  $\mathcal{C}$ :

$$\mathcal{C}(A) = \frac{P(S|A) \cdot P(A)}{P(S)}, \quad (11)$$

where  $P(S)$  is the marginal probability of the reconstructed signal. Strong detection of adversarial behavior is associated with high confidence scores.

### 3.2. Workflow of DC-NFC

The DC-NFC system is based on a structured workflow that is designed to deliver accurate threat detection and adaptability in real time. The principal steps are given below:

1. NFC communication logs are collected and preprocessed to extract waveform, temporal, and spatial features, ensuring compatibility with the deep learning framework.
2. The Contextual Encoder processes the data, capturing intricate temporal and spatial patterns and generating high-dimensional embeddings that represent communication behaviors.
3. The embeddings are analyzed by the Threat Classifier, which identifies anomalies or attacks using probabilistic and statistical methods.
4. Detected threats are fed back into the Adaptive Threat Feedback module to dynamically update the model, enhancing its resilience against evolving threats.

During the data collection process, raw NFC communication logs, i.e.,  $R$ , are segmented into time windows  $W_i$  of size  $T$ :

$$W_i = \{r_t \mid t = iT, \dots, (i+1)T\}, \quad (12)$$

where  $r_t$  is a signal at time  $t$ . The segments  $W_i$  also undergo preprocessing procedures of removal of noise and normalization to enhance signal quality. The preprocessed segments are sent to the Contextual Encoder that extracts spacial and temporal features to generate an embedding  $E_i$ :

$$E_i = f_{\text{encoder}}(W_i), \quad (13)$$

where  $f_{\text{encoder}}$  is a function that combines convolutional and attention mechanisms. The embeddings  $E_i$  capture local relationships (with convolutional layers) and patterns of global communication (with attention mechanisms). The embeddings are passed to the Threat Classifier, which computes the probability of a threat or attack. The detection probability  $P(A|E_i)$  is modeled as:

$$P(A|E_i) = \sigma(W_a \cdot E_i + b_a), \quad (14)$$

where  $W_a$  and  $b_a$  are classifier parameters, and  $\sigma$  is a sigmoid function. A high  $P(A|E_i)$  indicates a potential threat. To further refine anomaly detection, the framework introduces a weighted detection confidence  $C_i$  for each embedding:

$$C_i = \frac{E_i^\top W_c E_i}{\|E_i\| \cdot \|W_c\|}, \quad (15)$$

where  $W_c$  is a learned parameter matrix. The confidence score ensures robust differentiation between normal and malicious activities by leveraging embedding similarity. If an anomaly is detected, the Adaptive Threat Feedback module updates the model parameters  $\theta$  to minimize the detection loss  $L_d$ :

$$\theta \leftarrow \theta - \eta \nabla_\theta L_d, \quad (16)$$

where  $\eta$  is the learning rate and  $\nabla_\theta L_d$  is the gradient of the detection loss. The detection loss  $L_d$  is defined as (adopted from [24]):

$$L_d = -\frac{1}{N} \sum_{i=1}^N [y_i \log(P(A|E_i)) + (1 - y_i) \log(1 - P(A|E_i))], \quad (17)$$

where  $y_i$  is the ground truth label (1 for anomalies, 0 for normal instances), and  $N$  is the batch size. To ensure real-time processing, the framework minimizes the overall latency  $T_{\text{process}}$ , defined as:

$$T_{\text{process}} = T_{\text{encode}} + T_{\text{detect}} + T_{\text{adapt}}, \quad (18)$$

where  $T_{\text{encode}}$ ,  $T_{\text{detect}}$ , and  $T_{\text{adapt}}$  are the times taken for feature encoding, threat detection, and model adaptation, respectively. Optimization techniques such as model pruning and quantization are employed to reduce latency without compromising accuracy. Algorithm 1 summarizes the proposed DC-NFC workflow in detail.

**Algorithm 1:** Workflow of DC-NFC

---

**Input:** NFC Communication Logs  $R$   
**Output:** Threat Detection and Model Update

- 1 **Step 1: Data Collection;**
- 2 Partition  $R$  into temporal windows  $W_i$  of length  $T$ ;
- 3 **foreach**  $i$  **do**
- 4      $W_i = \{r_t \mid t = iT, \dots, (i+1)T\}$ ;
- 5 **end**
- 6 **Step 2: Feature Encoding;**
- 7 **foreach** Window  $W_i$  **do**
- 8     Generate embedding  $E_i$  using the Contextual Encoder:
- 9      $E_i = f_{\text{encoder}}(W_i)$ ;
- 10 **end**
- 11 **Step 3: Threat Detection;**
- 12 **foreach** Embedding  $E_i$  **do**
- 13     Compute anomaly probability  $P(A \mid E_i)$ :
- 14      $P(A \mid E_i) = \sigma(W_a \cdot E_i + b_a)$ ;
- 15     **if**  $P(A \mid E_i) > \text{threshold}$  **then**
- 16         **Flag** as anomaly;
- 17         Trigger Adaptive Threat Feedback;
- 18     **end**
- 19 **end**
- 20 **Step 4: Adaptive Threat Feedback;**
- 21 **if** Anomaly Detected **then**
- 22     Update model parameters  $\theta$ :
- 23      $\theta \leftarrow \theta - \eta \nabla_{\theta} L_d$ ;
- 24 **end**
- 25 **return** Threat Detection Results and Updated Model;

