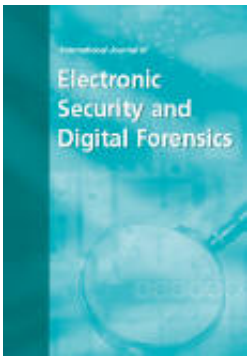
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Electronic Security and Digital Forensics

ISSN online: 1751-9128 - ISSN print: 1751-911X
https://www.inderscience.com/ijesdf

Efficient digital forensics in the IoT environment: a hybrid framework using deep-federated learning

Waad Almadud, Asma Abdulghani Al-Shargabi

DOI: 10.1504/IJESDF.2027.10073649

Article History:

Received:	21 November 2024
Last revised:	22 February 2025
Accepted:	24 February 2025
Published online:	07 January 2026

Copyright © 2026 Inderscience Enterprises Ltd.

Efficient digital forensics in the IoT environment: a hybrid framework using deep-federated learning

Waad Almadud and
Asma Abdulghani Al-Shargabi*

Department of Information Technology,
College of Computer,
Qassim University,
Buraydah, Saudi Arabia
Email: 441212469@qu.edu.sa
Email: as.alshargabi@qu.edu.sa
*Corresponding author

Abstract: In an era of interconnected devices, robust cybersecurity is essential. This research presents a deep learning-based forensics framework for investigating and identifying cyber-attacks in IoT ecosystems. At its core, a hybrid CNN-LSTM model, enhanced by particle swarm optimisation (PSO), dynamically optimises parameters for peak performance. Integrating federated learning (FL), the framework ensures effective generalisation across diverse IoT datasets while preserving data privacy. This lightweight yet highly accurate solution outperforms existing models in accuracy and efficiency. The proposed framework achieves 97.66% accuracy and improves time efficiency by 76.82%, detecting various cyber-attacks across IoT applications such as vehicle networks, smart homes, and smart cities. This advancement strengthens IoT security and provides an efficient method for tracing malicious activities.

Keywords: digital forensics; internet of things; IoT; attack detection; deep learning; federated learning; particle swarm optimisation; PSO; optimisation; efficient algorithm.

Reference to this paper should be made as follows: Almadud, W. and Al-Shargabi, A.A. (2026) 'Efficient digital forensics in the IoT environment: a hybrid framework using deep-federated learning', *Int. J. Electronic Security and Digital Forensics*, Vol. 18, No. 7, pp.1–33.

Biographical notes: Waad Almadud received her BSc in Computer Science from Qassim University, Saudi Arabia. She is currently pursuing her MSc in Cybersecurity at the College of Computer, Qassim University. Her research interests include cybersecurity, digital forensics, IoT security, and deep learning. She has contributed to research projects in artificial intelligence and smart system security.

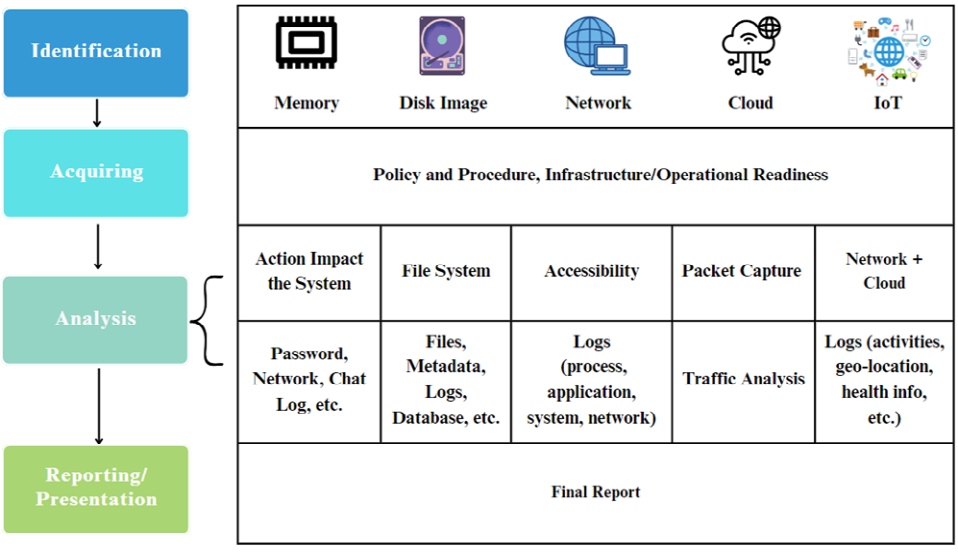
Asma Abdulghani Al-Shargabi is an Assistant Professor of Computer Science at Qassim University, Saudi Arabia. She earned her BSc (1999) and MSc (2006) from the University of Science and Technology, Yemen, and PhD (2015) from De Montfort University, UK. Previously, she served at UST in Sana'a (2015–2019) and held leadership roles in quality assurance. Her research spans ubiquitous computing, IoT, AI, cybersecurity, data science, and Arabic language processing. She has published widely in high-impact journals and conferences and actively contributes to academic programs and curriculum development.

Copyright © The Author(s) 2025. Published by Inderscience Publishers Ltd. This is an Open Access Article distributed under the CC BY license. (http://creativecommons.org/licenses/by/4.0/)

1 Introduction

Digital forensics has become an indispensable field in the modern era of technology. From cybercrimes to data breaches, digital forensics plays a crucial role in uncovering digital trails, reconstructing events, and attributing actions to individuals or entities (Raval, 2020). Digital forensics involves several distinct types of activities. As shown in Figure 1 the first is identification, which involves recognising and locating potential sources of digital evidence. Once identified, the next step is acquisition, where the evidence is collected and preserved in a forensically sound manner to prevent tampering or alteration. Following acquisition, the evidence undergoes analysis, where forensic experts examine and interpret the data to extract relevant information and uncover insights. Finally, the results of the analysis are documented in a comprehensive report or presented in a clear and concise manner to stakeholders, such as law enforcement, legal professionals, or other relevant parties. These types of digital forensics activities work together to ensure a thorough and systematic approach to investigating and presenting digital evidence in legal and investigative proceedings (Salih and Dabagh, 2023).

Figure 1 Phases of digital forensics (see online version for colours)



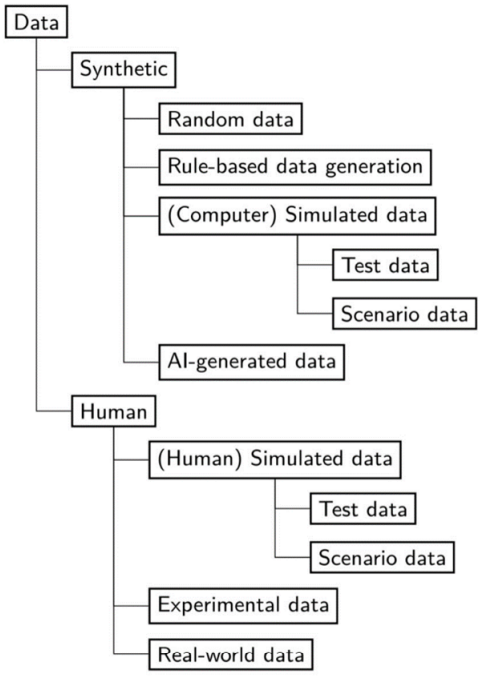
Over the past few years, the digital landscape has witnessed significant changes, giving rise to new avenues of investigation and presenting novel obstacles. One notable development is the growing reliance on cloud-based services for data storage and communication. As individuals and organisations increasingly entrust their information to cloud platforms, digital forensic investigators must adapt their methodologies to effectively collect and analyse evidence from these environments. Innovative techniques and tools have emerged to navigate the complexities of cloud-based investigations, ensuring the preservation and admissibility of evidence.

Emerging technologies such as artificial intelligence (AI) have also made a profound impact on digital forensics. AI-powered tools and algorithms have revolutionised the analysis of vast volumes of digital evidence, enabling investigators to uncover hidden

patterns, classify data, and prioritise their efforts. These advancements have accelerated the investigative process and enhanced the accuracy of findings, contributing to more efficient and effective digital forensic investigations (Balushi et al., 2022).

Furthermore, the proliferation of internet of things (IoT) devices has expanded the scope of digital forensics (Yaqoob et al., 2019). These interconnected devices, found in homes, businesses, and public spaces, have the potential to yield valuable evidence (Koroniotis et al., 2019). However, IoT forensics presents unique challenges, including data acquisition, device heterogeneity, and data integrity. Researchers and practitioners in the field have been actively developing techniques and methodologies to address these challenges and extract meaningful insights from IoT devices (Tageldin and Venter, 2023).

Figure 2 Types of data in digital forensics



Source: Breitinger and Jotterand (2023)

As shown in Figure 2, data in the context of digital forensics can be categorised into different types. Synthetic data is generated by software with some degree of autonomy, while random data is unique and often used for validation purposes. Rule-based data generation is deterministic and ideal for forensic tool testing, while (computer) simulated data relies on existing functionality and can be practical for generating complex data. Test data and scenario data are produced by simulators for specific or complex scenarios, respectively. AI-generated data, although not common yet, is expected to become more prevalent in the future. Human-driven dataset creation involves manual interaction and should be shared for reproducibility. Experimental data is produced through organised activities, and real-world data is created by humans without the intention of creating a forensic dataset.

The rapid expansion of IoT devices has significantly increased the risk of cyber-attacks, with a 37% rise in global IoT malware in the first half of 2023, totalling 77.9 million attacks. This sharp increase, compared to 57 million attacks in the same period of 2022, highlights the growing security challenges posed by IoT ecosystems. These devices generate vast amounts of network flow data, making manual forensic analysis impractical and necessitating automated, intelligent methods for effective cybersecurity. However, existing digital forensic solutions often fail to balance accuracy, efficiency, and computational feasibility, limiting their applicability in real-world scenarios. There is a pressing need for a lightweight, optimised forensic framework that integrates deep learning techniques to detect cyber threats while maintaining forensic integrity.

The complexity of IoT networks further complicates forensic investigations, requiring advanced techniques to analyse massive, decentralised datasets securely. Federated learning (FL) has emerged as a promising solution, allowing devices to collaboratively train models while keeping data localised, thereby enhancing privacy and reducing transmission overhead. Additionally, deep learning and AI advancements offer new opportunities to improve attack detection, forensic efficiency, and investigative accuracy. Despite these advancements, there remains a gap in developing user-friendly, scalable, and high-performing forensic tools tailored for IoT environments. Addressing these challenges is crucial to strengthening IoT security, optimising digital forensic processes, and enabling real-time cyber threat mitigation.

This research proposes a lightweight yet precise model for detecting abnormal events and cyber-attacks in IoT environments. It will assess the model's generalisation across diverse device types, ensuring robustness in heterogeneous settings. Furthermore, the study will explore how FL enhances the model's adaptability to emerging attack patterns while preserving data privacy in distributed IoT networks. By integrating cutting-edge AI and FL techniques, this work aims to advance digital forensic capabilities, providing an effective, scalable solution for modern cybersecurity challenges.

The rest of this paper is structured as follows: Section 2 provides background information on digital forensics in the IoT, with an emphasis on deep learning and FL. Section 3 presents a literature review that covers AI and hybrid techniques in IoT applications. Section 4 outlines the methodology, detailing the phases of the proposed hybrid deep-FL framework. Section 5 discusses the results, analysing the findings and their implications. Finally, Section 6 concludes the research and touch upon future work. Building on this foundation, we delve into the technical background of IoT forensics and FL to contextualise our approach.

2 Background

This section provides a background of the role of digital forensics in the IoT, highlighting how deep learning has been utilised to detect attacks within this environment, as well as the contribution of FL in this context.

2.1 Digits forensics in IoT

As criminal activities shift to the digital realm, law enforcement has adapted through digital forensics, which began in 1984 with the FBI developing computer investigation

programs. Various organisations have proposed definitions and standards, leading to different investigation models that share common phases. Rodney McKemmish defined digital forensics as "the process of identifying, preserving, analysing, and presenting digital evidence in a manner that is legally acceptable" (Pollitt, 1995). The digital forensic research workshop (DFRWS) investigation model, introduced in 2001, outlines six key phases: identification, preservation, collection, examination, analysis, and presentation (Yusoff et al., 2011).

Simultaneously, the IoT has emerged, creating a network of interconnected devices that communicate with each other. This network allows for machine-to-machine interactions and on-demand services at lower costs (Ronen et al., 2017).

In this IoT landscape, digital forensics plays a critical role in collecting and preserving digital artefacts from these devices, enabling investigators to reconstruct events and assess security breaches. However, challenges arise from compromised IoT devices, which can exhibit behavioural fingerprints like excessive resource consumption. Key challenges for digital forensics include detecting relevant data sources, a lack of high-quality data for developing automated tools, privacy concerns, and the geographically dispersed nature of contemporary cyberattacks (Kaur and Kaur, 2012).

Detecting incidents is a crucial first step in the digital investigation process. As IoT device usage increases, the importance of digital forensics in addressing cyber threats becomes even more pronounced.

2.2 Deep learning for attack detection in IoT

While the IoT ecosystem offers significant benefits in automation and efficiency, it is increasingly vulnerable to cyber-attacks due to its expansive attack surface. Traditional security measures may fall short in effectively detecting and mitigating threats, as IoT devices often operate in real-time and are deployed in diverse environments.

Deep learning has emerged as the preferred approach for network forensics (NFs) due to its ability to identify intricate patterns, execute efficiently, and perform well with substantial amounts of data (Hazarika and Medhi, 2016; Sun et al., 2015). As a subset of artificial neural networks characterised by a deep architecture with multiple hidden layers, deep learning models excel in analysing logs, network traffic, and sensor data. This analysis often requires reverse-engineering to detect signs of an attack, which can be challenging for humans to perform iteratively. Consequently, various machine learning models have been utilised to leverage their discriminative capabilities (Meffert et al., 2017).