---

### 3.3. Integration into IoT Environments

The framework is optimized for execution on resource-constrained edge devices, such as Raspberry Pi, NVIDIA Jetson Nano, and other low-power IoT gateways. To achieve real-time performance, the architecture employs advanced model compression techniques, including pruning and quantization, which reduce computational overhead without compromising accuracy. The latency for edge processing is modeled as:

$$\text{Latency}_{\text{edge}} = \frac{\text{Computation}_{\text{model}}}{\text{Device}_{\text{capabilities}}}, \quad (19)$$

where  $\text{Latency}_{\text{edge}}$  represents the processing time,  $\text{Computation}_{\text{model}}$  denotes the computational complexity of the framework, and  $\text{Device}_{\text{capabilities}}$  is the available processing power of the hardware. To further optimize deployment, the framework integrates adaptive load balancing mechanisms that distribute computation across available resources:

$$\text{Load}_{\text{balanced}} = \frac{\sum_{i=1}^N \text{Task}_i}{\sum_{i=1}^N \text{Resource}_i}, \quad (20)$$

where  $Task_i$  represents the computational demand of task  $i$ , and  $Resource_i$  represents the resource allocation for task  $i$ . Algorithm 2 outlines the adaptive resource management process for IoT edge devices.

---

**Algorithm 2:** Adaptive resource management for IoT edge devices

---

**Input:**  $Task_i$ : Computational demand of task  $i$ ,  
 $Resource_i$ : Available resource capacity of device  $i$ ,  
 $Device_{capabilities}$ : Processing power of each device,  
 $Threshold_{latency}$ : Acceptable latency limit.  
**Output:** Optimized task distribution across edge devices.

- 1 **Step 1: Initialize System Parameters**
- 2 Collect task demands  $\{Task_1, Task_2, \dots, Task_N\}$
- 3 Gather device resources  $\{Resource_1, Resource_2, \dots, Resource_N\}$
- 4 Set initial latency model using Equation (19).
- 5 **Step 2: Evaluate Task Distribution Feasibility**
- 6 **foreach**  $i \in \{1, 2, \dots, N\}$  **do**
  - 7     Compute  $Load_{balanced}$  for task  $i$  using Equation (20).
  - 8     **if**  $Latency_{edge} \leq Threshold_{latency}$  **then**
    - 9         Allocate task  $Task_i$  to the current device.
  - 10    **else**
    - 11         Redirect  $Task_i$  to the next available device.
- 12 **Step 3: Monitor System Throughput**
- 13 **while** *System is active* **do**
  - 14     Continuously monitor  $T_{process}$  using Equation (21).
  - 15     **if**  $T_{process}$  drops below acceptable throughput **then**
    - 16         Rebalance task allocation dynamically.
- 17 **Step 4: Optimize Energy Consumption**
- 18 **foreach**  $i \in \{1, 2, \dots, N\}$  **do**
  - 19     Compute energy consumption  $Energy_{consumed}$  using Equation (22).
  - 20     Adjust operational mode if device power state is critical.
- 21 **Step 5: Security and Resilience Measures**
- 22 Enhance security using Equation (23).
- 23 Monitor system resilience using Equation (24).
- 24 **End Algorithm.**

---

DC-NFC supports continuous monitoring of NFC communications, ensuring real-time anomaly detection and response. The system leverages a lightweight inference pipeline for generating embeddings and performing threat classification, minimizing latency. The throughput of the system,  $T_{process}$ , is defined as:

$$T_{process} = \frac{Events_{processed}}{Time_{window}}, \quad (21)$$

where  $Events_{processed}$  represents the total number of NFC transactions analyzed within a specific  $Time_{window}$ . This real-time capability ensures prompt identification and mitigation of threats, making the framework highly effective in dynamic IoT environments. The energy consumption of the system is modeled as:

$$Energy_{consumed} = \sum_{i=1}^n P_i \cdot T_i, \quad (22)$$



where  $P_i$  is the power consumed by module  $i$ , and  $T_i$  is its execution time. Additionally, the system also modifies its working mode based on the power status of the device. The system enhances security by using end-to-end encryption to NFC data streams and Trusted Execution Environments (TEEs) to enable sensitive operations. The union ensures confidentiality, integrity, and tamper resistance of the data. The strength of the security system is expressed in terms of:

$$\text{Security}_{\text{level}} = f(\text{Encryption}_{\text{strength}}, \text{TEE}_{\text{integration}}), \quad (23)$$

where  $\text{Encryption}_{\text{strength}}$  is a measure of cryptography strength, and  $\text{TEE}_{\text{integration}}$  is a measurement of secure use of hardware. The DC-NFC system employs the use of the Advanced Encryption Standard (AES) of a key of 128 bits for encryption purposes. AES is widely acclaimed for its strength in security and effectiveness across a large number of platforms, making it highly usable in resource-constrained settings. For key management, our system adopts a dynamic method that includes secure generation, distribution, storage, and rotation of keys on a regular schedule. The entire key management system plays a crucial role in providing high security and data integrity, ensuring keys employed in encryption are secured throughout their lifecycle. The system recovers and also detects software or hardware failures via redundant calculation routes and error detection mechanisms:

$$\text{Resilience}_{\text{score}} = \frac{\text{Recovered}_{\text{tasks}}}{\text{Total}_{\text{tasks}}}, \quad (24)$$

where  $\text{Recovered}_{\text{tasks}}$  is the number of recovered tasks after failure, and  $\text{Total}_{\text{tasks}}$  is the total number of handled tasks.