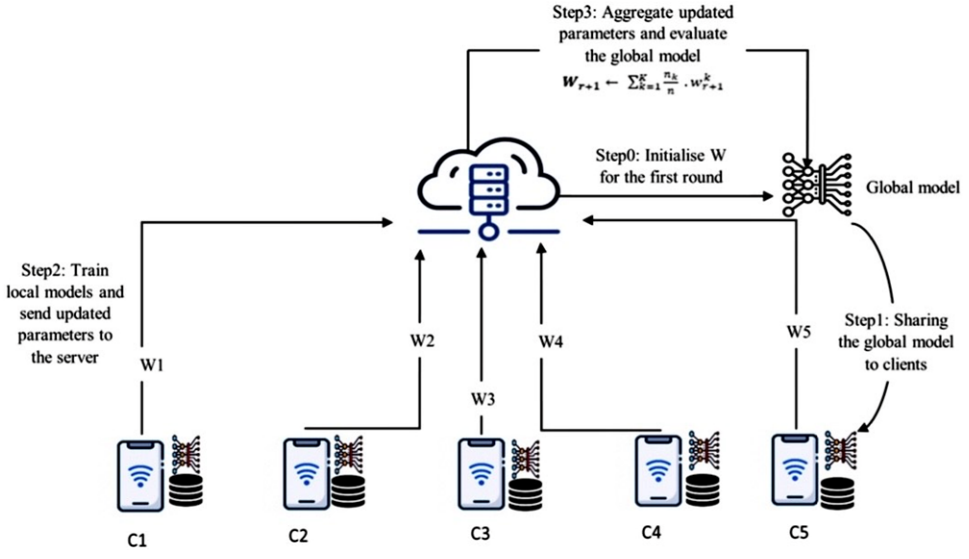
In recent years, deep learning has garnered increased attention from the research community due to its capacity to deeply understand data and its variations through the use of generative and discriminative models. By stacking numerous hidden layers, deep learning enhances the predictive capabilities of neural networks, enabling the identification of complex patterns within the data (Hossain et al., 2018).

As the IoT landscape continues to expand, the integration of deep learning for attack detection presents a promising avenue for improving the resilience of connected devices against cyber threats. This approach not only helps identify known attack patterns but also detects novel and sophisticated threats that may otherwise go unnoticed.

2.3 FL for attack detection in IoT

FL has emerged as a promising approach for enhancing security in the IoT. As represented in Figure 3, this decentralised method allows devices to collaboratively learn from local data without transmitting sensitive information to a central server, addressing critical privacy concerns. As IoT devices generate vast amounts of data, traditional centralised methods become less effective and more vulnerable to attacks (Yang et al., 2022).

Figure 3 FL process (see online version for colours)



Source: Mohamed et al. (2023)

In attack detection, FL enables devices to improve their security measures by training local models and sharing only the model updates. This approach helps in recognising diverse attack patterns while maintaining data privacy. Recent advancements in FL have optimised communication and reduced data exchange, improving the efficiency of anomaly detection in IoT networks (Liu et al., 2023).

By facilitating collaborative learning, FL enhances the resilience and adaptability of attack detection mechanisms, making it a vital framework for evolving threats in the IoT landscape.

3 Literature review

This section introduces what has been done so far. The related works introduced different approaches; some of them used AI algorithms, some used hybrid solutions by utilising AI algorithms with other techniques such as blockchain and hash algorithms. The previous work applied their approaches using different IoT environments/applications such as smart cities, smart homes and, vehicles networks.

Nwafor and Olufowobi (2019) introduced a visualisation framework aimed at improving the detection of abnormal events within an IoT ecosystem. The framework is designed to extract valuable insights from data interactions by visually depicting system events in IoT devices. It holds significant potential for applications such as digital forensic analysis, identification of system faults, intrusion detection, and situational awareness for system administrators and consumers. The implementation details of the framework were explored using a smart home system as a specific use case.

Verma et al. (2019) proposed an efficient digital forensic framework that enhances automation while safeguarding suspect data privacy. The framework incorporates case information, case profile data, and expert knowledge, utilising machine learning algorithms to provide investigators with relevant evidence. It strikes a balance between investigative requirements and the privacy of irrelevant suspect files, improving investigation efficiency without compromising outcomes. The study also presents a machine learning implementation for predicting the evidential relevance of files in a forensic image. Although baseline algorithms showed high false negative rates (FNRs), the use of bagging techniques significantly reduced false negatives. Machine learning techniques for assessing privacy showed promising results with the k-means algorithm, despite less favourable results for a specific class.

Brotsis et al. (2019) focused on the smart home domain, introducing a blockchain-based solution that utilises a private forensic evidence database and a permissioned blockchain. This solution addresses security services such as integrity, authentication, and non-repudiation. Areas of high priority include the detection of compromised devices and the collection of evidence regarding malicious behaviour in IoT networks. To tackle these challenges, the proposed solution employs intrusion detection systems (IDS) and distributed ledger technology (DLT) in the form of a blockchain. The system utilises mechanisms at a smart home's gateway for profiling, monitoring, and anomaly detection of IoT devices. This enhances the detection of known threats and zero-day vulnerabilities, allowing for immediate forensic evidence collection. Collected data, along with metadata necessary for correlation and investigation, are stored in an evidence database hosted by the internet service provider (ISP). Metadata are published on a blockchain maintained by ISPs, enabling law enforcement agencies (LEAs) to trace back attacks to their sources. The proposed solution, the cyber-trust blockchain (CTB), enables entities involved in the investigation process, such as LEAs and prosecutors, to access and handle digital evidence. It ensures the chain-of-custody by recording and preserving the chronological history of evidence handling. The CTB solution is built on Hyperledger Fabric and designed as a permissioned blockchain to meet privacy requirements.

Elhoseny et al. (2020) proposed an IoT-enabled optimal deep learning-based convolutional neural network (ODL-CNN) for suspect identification in forensic sketch synthesis (FSS). The DL-CNN model's hyperparameters were optimised using the improved elephant herd optimisation (IEHO) algorithm. The proposed method involves capturing surveillance videos using IoT-based cameras and feeding them into the ODL-CNN model. The method includes pre-processing with contrast enhancement using Gamma correction. The ODL-CNN model generates sketches of the input images, which are then compared to professional sketches based on eyewitness directions. When the similarity between the sketches is high, the suspect is identified. The presented ODL-CNN model was evaluated qualitatively and quantitatively, demonstrating effective

performance with a high peak signal to noise ratio (PSNR), structural similarity (SSIM), and accuracy. Simulation analysis demonstrated an average PSNR of 20.11dB, average SSIM of 0.64, and average accuracy of 90.10%.

Koroniotis et al. (2020) introduced a NFs framework called particle deep framework (PDF) for identifying and tracing attack behaviours in IoT networks. The PDF framework consists of three key functions:

1 extracting and verifying network data flows to handle encrypted networks

2 using a particle swarm optimisation (PSO) algorithm to adapt deep learning parameters automatically

3 developing a deep neural network (DNN) based on the PSO algorithm to detect and trace abnormal events in IoT networks.

The PDF framework was evaluated using Bot-IoT and UNSW_NB15 datasets and compared to other deep learning techniques. Experimental results demonstrated high accuracy of 0.999, a false positive rate (FPR) of 0, and a FNR of 9.5E-5, with a processing speed of 14,762 records per second.

Neaimi et al. (2020) addressed the challenge faced by digital forensic experts in finding evidence from corrupted files, often tampered with by criminals to hide evidence. The researchers proposed a solution using a convolutional neural network (CNN) trained on a dataset of various file types. The model can identify file types even if the files are damaged. The proposed technique randomly changes the header and footer of the file's hexadecimal value, which is used to identify file types. Additionally, one of the first or last five elements of the trained file is randomly modified to detect the file type even if the hexadecimal value changes. A proof of concept was demonstrated through a local web page that takes files as input, reads their hexadecimal values, and tests them against the trained model.

Wiyono and Cahyani (2020) discussed the problem of botnet activities in the IoT and the lack of effective NFs techniques to identify and track these activities. The researchers proposed the use of the decision tree C4.5 algorithm combined with network flow identification as a classification technique for conducting NFs. The researchers highlighted that many IoT devices were insecure and susceptible to cyberattacks, including distributed denial of service (DDoS) attacks, spamming, and phishing. However, there was a lack of NFs techniques to classify and track sophisticated botnet activities. To address this, the researchers developed a new classification technique based on the identification of network flows. They combined feature selection and the decision tree C4.5 algorithm to effectively identify, classify attacks, and assist in tracing botnet activity in the IoT. They used the Bot-IoT dataset for their research, which included features that explained different instructions and showed different botnet attack activity in an IoT network.

Saba et al. (2022) explored the utilisation of deep learning algorithms for fortifying the IoT system of a smart city against cyber threats. The researchers employed various machine learning classifiers and a deep learning model for intrusion detection, utilising seven datasets from the TON_IoT telemetry dataset. Their approach aimed to enhance the efficiency of IDSs by employing machine learning algorithms to detect anomalies in IoT network systems. The TON_IoT telemetry dataset was utilised in their experiments, employing machine learning classifiers and a deep learning model, resulting in an

impressive accuracy of 99.7%. The intrusion detection datasets included Thermostat, GPS Tracker, Garage Door, and Modbus.

Sangher et al. (2022) conducted an analysis of a botnet dataset using deep learning classification to assess the efficacy of deep learning in forensic analysis. They proposed a unique comparison between AI approaches, such as support vector machines (SVMs) and K-nearest neighbours (KNN), and deep learning approaches, particularly neural networks, to determine the superior approach in learning attack patterns. The study delved into IoT forensics models applied to a composite information repository formed by combining results from the analysis of the Torii botnet test with the CTU-13 dataset of botnet attacks on IoT environments. Methodologies, experimental approaches, and tools like Wireshark for network traffic analysis and feature extraction were discussed. Results encompassed the extraction of various features from the Torii botnet analysis, including registry paths, process spawning, and attack patterns.

Mohamed et al. (2023) investigated the application of FL in digital forensics for IoT networks. Addressing challenges in investigating cyber-attacks in IoT environments due to the volatile and heterogeneous nature of IoT devices, they proposed a deep FL-based method. The approach involved training a CNN model locally on IoT network data using FL rounds. Only the learned hyperparameters were shared with the federated server instead of the evidence data, ensuring data privacy. Evaluation using the ToN-IoT dataset showcased the proposed method's superiority, achieving an 81.69% detection accuracy in less training time while prioritising data privacy.

Avanija et al. (2023) introduced a novel NFs framework, PDF, designed for IoT networks. The PDF framework aimed at identifying and tracing attack activities in IoT networks by collecting network data flows and verifying their integrity using a PSO algorithm. The study utilised two datasets, 'Bot-IoT' and 'UNSW NB15', to assess the PDF framework's performance. Integrating deep learning methods, the PDF framework employed a DNN trained with the PSO algorithm for the detection and tracing of cyber-attack occurrences.

Djenna et al. (2023) proposed a methodology based on unsupervised long short-term memory (LSTM) and supervised CNN models for early identification and detection of botnet attacks. Evaluation using the CTU-13 and IoT-23 datasets demonstrated the method's superior performance, achieving a success rate of over 98.7% with a low FPR of 0.04%.

Mazhar et al. (2022) introduced an intelligent forensic analysis system for detecting attacks on IoT devices through a machine-to-machine (M2M) framework. The framework consisted of four modules, including attack traffic generation, traffic analysis and log generation, forensic server utilisation, and machine learning model application for attack detection. The decision tree algorithm demonstrated the best performance with 97.29% accuracy among various machine learning models.

Arshad et al. (2022) presented a framework for intelligent forensic analysis and detection of attacks on IoT devices using a node-to-node (N2N) approach. The framework incorporated forensic tools and machine learning techniques for identifying attack types. Machine learning algorithms, such as RFC classifier, DT classifier, Naive Bayes classifier, LDA classifier, MLP classifier, and ensemble (voting classifier), were employed, with the decision tree algorithm exhibiting the highest accuracy at 97.29%. The framework addressed IoT device security and evidence collection.

Zhang et al. (2023) introduced a malware detection approach combining CNNs with memory forensics. By leveraging the symmetric features of malware, the method achieved a high prediction accuracy of up to 97.48% and effectively detected fileless attacks, outperforming common machine learning methods.

Almutairi and Moulahi (2023) proposed a FL framework for training models locally on IoT devices with a focus on data privacy. The trained models were aggregated using blockchain technology, ensuring lightweight blockchain and efficient gas consumption. The framework demonstrated high accuracy (over 98%) using MLP in the FL phase and efficient gas consumption in the blockchain.

As we have reviewed, the landscape of digital forensics in IoT environments are significantly influenced by the emergence of AI techniques (Nwafor and Olufowobi, 2019; Verma et al., 2019; Elhoseny et al., 2020; Koroniotis et al., 2020; Neaimi et al., 2020; Zhang et al., 2023). However, a major concern is the lack of detailed validation for proposed frameworks, which raises questions about their reliability and robustness (Nwafor and Olufowobi, 2019). Additionally, the absence of clear implementation details for processes like generating provenance graphs and categorising trace data adds ambiguity to forensic methodologies (Nwafor and Olufowobi, 2019).

Various AI techniques, including CNNs, decision tree C4.5, PSO, DNNs, and LSTM, have been widely applied (Verma et al., 2019; Elhoseny et al., 2020; Koroniotis et al., 2020; Neaimi et al., 2020; Zhang et al., 2023). Notable outcomes include the C4.5 decision tree showing promise in classifying botnet activities (Verma et al., 2019), while CNNs demonstrated high accuracy in identifying tampered files (Neaimi et al., 2020). FL methods, particularly those using CNN models, effectively identified cyber-attacks in IoT environments while ensuring data privacy (Zhang et al., 2023).