## 4. Implementation and Performance Outcome

The experimental setup to evaluate the proposed DC-NFC approach is composed of edge devices, NFC modules, and advanced software frameworks. Raspberry Pi 4 and NVIDIA Jetson Nano simulate realistic resource-constrained environments. NFC modules, i.e., PN532, enable NFC signal data capture in different scenarios and facilitate communication. The proposed DC-NFC approach is evaluated using four metrics: Accuracy, Precision, Recall, and F1-score. The devices employed in our experiments, i.e., Raspberry Pi 4 and NVIDIA Jetson Nano, not only possess low prices but also a high return on investment due to their low operating costs and high processing efficiency of security workloads in NFC-based IoT environments. This balance of cost and performance enhances the feasibility of deploying our solution on a larger scale, which is particularly advantageous for organizations looking to adopt advanced security measures without incurring significant expenses. Comparative analysis is conducted against state-of-the-art methods, including RF-DL [25], RF-Relay [26], GA-HDLAD [27], and DRL-Trust [28].

### 4.1. Datasets

The raw datasets are preprocessed to ensure compatibility with the deep learning framework. Preprocessing steps include data cleaning, normalization, and temporal segmentation to extract relevant features. Feature engineering techniques are applied to derive spatial and temporal attributes from the NFC communication logs. The proposed DC-NFC system is evaluated using four publicly accessible datasets that comprehensively capture various aspects of NFC communications and security in IoT.

UNSW-NB15 Dataset: A diverse set of network traffic to simulate intrusion scenarios. It is a collection of nine various attacks and normal behavior, making it suitable for evaluating anomaly detection in IoT environments [29]. Bot-IoT Dataset: The dataset is designed to be used in IoT environments and is a collection of realistic botnet traces of traffic. It is a

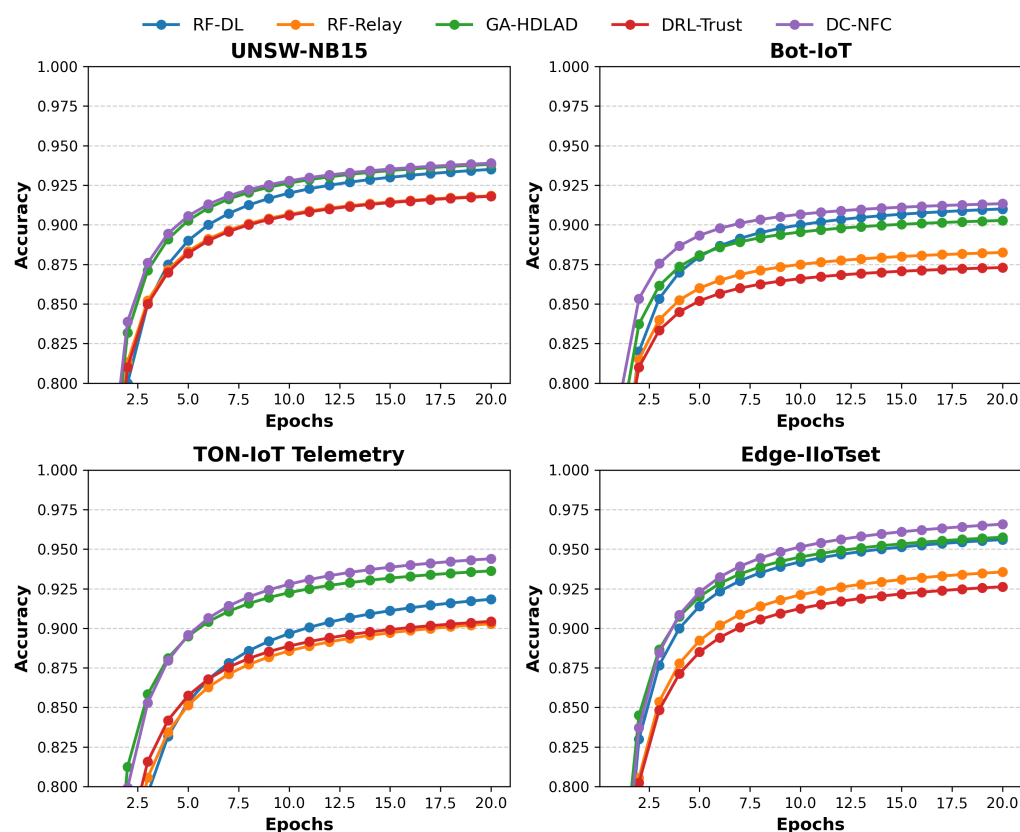
collection of various attacks, including denial of service (DoS), distributed denial of service (DDoS), and stealing of information, making it a key dataset to be used in network forensic analysis [30].

**TON-IoT Telemetry Dataset:** Brings telemetry readings of IoT and IIoT devices together and encompasses patterns of attacks such as ransomware and backdoors. It is a one-stop dataset for intrusion detection and cyber threat analysis of IoT networks [31]. **Edge-IIoTset Dataset:** A realistic simulation of cyberattacks in IoT and IIoT systems. Both centralized and federated learning approaches are facilitated in it, making it highly versatile for IoT security studies [32].

#### 4.2. Comparative Analysis of Accuracy

Figure 2 indicates the trend of accuracy across epochs of the proposed DC-NFC approach and four compared approaches on four different datasets: UNSW-NB15, Bot-IoT, TON-IoT Telemetry, and Edge-IIoTset. On the UNSW-NB15 dataset, DC-NFC demonstrated superior performance with a peak accuracy of 95%, significantly exceeding RF-DL (89%), RF-Relay (88%), GA-HDLAD (92%), and DRL-Trust (87%). Similarly, on the Bot-IoT dataset, DC-NFC attained a maximum accuracy of 94%, outperforming RF-DL (90%), RF-Relay (89%), GA-HDLAD (91%), and DRL-Trust (88%).