Despite these advancements, limitations and opportunities for improvement remain (Nwafor and Olufowobi 2019; Verma et al., 2019; Elhoseny et al., 2020; Koroniotis et al., 2020; Neaimi et al., 2020; Zhang et al., 2023). Dataset challenges, such as limited diversity and availability, present opportunities for future research to explore more comprehensive data sources (Verma et al., 2019; Zhang et al., 2023). Concerns about real-time responsiveness in IoT data suggest the need for enhancements to make forensic frameworks more adaptable to dynamic settings (Elhoseny et al., 2020).

The combination of deep learning algorithms with optimisation techniques emerges as a promising avenue (Koroniotis et al., 2020; Zhang et al., 2023). Hybrid approaches could deliver efficient and cost-effective solutions, addressing resource considerations in real-world IoT systems (Koroniotis et al., 2020; Zhang et al., 2023). Additionally, research gaps in FL cooperation and the generalisability of defences highlight areas for improvement (Zhang et al., 2023).

As a final point, the current state of digital forensics in IoT environments is marked by dynamic advancements and evolving challenges. The choice of AI algorithms depends on specific applications, with decision tree C4.5, CNN, and FL showing substantial promise. Addressing limitations related to dataset diversity, real-time responsiveness, and cooperation in FL will enhance the robustness and applicability of forensic frameworks. Exploring hybrid solutions that combine deep learning with optimisation algorithms presents an exciting trajectory for future research, offering potential for efficient and cost-effective solutions within the IoT domain.

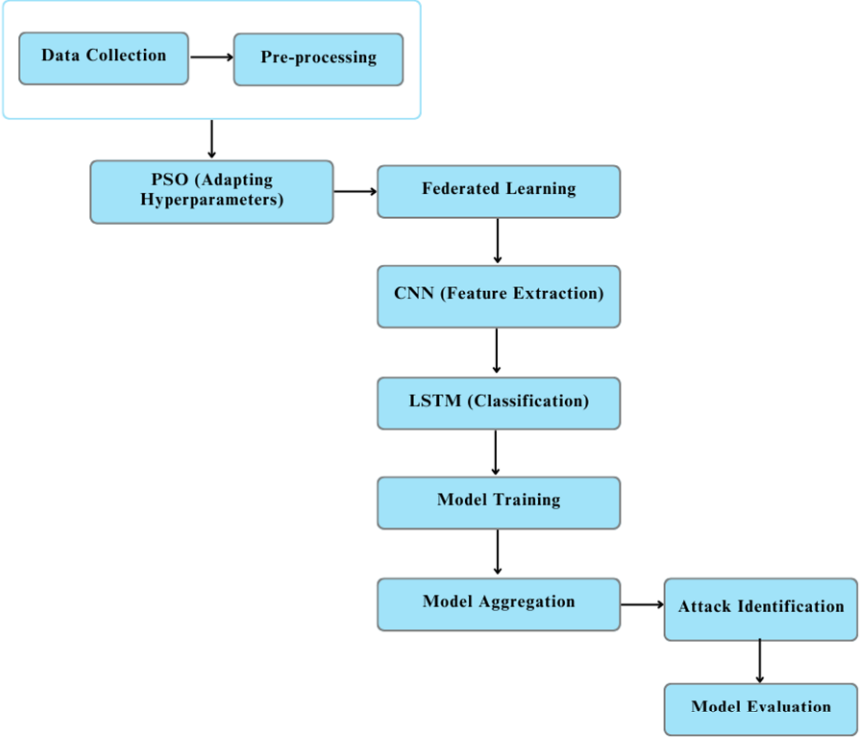
4 Proposed hybrid deep-FL framework for digital forensic

The studies referenced earlier tend to be centralised, which can lead to significant time consumption and high computational costs. Additionally, the data weight can be substantial, making real-world application challenging, especially given the limited resources of IoT devices. Moreover, these approaches often compromise data privacy, as sensitive forensic data must be transmitted to centralised servers for processing, increasing the risk of exposure.

To tackle these issues, we present the phases of the hybrid deep and FL forensics framework. This framework leverages a combination of CNNs, LSTM networks, PSO, and FL to enhance digital forensic analysis in IoT environments. CNNs were chosen for their ability to efficiently extract spatial patterns from IoT data, making them well-suited for recognising attack signatures and anomalies in network traffic. LSTMs, on the other hand, are essential for capturing temporal dependencies and sequential patterns in IoT data, enabling the model to analyse attack progression over time and detect complex cyber threats that evolve dynamically.

To further optimise the deep learning models, PSO is integrated into the framework for hyperparameter tuning. Unlike traditional optimisation techniques, PSO efficiently searches for optimal model parameters with reduced computational overhead, enhancing accuracy while minimising the need for extensive manual tuning. This makes the model more adaptive and computationally efficient, addressing the resource constraints of IoT devices.

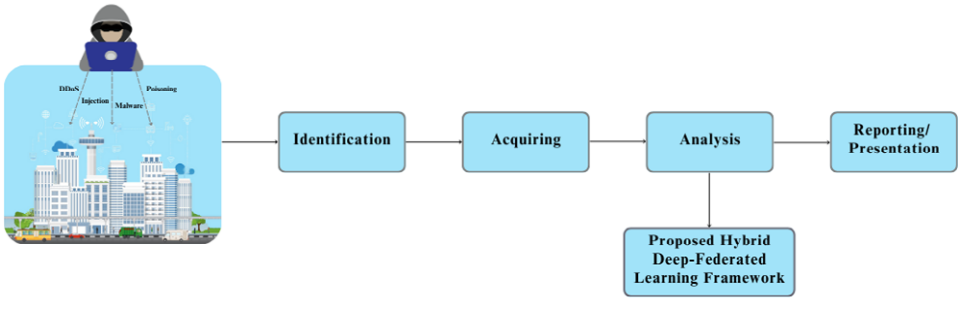
Figure 4 Proposed hybrid deep-FL framework (see online version for colours)



Additionally, FL is incorporated to ensure data privacy and reduce the reliance on centralised processing. By enabling IoT devices to collaboratively train models while retaining data locally, FL mitigates privacy concerns and minimises communication costs. This decentralised learning approach also enhances model generalisation by allowing training across diverse IoT environments, improving resilience against evolving attack patterns. Moreover, the combination of FL with CNN-LSTM and PSO fosters collaboration, minimises bias, and maintains flexibility across different network conditions, as illustrated in Figure 4. In the following sections, we will explore each phase of the proposed framework in detail.

The framework is particularly suited for the Analysis stage of digital forensics, as illustrated in Figure 5. Where it processes data such as logs and network traffic to classify them as normal or malicious. The model's metrics, including accuracy and recall, underscore its ability to detect patterns and anomalies effectively. Beyond its primary role in analysis, the model demonstrates versatility by supporting the identification stage, flagging suspicious activities in real time, and contributing to the reporting stage by providing detailed classifications and performance insights. This adaptability allows the framework to enhance multiple phases of digital forensics while maintaining its core focus on efficient and accurate analysis.

Figure 5 Role of the proposed hybrid deep-FL framework in digital forensics phases (see online version for colours)



4.1 Phase 1: data collection and pre-processing

4.1.1 Data collection

We have utilised five datasets in this experiment to strengthen the reliability and applicability of the model. By drawing from a variety of sources, we aimed to ensure that the model could effectively handle diverse scenarios and reduce potential biases that might stem from relying on just one dataset. This strategy not only provided a more comprehensive evaluation of the model's performance but also enhanced its ability to generalise to real-world applications. Ultimately, this approach led to deeper insights and more meaningful conclusions. We will provide a brief overview of each, Table 1 give a description about the datasets:

- *ToN_IoT:* The ToN_IoT dataset (Moustafa, 2024) encompasses diverse telemetry data collected from IoT and IIoT sensors, alongside network traffic from various operating systems, including Windows (7 and 10) and Ubuntu (14 and 18). Created in a virtualised environment, it simulates interactions between IoT, cloud, and

edge/fog systems. This dataset captures both normal operations and various cyber-attacks, such as DoS, DDoS, and ransomware, targeting IoT gateways and web applications.

- *UNSW_NB15:* UNSW_NB15 (Moustafa, 2024) released by the Australian Center for Cyber Security in 2015. It features raw network packets generated using the IXIA Perfect Storm tool, covering nine distinct attack scenarios, including DoS and reconnaissance. With over 2.5 million network traffic streams comprised of both benign and malicious activities.
- *Bot_IoT:* Developed within the Cyber Range Lab at UNSW Canberra Cyber, the Bot_IoT dataset simulates a realistic network environment characterised by normal and botnet traffic. It includes various attack types, such as DDoS and keylogging, organised by the protocols employed, providing insights into the behaviour of botnets in IoT contexts (Koroniotis et al., 2019).
- *UNB_CIC_IoT_2023:* The UNB_CIC_IoT_2023 dataset (Neto et al., 2023) originates from the University of New Brunswick's Centre for Cybersecurity. It features extracted network traffic data from 105 IoT devices subjected to 33 different cyber-attacks, including DDoS, spoofing, and the Mirai botnet, highlighting the vulnerabilities of IoT infrastructures.
- *RT_IoT_2022:* The RT-IoT2022 dataset (Sharmila and Nagapadma, 2023) is derived from a real-time IoT infrastructure, offering a comprehensive view of network dynamics involving both typical and malicious activities. It includes data from various IoT devices and simulates sophisticated attack scenarios such as SSH brute-force attacks and DDoS assaults, captured meticulously using the Zeek network monitoring tool.

4.1.2 Pre-processing

In the pre-processing phase, several techniques are applied to prepare the data for efficient model training and improve performance. Here is a detailed explanation of each technique used:

- *Splitting data into features and targets:* In this step, the data is divided into two parts: features (independent variables) and targets (dependent variables). Features are the input variables used by the model to make predictions. They can be both numeric (e.g., packet size, number of requests) and categorical (e.g., protocol type, service). Targets represent the variable the model is trained to predict, such as attack types or labels indicating whether the event is normal or an attack. Splitting the data helps structure it in a way that is easy to feed into machine learning models, separating what the model needs to learn from what it needs to predict.
- *Pipeline for text data:* This step consists of several sub-techniques. Handling missing values is done using a 'SimpleImputer', which fills in missing categorical data, typically using the most frequent value from the dataset to ensure continuity. This ensures that no missing data disrupts the training process, particularly in categorical variables that often need to be complete for encoding. Encoding is applied through 'OneHotEncoder', which converts categorical variables into numerical format by

creating binary columns for each category. Machine learning models require inputs to be numeric, and encoding transforms categories into a format the models can understand. Finally, standardisation is applied with 'StandardScaler', which normalises the encoded features by subtracting the mean and scaling to unit variance. This process ensures that features with larger numerical values do not dominate the training process.

- *Pipeline for numeric data:* Handling missing values is similarly applied to numeric data using 'SimpleImputer', where missing values are filled with the mean of the column to avoid introducing extreme values. Standardisation is then applied to normalise the numeric features, again using 'StandardScaler', which scales the values based on their mean and variance. Standardising helps to prevent models from being biased toward features with larger numeric ranges, enabling the model to learn more effectively.
- *ColumnTransformer:* A 'ColumnTransformer' is used to apply different transformations to specific columns of the dataset based on their data type. For example, the text processing pipeline is applied to categorical columns, while the numeric processing pipeline is applied to numerical columns. The 'ColumnTransformer' ensures that both pipelines are applied to the appropriate features in the dataset, enabling simultaneous processing of mixed data types (text and numeric).
- *Addressing class imbalance with data augmentation:* Many datasets suffer from class imbalance, where certain attack types are underrepresented. To tackle this, data augmentation techniques like random 'OverSampling' and synthetic minority over-sampling technique (SMOTE) are applied. Random 'OverSampling' increases the number of instances in minority classes by duplicating existing samples, preventing the model from becoming biased toward the majority class. In addition, SMOTE offers a more advanced method by generating new synthetic instances through interpolation between similar minority class samples. Rather than simply duplicating, SMOTE creates diverse examples that provide the model with more meaningful variation. By balancing the dataset using these techniques, the model is better equipped to generalise across all classes, resulting in more accurate predictions and preventing bias toward the majority.
- *Conversion to TensorFlow datasets:* Once pre-processing is completed, the data is converted into TensorFlow datasets, which are optimised for use within deep learning frameworks. This step involves batching the data (e.g., in batches of 32) so that each batch is processed sequentially during training. TensorFlow's Dataset API is used for efficient data handling, reducing memory usage and accelerating training. Converting to TensorFlow datasets ensures that the deep learning model can handle large-scale data efficiently.
- *Application of necessary augmentations:* In addition to oversampling, other augmentations such as shuffling and feature scaling might be applied. TensorFlow's built-in augmentation functionalities, like 'shuffle()', ensure that the data is randomised, preventing the model from memorising the order of the data. This step further improves generalisation and prevents overfitting.

- *Final processed data ready for neural network training:* After pre-processing, the data is fully standardised, balanced, and transformed into a format optimised for neural network training. Each batch of data is carefully designed to optimise memory use and speed up the training process, which is particularly important for complex models like CNNs and LSTMs. These pre-processing steps ensure that the hybrid model can handle mixed data types (text and numeric), balance class distribution, and process the data efficiently for training, ultimately improving the model's performance and generalisation across various attack types.

4.2 Phase 2: PSO

In this proposed framework, PSO (Wang et al., 2018) is used to fine-tune the hyperparameters of a hybrid LSTM-CNN model. PSO mimics the social behaviour of birds flocking or fish schooling to find optimal solutions. Here, each 'particle' in the swarm represents a candidate solution, specifically a set of hyperparameters like the number of neurons in the LSTM and dense layers. These particles traverse the solution space, guided by both their personal best position and the global best position (the best solution discovered by any particle so far).

As the particles explore, they adjust their positions based on a balance of exploration (searching new areas) and exploitation (improving known good areas), refining the model's architecture for optimal performance. The fitness of each particle is evaluated based on the model's validation loss after training on the dataset for one epoch. The goal is to minimise this loss, meaning the PSO algorithm drives the search toward configurations that improve model performance. Ultimately, PSO identifies the best-performing hyperparameters for constructing the final model, offering a more efficient and effective approach to model tuning compared to manual trial and error.