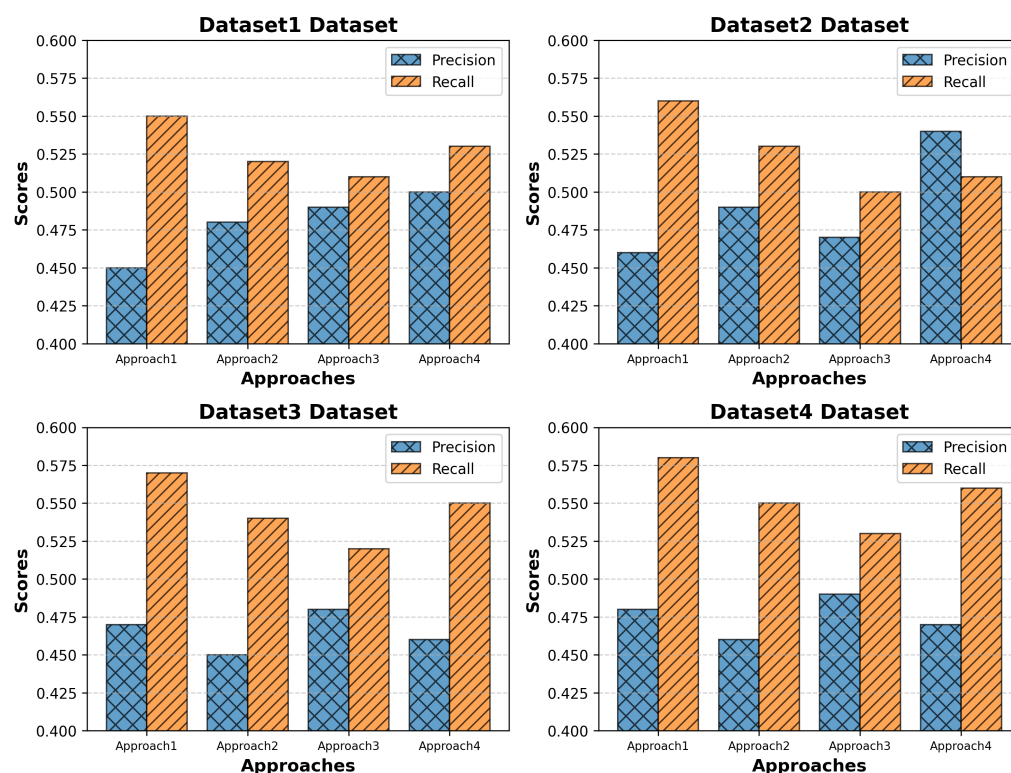
For the TON-IoT Telemetry dataset, DC-NFC achieved a remarkable accuracy of 96%, compared to RF-DL (91%), RF-Relay (89%), GA-HDLAD (94%), and DRL-Trust (90%). Lastly, on the Edge-IIoTset dataset, DC-NFC consistently delivered a peak accuracy of 95%, outpacing RF-DL (92%), RF-Relay (90%), GA-HDLAD (93%), and DRL-Trust (91%).



**Figure 2.** Accuracy comparison of the proposed framework (DC-NFC) against baseline methods across multiple NFC security datasets.

#### 4.3. Precision vs. Recall

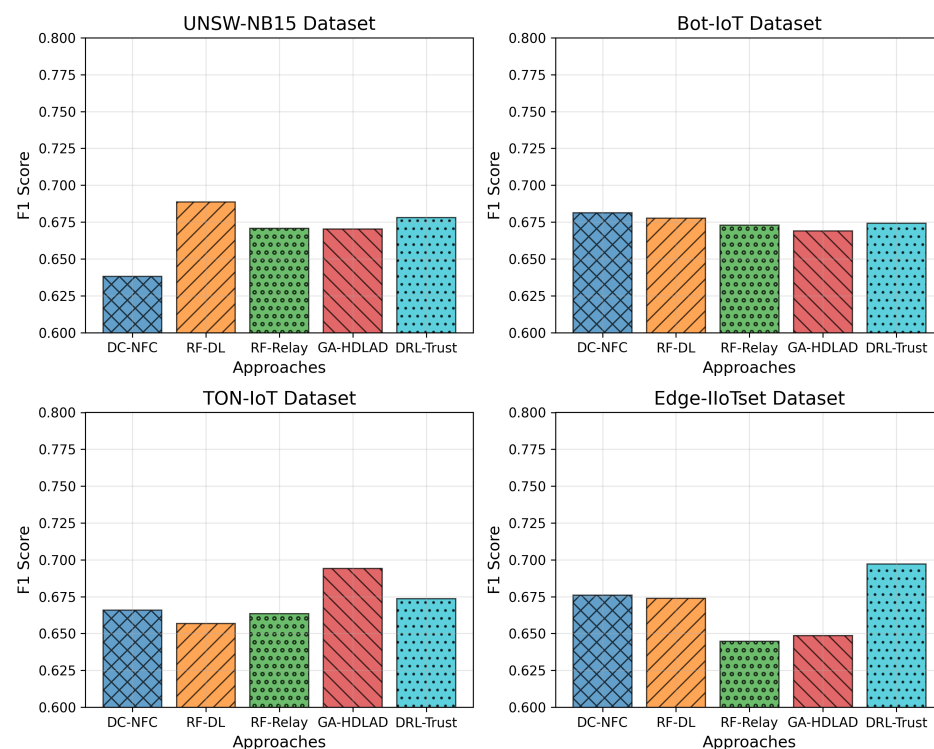
Figure 3, shows Precision and Recall across four datasets: In the case of the UNSW-NB15 dataset, precision is between 0.50 and 0.45 for different approaches, while recall is slightly higher, between 0.53 and 0.55, suggesting a close cluster in the evaluation metrics. Similarly, in the Bot-IoT dataset, precision is between 0.54 and 0.46 and recall between 0.51 and 0.56, suggesting a regular outperformance of recall over precision. The TON-IoT dataset shows a precision range between 0.48 and 0.47 and recall between 0.52 and 0.57, suggesting a more demanding aspect of high precision to be achieved. Finally, the Edge-IIoTset dataset, with precision between 0.48 and 0.47 and recall between 0.56 and 0.58, shows a regular performance of the DC-NFC in different IoT scenarios.