4.3 Phase 3: FL

In our framework, the FL process is implemented by simulating multiple clients, each training a model independently on local data, followed by the aggregation of the models' learned parameters to create a shared global model. First, ten clients are simulated, each with access to a unique portion of the training data. Each client receives a separate instance of the CNN model and trains it on a subset of 1,000 samples, representing a decentralised data scenario.

The FederatedLearning class orchestrates the FL process by managing both local training and the aggregation of client models. In each round of training, every client locally trains its model using its dataset for a specified number of epochs. The model weights from each client are saved and later combined through a process called federated averaging (FedAvg). This method calculates the average of the weights from all clients for each model layer, effectively combining their learned knowledge into a single global model.

Table 1 Description of the datasets

Dataset	*Total size*	*Classes*	*Number of features*	*Balance*	*Missing values*	*Augmentation applied*	*Balance before augmentation*	*Balance after augmentation*
ToN_IoT	190,753	8: normal, backdoor, injection, password, DDoS, Ransomware, XSS, scanning	6	Imbalanced	No	Yes	Normal: 15,000 Attack: 24,944	Normal: 91,265 Attack: 91,265
UNSW_NB15	175,341	10: normal, generic, exploits, fuzzers, DoS, reconnaissance, analysis, backdoor, shellcode, worms	49	Imbalanced	No	Yes	Normal: 56,000 Attack: 119,341	Normal: 56,000 Attack: 56,000
Bot_IoT	27,289	2: normal, attack	26	Imbalanced	Yes	Yes	Normal: 7,290 Attack: 19,999	Normal: 7,290 Attack: 7,290
UNB_CIC_IoT2023	252,744	8: normal, DDoS, DoS, reconnaissance, web-based, brute-force, spoofing, Mirai botnet	47	Imbalanced	Yes	Yes	Normal: 50,498 Attack: 202,246	Normal: 50,498 Attack: 50,498
RT_IoT2022	123,117	9: DoS SYN Hping, thing speak, ARP poisoning, MQTT Publish, NMAP UDP scan, NMAP Xmas tree scan, NMAP OS detection, NMAP TCP scan, other	83	Imbalanced	No	Yes	Normal: 94,659 Attack: 118,265	Normal: 94,659 Attack: 94,659

After the weights are aggregated, all clients' models are updated with the new global weights, ensuring they start the next round of training with the same shared model. This process of local training, followed by weight aggregation, repeats for a set number of rounds. After the training rounds are completed, the global model from one client is tested on a separate test dataset, and its accuracy is evaluated. Since all clients' models are identical after the final aggregation, the performance on the test data is representative of the global model.

This approach ensures data privacy (Zhu et al., 2019), as each client only shares its model weights rather than raw data. The FedAvg method effectively combines the knowledge from all clients into a single model, leading to better generalisation without requiring centralised data storage.

4.4 Phase 4: hybrid deep learning model

4.4.1 Data splitting

Each dataset was split into 80% training and 20% testing sets using stratified sampling to maintain class balance. Both pipelines were applied across datasets to ensure that all features were standardised.

The hybrid model integrates CNNs and LSTM networks to process sequential data, such as time series.

4.4.2 CNN component

The Conv1D layer detects patterns using 50 filters with a kernel size of 3, allowing the model to learn distinct local features. The ReLU activation introduces nonlinearity, while 'same' padding ensures the output shape matches the input size. This is followed by batch normalisation, which improves the training process by standardising activations.

4.4.3 LSTM component

The LSTM layer processes the extracted features from the CNN component, with eight units and return_sequences = false, which allows the network to output a single vector representation of the input sequence. LSTMs are well-suited for handling temporal dependencies in time series data.

4.4.4 Integration

After LSTM processing, the data is flattened and passed through fully connected dense layers for classification. The first dense layer contains 50 neurons, followed by a dropout layer to prevent overfitting. Another dense layer of 50 neurons is used before the final output. The sigmoid activation function at the output layer produces a binary probability, classifying the input into one of two categories.

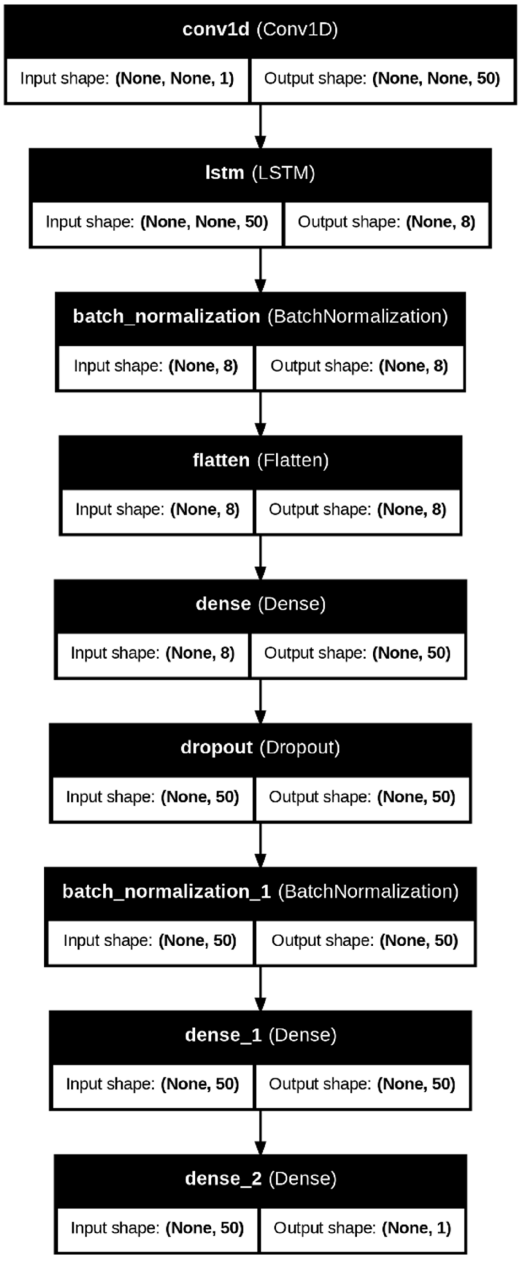
4.4.5 Attack identification

Attack Identification is the core function of the hybrid model, aimed at detecting and classifying network anomalies into either normal or attack types.

4.4.6 Feature extraction

The CNN component captures spatial patterns from the data, essential for identifying characteristic traits of different network activities. The LSTM component handles temporal dependencies, which are critical for understanding sequential behaviours in network traffic that may indicate attacks.

Figure 6 Internal structure of hybrid model



4.4.7 Classification

After feature extraction, dense layers classify the data into normal or attack types. The final layer of the model applies a sigmoid activation function, outputting a probability that determines if the input corresponds to an attack or a normal event. This combination of feature extraction and classification enables the model to accurately detect various network attacks, leveraging both spatial and temporal information.

As depicted in Figure 6, this architecture effectively combines the sequential processing capabilities of LSTMs with the pattern detection power of CNNs, enabling accurate binary classification.