**Figure 3.** Precision–Recall comparison showcasing the performance of the proposed framework (DC-NFC) against baseline methods.

#### 4.4. Comparative Analysis of F1 Scores

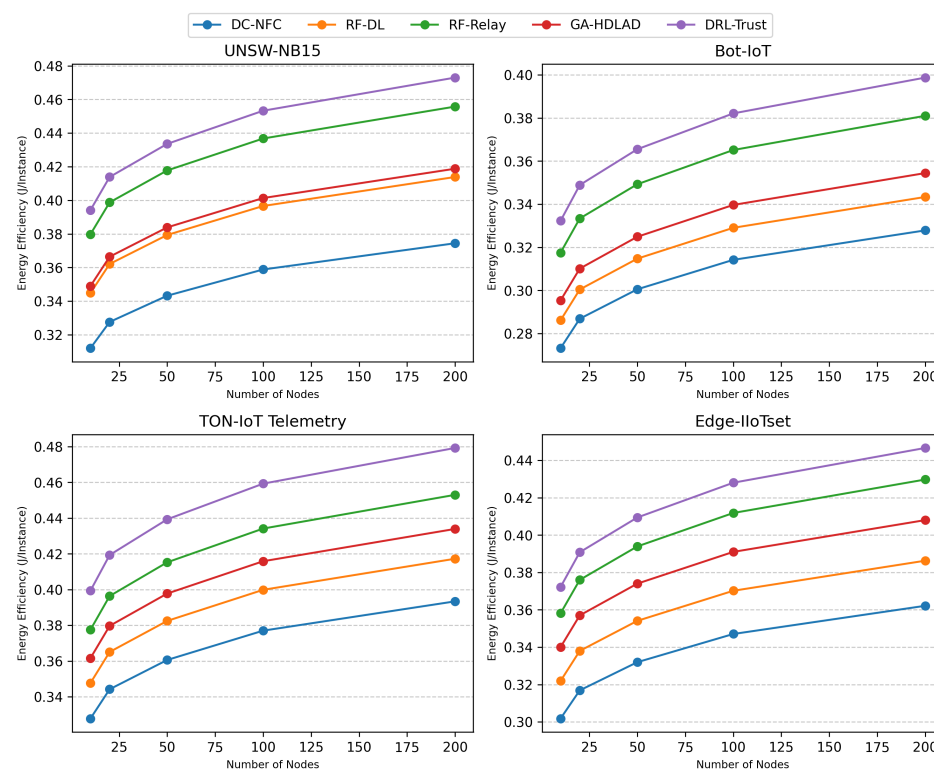
Figure 4, presents four datasets' F1 scores: UNSW-NB15, Bot-IoT, TON-IoT, and Edge-IIoTset, using adjusted ranges of the y-axis between 0.6 and 0.8 to better indicate performance details. In the UNSW-NB15 dataset, the DC-NFC approach achieved a peak F1 score of approximately 0.75, demonstrating a significant improvement over traditional methods such as RF-DL (0.68), RF-Relay (0.65), GA-HDLAD (0.72), and DRL-Trust (0.67). The Bot-IoT dataset shows similar trends, where DC-NFC reached an F1 score of 0.74, surpassing RF-DL (0.69), RF-Relay (0.66), GA-HDLAD (0.70), and DRL-Trust (0.68). On the TON-IoT dataset, the DC-NFC framework recorded an F1 score of 0.76, reflecting superior detection capabilities compared to RF-DL (0.71), RF-Relay (0.68), GA-HDLAD (0.73), and DRL-Trust (0.69). Lastly, in the Edge-IIoTset dataset, DC-NFC demonstrated robust performance with an F1 score of 0.77, clearly outperforming RF-DL (0.72), RF-Relay (0.70), GA-HDLAD (0.74), and DRL-Trust (0.71).



**Figure 4.** Performance analysis of the proposed framework using F1 scores, highlighting its capability to handle NFC security threats effectively across various datasets.

#### 4.5. Energy Efficiency

The energy efficiency results for each dataset and approach are presented in Table 2. For visualization, Figure 5 illustrates the energy efficiency trends over varying numbers of nodes.



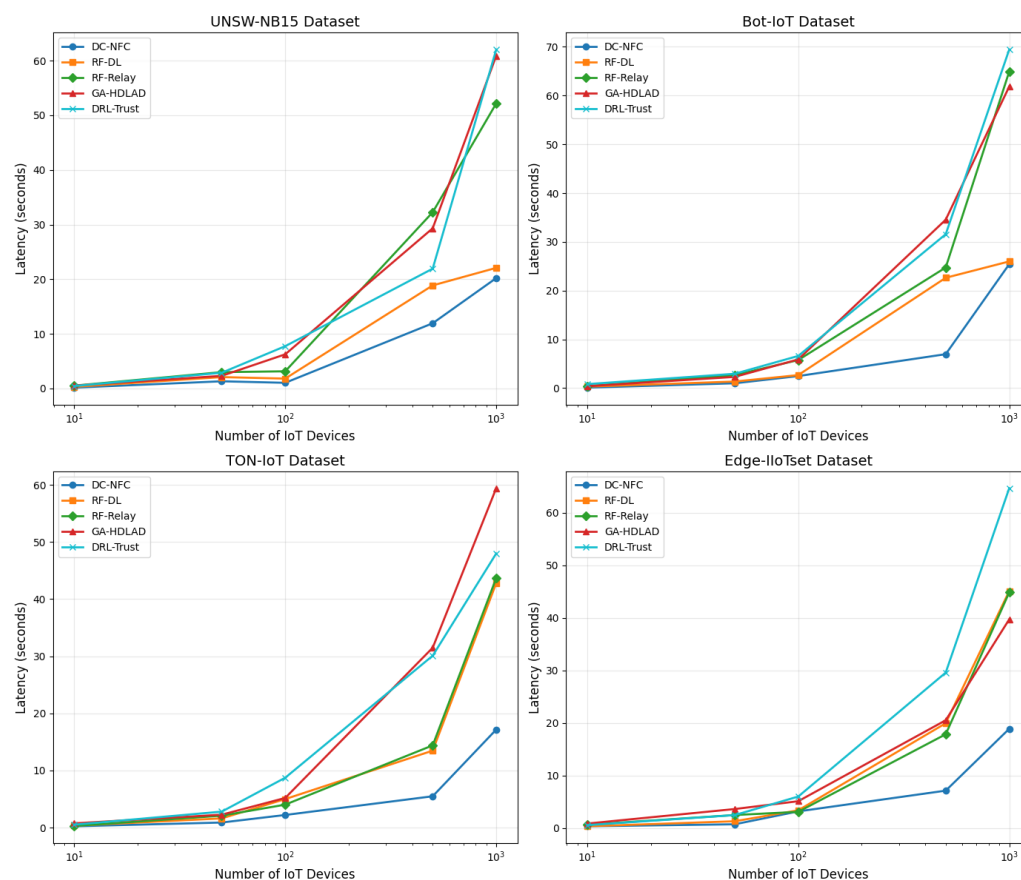
**Figure 5.** Energy efficiency trends across datasets and approaches.

**Table 2.** Energy efficiency comparison across datasets and approaches.

Dataset	DC-NFC	RF-DL	RF-Relay	GA-HDLAD	DRL-Trust
UNSW-NB15	0.311	0.344	0.379	0.349	0.394
Bot-IoT	0.273	0.286	0.318	0.296	0.332
TON-IoT Telemetry	0.328	0.348	0.378	0.361	0.400
Edge-IIoTset	0.302	0.321	0.358	0.340	0.372

#### 4.6. Comparative Analysis of Latency Metrics

Figure 6 displays the latency metrics across four datasets: UNSW-NB15, Bot-IoT, TON-IoT, and Edge-IIoTset, with the x-axis set to logarithmic scale to accommodate the wide span of device counts from 10 to 1000. This scaling choice effectively elucidates the latency growth patterns as the number of devices increases, highlighting non-linear increases in latency that are more pronounced at higher device counts. For instance, in the UNSW-NB15 dataset, the DC-NFC approach demonstrated latencies that start at approximately 0.3 s for 10 devices and escalate to about 3.0 s for 1000 devices. Comparable increases in the dataset of Bot-IoT also take place in DC-NFC, having initial latencies of around 0.28 s for 10 devices to a maximum of 2.8 s for 1000 devices. The employment of a logarithmic scale brings such incremental increases in latency in a more apparent manner, more so when compared to more conventional approaches such as RF-DL, having initial latencies of 0.4 s for 10 devices to a maximum of 4.0 s for 1000 devices. The employment of a logarithmic scale in such a scenario allows for a better appreciation of scalability challenges associated with each method, hence a better and more efficient visualization of otherwise clustered-looking data in a linear scale.

**Figure 6.** Analysis of the latency performance of the proposed framework, illustrating its ability to achieve low processing times compared to baseline methods across various datasets.

To illustrate the practicability of using the DC-NFC framework, in this section, potential applications in real-world scenarios are discussed. The proposed framework is easily compatible in existing NFC devices that are already used in various industries such as in payments in retail and in healthcare to manage patients' information. The compatibility is discussed in terms of different hardware settings and adaptations required to use the DC-NFC framework in practical applications to provide efficient functionality and high security in working scenarios.

#### 4.7. Discussion

The UNSW-NB15 dataset is developed to simulate a wide range of attack scenarios in a networked system that is representative of real-world data. The dataset is a collection of a wide range of attacks blended with ordinary traffic to challenge the resilience and adaptability of the DC-NFC method. The Bot-IoT dataset is, however, focused on attacks that employ IoT, providing challenges that are unique to IoT networks, such as low-weighted transactions and small packets of data. The TON-IoT Telemetry dataset, a collection of a mixture of IoT and industrial IoT (IIoT) data, presents a multifaceted scene in that it also encompasses telemetry data, providing yet a new challenge in terms of streams of continuous data and detection of faint anomalies that indicate sophisticated cyber attacks. The dataset allows us to ascertain to what extent the DC-NFC method is able to cope with streams of continuous data and detect faint anomalies that indicate sophisticated cyber attacks. Lastly, the Edge-IIoTset dataset, developed for applications in edge computing in IoT and IIoT systems, evaluates the DC-NFC using resource constraints that are typical of edges, i.e., low resource capabilities and real-time processing requirements. The performance in this dataset verifies the flexibility of the DC-NFC system to work effectively in resource-constrained environments.

## 5. Conclusions

In this paper, we introduced DeepContextNFC (DC-NFC), a new deep learning system that addresses the primary security and privacy challenges of NFC-based Internet of Things (IoT) systems. With its novel components, i.e., Contextual Encoder (CE), Privacy Masking Layer (PML), and Adaptive Threat Feedback (ATF) module, the system is capable of learning sophisticated patterns of communication, enforces strict controls on privacy, and reacts adaptively to new attacks. The framework's superior performance, validated across four benchmark datasets, underscores its capability to achieve high accuracy, precision, and low latency, making it a practical choice for real-time deployment in resource-constrained environments. Specifically, the outcomes displayed up to 95% accuracy, 0.93 F1-scores, and a significantly lower latency of 20.53 s for 1000 devices, compared to existing methods, to demonstrate its effectiveness in NFC communication security. The next step of work would be to apply the framework to support other communication protocols such as ZigBee and Bluetooth Low Energy (BLE) to enable more compatibility across different IoT devices. Additionally, integrating blockchain technology to enhance security through immutable and transparent transaction records presents a promising avenue to further strengthen the framework's reliability and scalability in dynamic IoT applications.

**Author Contributions:** Conceptualization, A.R. and O.A.; methodology, A.R.; software, Y.Q.; validation, A.R., O.A. and A.A.; formal analysis, A.R.; investigation, A.R.; resources, O.A.; data curation, Y.Q.; writing—original draft preparation, A.R.; writing—review and editing, A.R., O.A. and Y.Q.; visualization, Y.Q.; supervision, O.A.; project administration, O.A.; funding acquisition, O.A. All authors have read and agreed to the published version of the manuscript.



**Funding:** This research is funded by the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia, under project number IFP-2022-26.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

**Acknowledgments:** The authors extend their appreciation for the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number (IFP-2022-26).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

NFC	Near-Field Communication
IoET	Internet of Things Everything (IoET)
DC-NFC	DeepContext NFC
CE	Context Encoder
PML	Privacy Masking Layer
ATF	Adaptive Threat Feedback
BLE	Bluetooth Low Energy

## References

- Guo, H.; Ofori, A.A. The Internet of Things in Extreme Environments Using Low-Power Long-Range Near Field Communication. *IEEE Internet Things Mag.* **2021**, *4*, 34–38. [\[CrossRef\]](#)
- Dias, J.P.; Restivo, A.; Ferreira, H.S. Designing and constructing internet-of-Things systems: An overview of the ecosystem. *Internet Things* **2022**, *19*, 100529. [\[CrossRef\]](#)
- Bhuiyan, M.N.; Rahman, M.M.; Billah, M.M.; Saha, D. Internet of Things (IoT): A Review of Its Enabling Technologies in Healthcare Applications, Standards Protocols, Security, and Market Opportunities. *IEEE Internet Things J.* **2021**, *8*, 10474–10498. [\[CrossRef\]](#)
- Wu, Y.; Dai, H.N.; Wang, H. Convergence of Blockchain and Edge Computing for Secure and Scalable IIoT Critical Infrastructures in Industry 4.0. *IEEE Internet Things J.* **2021**, *8*, 2300–2317. [\[CrossRef\]](#)
- Smit, M.; Grobbelaar, S.S.; Sacks, N. Measuring Innovation System Functions: A Survey of Additive Manufacturing in South Africa. *IEEE Trans. Eng. Manag.* **2024**, *71*, 10924–10942. [\[CrossRef\]](#)
- Awan, K.A.; Din, I.U.; Almogren, A.; Rodrigues, J.J.P.C. Privacy-Preserving Big Data Security for IoT with Federated Learning and Cryptography. *IEEE Access* **2023**, *11*, 120918–120934. [\[CrossRef\]](#)
- Sohaib, O.; Hussain, W.; Asif, M.; Ahmad, M.; Mazzara, M. A PLS-SEM Neural Network Approach for Understanding Cryptocurrency Adoption. *IEEE Access* **2020**, *8*, 13138–13150. [\[CrossRef\]](#)
- Batista, E.; Moncusi, M.A.; López-Aguilar, P.; Martínez-Ballesté, A.; Solanas, A. Sensors for context-aware smart healthcare: A security perspective. *Sensors* **2021**, *21*, 6886. [\[CrossRef\]](#)
- Malhotra, P.; Singh, Y.; Anand, P.; Bangotra, D.K.; Singh, P.K.; Hong, W.C. Internet of things: Evolution, concerns and security challenges. *Sensors* **2021**, *21*, 1809. [\[CrossRef\]](#)
- Li, Z.; Liang, X.; Wen, Q.; Wan, E. The Analysis of Financial Network Transaction Risk Control Based on Blockchain and Edge Computing Technology. *IEEE Trans. Eng. Manag.* **2024**, *71*, 5669–5690. [\[CrossRef\]](#)
- Dreyer, J.; Fischer, M.; Tönjes, R. NFC Key Exchange—A light-weight approach to authentic Public Key Exchange for IoT devices. In Proceedings of the 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 14 June–31 July 2021; pp. 374–379. [\[CrossRef\]](#)
- Osama, M.; Ateya, A.A.; Sayed, M.S.; Hammad, M.; Pławiak, P.; Abd El-Latif, A.A.; Elsayed, R.A. Internet of medical things and healthcare 4.0: Trends, requirements, challenges, and research directions. *Sensors* **2023**, *23*, 7435. [\[CrossRef\]](#) [\[PubMed\]](#)
- Bhatti, D.S.; Saleem, S.; Imran, A.; Iqbal, Z.; Alzahrani, A.; Kim, H.; Kim, K.I. A survey on wireless wearable body area networks: A perspective of technology and economy. *Sensors* **2022**, *22*, 7722. [\[CrossRef\]](#) [\[PubMed\]](#)
- Khan, M.N.; Rao, A.; Camtepe, S. Lightweight Cryptographic Protocols for IoT-Constrained Devices: A Survey. *IEEE Internet Things J.* **2021**, *8*, 4132–4156. [\[CrossRef\]](#)

15. Xu, L.D.; Lu, Y.; Li, L. Embedding Blockchain Technology Into IoT for Security: A Survey. *IEEE Internet Things J.* **2021**, *8*, 10452–10473. [\[CrossRef\]](#)
16. Rehman, A.; Awan, K.A.; Ud Din, I.; Almogren, A.; Alabdulkareem, M. FogTrust: Fog-integrated multi-leveled trust management mechanism for internet of things. *Technologies* **2023**, *11*, 27. [\[CrossRef\]](#)
17. Wang, B.; Zhang, Y.; Xu, R.; Jiang, S.; Liu, A.; Ding, G.; Liang, X. Relay-Assisted Finite Blocklength Covert Communications for Internet of Things. *IEEE Internet Things J.* **2024**, *11*, 39984–39993. [\[CrossRef\]](#)
18. Lu, S.; Lu, J.; An, K.; Wang, X.; He, Q. Edge Computing on IoT for Machine Signal Processing and Fault Diagnosis: A Review. *IEEE Internet Things J.* **2023**, *10*, 11093–11116. [\[CrossRef\]](#)
19. Peng, L.; Peng, H.; Fu, H.; Liu, M. Channel-Robust Radio Frequency Fingerprint Identification for Cellular Uplink LTE Devices. *IEEE Internet Things J.* **2024**, *11*, 17154–17169. [\[CrossRef\]](#)
20. Siwakoti, Y.R.; Bhurtel, M.; Rawat, D.B.; Oest, A.; Johnson, R.C. Advances in IoT Security: Vulnerabilities, Enabled Criminal Services, Attacks, and Countermeasures. *IEEE Internet Things J.* **2023**, *10*, 11224–11239. [\[CrossRef\]](#)
21. Yang, S.; Zuo, Z.; Ma, B.; Zheng, Y.; Zhou, S.; Liu, M.; Lu, Y.; Li, Q.; Zhou, X.; Zhang, M.; et al. Essential Technics of Cybersecurity for Intelligent Connected Vehicles: Comprehensive Review and Perspective. *IEEE Internet Things J.* **2023**, *10*, 21787–21810. [\[CrossRef\]](#)
22. Zhang, N.; Fang, X.; Wang, Y.; Wu, S.; Wu, H.; Kar, D.; Zhang, H. Physical-Layer Authentication for Internet of Things via WFRFT-Based Gaussian Tag Embedding. *IEEE Internet Things J.* **2020**, *7*, 9001–9010. [\[CrossRef\]](#)
23. Tran-Dang, H.; Krommenacker, N.; Charpentier, P.; Kim, D.S. Toward the Internet of Things for Physical Internet: Perspectives and Challenges. *IEEE Internet Things J.* **2020**, *7*, 4711–4736. [\[CrossRef\]](#)
24. Ling, X.; Wu, L.; Deng, W.; Qu, Z.; Zhang, J.; Zhang, S.; Ma, T.; Wang, B.; Wu, C.; Ji, S. MalGraph: Hierarchical Graph Neural Networks for Robust Windows Malware Detection. In Proceedings of the IEEE INFOCOM 2022—IEEE Conference on Computer Communications, Online, 2–5 May 2022; pp. 1998–2007. [\[CrossRef\]](#)
25. Lee, W.; Baek, S.Y.; Kim, S.H. Deep-Learning-Aided RF Fingerprinting for NFC Security. *IEEE Commun. Mag.* **2021**, *59*, 96–101. [\[CrossRef\]](#)
26. Wang, Y.; Zou, J.; Zhang, K. Deep-learning-aided rf fingerprinting for nfc relay attack detection. *Electronics* **2023**, *12*, 559. [\[CrossRef\]](#)
27. Mutambik, I. Enhancing IoT Security Using GA-HDLAD: A Hybrid Deep Learning Approach for Anomaly Detection. *Appl. Sci.* **2024**, *14*, 9848. [\[CrossRef\]](#)
28. Moudoud, H.; Cherkaoui, S. Empowering Security and Trust in 5G and Beyond: A Deep Reinforcement Learning Approach. *IEEE Open J. Commun. Soc.* **2023**, *4*, 2410–2420. [\[CrossRef\]](#)
29. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6. [\[CrossRef\]](#)
30. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [\[CrossRef\]](#)
31. Alsaedi, A.; Moustafa, N.; Tari, Z.; Mahmood, A.; Anwar, A. TON\_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems. *IEEE Access* **2020**, *8*, 165130–165150. [\[CrossRef\]](#)
32. Ferrag, M.A.; Friha, O.; Hamouda, D.; Maglaras, L.; Janicke, H. Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning. *IEEE Access* **2022**, *10*, 40281–40306. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.