4.5 Phase 5: model evaluation

The purpose of this evaluation is to assess the model's performance, aiming to determine its potential for broader applications beyond just the training data. When evaluating models, especially in classification tasks, the confusion matrix is often used, which includes four key elements. These elements form the basis for many common performance metrics (Sokolova and Lapalme, 2009; Goodfellow et al., 2016), described below.

- *True positive (TP):* This refers to instances where the classifier correctly identifies an attack. Essentially, it means that the model successfully recognised an event as belonging to the attack class.
- *True negative (TN):* In this case, the model correctly identifies that the observed event is normal, meaning there's no attack.
- *False positive (FP):* Here, the model incorrectly flags normal event as an attack, producing a false alarm.
- *False negative (FN):* This occurs when the model fails to detect an attack, mistakenly labelling the malicious event as normal.
- *Accuracy (success rate):* Accuracy measures the proportion of both normal activities and attacks that the model correctly classifies. It is calculated by dividing the number of correct classifications (both TP and TN) by the total number of instances (TP, TN, FP, and FN). Mathematically:

$$\text{Accuracy} = \text{TP} + \text{TN}/\text{TP} + \text{TN} + \text{FP} + \text{FN} \tag{1}$$

- *Precision:* Precision indicates the proportion of predicted attacks that were correctly identified as actual attacks. It is calculated by dividing the true positives by the total predicted positives (TP + FP):

$$\text{Precision} = \text{TP}/\text{TP} + \text{FP} \tag{2}$$

- *Recall (detection rate):* Recall measures the proportion of actual attacks that were correctly identified by the model. It reflects how many attacks the model correctly detected out of the total number of attacks in the dataset. Mathematically:

$$\text{Recall} = \text{TP}/\text{TP} + \text{FN} \tag{3}$$

- *F1-score (harmonic mean):* The F1-score combines both precision and recall into a single value, providing a balanced measure of a model's performance. It is the harmonic mean of precision and recall, calculated as:

$$\text{F1-score} = 2 \times (\text{precision} \times \text{recall}) / (\text{precision} + \text{recall}) \tag{4}$$

- *FPR:* This metric shows the percentage of normal traffic that was incorrectly classified as malicious. The FPR is calculated by dividing the false positives by the total number of normal traffic instances (TN + FP):

$$\text{FPR} = \text{FP} / \text{TN} + \text{FP} \tag{5}$$

- *FNR:* This metric shows the percentage of actual malicious traffic that was incorrectly classified as normal. The FNR is calculated by dividing the false negatives by the total number of actual malicious instances (TP + FN):

$$\text{FNR} = \text{FN} / \text{TP} + \text{FN} \tag{6}$$

- *Error rate (ER):* This metric indicates the percentage of all classifications that were incorrect, encompassing both false positives (FP) and false negatives (FN). The ER is calculated as follows:

$$\text{ER} = (\text{FP} + \text{FN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \tag{7}$$

- *Time efficiency (TE):* Time efficiency refers to the ability to complete a task in the shortest possible time. To calculate the execution time of a model or algorithm, the start time is subtracted from the end time. This measurement is then compared to a benchmark to evaluate the model's efficiency in terms of time.

$$\text{Time efficiency (TE)} = \text{End time} - \text{Start time} \tag{8}$$

With evaluation metrics established, we now assess the model's performance across various datasets in terms of accuracy, precision, recall, F1-score, FPR, FNR, ER, and efficiency.

5 Experiment results and discussion

In this section, we will analyse and discuss the performance of our proposed model across various datasets. The evaluation focuses on key metrics such as accuracy, precision, recall, FPR, FNR, F1-score, efficiency (time consumed), and ER. By comparing these results, we aim to assess the model's effectiveness in detecting and classifying attacks, as well as identifying areas for improvement.

5.1 Environment

The proposed framework was developed on Google Colab Pro with 50.99 GB of RAM, an NVIDIA Tesla T4 GPU. Python 3 code was employed to build and train the CNN-LSTM model, as well as to identify hyperparameters using PSO. The following Python packages were utilised: NumPy, Pandas, and Scikit-learn for matrix manipulation

and data pre-processing. Keras and TensorFlow for building the model and FL, Pyswarm for the PSO process.

5.2 Experiment results and discussion

The performance metrics across various datasets indicate strong results in attack detection and classification using the proposed model, with notable differences in the performance depending on the dataset, as shown in Table 2.

The characteristics of each dataset played a crucial role in shaping the model's performance. For ToN_IoT, the model achieved perfect scores across all performance metrics (100% accuracy, precision, recall, and F1-score), with no false positives or false negatives. This highlights the dataset's alignment with the model's learning capabilities. The well-defined attack patterns and balanced distribution likely contributed to the model's ability to distinguish between normal and malicious activities with absolute precision, consuming just 77.168 seconds.

Table 2 Proposed model performance across different datasets

Dataset	*Accuracy*	*Precision*	*Recall*	*FPR*	*FNR*	*F1-score*	*Efficiency (time consumed)*	*ER*
ToN_IoT	100%	100%	100%	0%	0%	100%	77.168 s	0%
UNSW_NB15	95.45%	95.56%	95.45%	6.07%	3.4%	95.44%	93.85 s	4.8%
Bot_IoT	99.98%	99.98%	99.98%	0%	0.025%	99.98%	181.342 s	0.0125%
UNB_CIC_IoT2023	79.93%	77.46%	71.25%	1.31%	28.24%	69.21%	158.919 s	20.07%
RT_IoT2022	98.97%	96.46%	94.92%	0.97%	5.43%	95.54%	98.188 s	1.03%

In contrast, UNSW_NB15 posed more challenges due to its complex attack types and imbalanced distribution, leading to a slightly lower accuracy (95.45%) and recall (95.45%), with a noticeable FPR of 6.07% and a FNR of 3.4%. The presence of adversarial samples and overlapping feature spaces between normal and attack traffic likely caused some misclassifications. The efficiency was moderate at 93.85 seconds, with an ER of 4.8%, indicating that while the model classified most instances correctly, some false alarms and missed detections remained.

The Bot_IoT dataset, despite its complexity, resulted in near-perfect performance (99.98% accuracy) with minimal FNR (0.025%), reflecting the model's exceptional ability to classify the attack types with high precision. However, the model required more processing time (181.342 seconds), indicating the computational effort needed due to the dataset's size and variability. The ER was also incredibly low at 0.0125%, highlighting the model's efficiency in handling the dataset.

On the other hand, UNB_CIC_IoT2023 proved to be the most challenging, with the lowest accuracy (79.93%) and the highest FNR (28.24%). This suggests the dataset contained a higher degree of noise, ambiguous attack instances, or possibly newer attack patterns that the model struggled to classify correctly. The feature distributions may not have been as distinct, making it harder for the model to form clear decision boundaries. The efficiency was also lower at 158.919 seconds, with an ER of 20.07%, indicating significant room for improvement in adapting to this dataset.

Lastly, the RT_IoT2022 dataset yielded strong results with a 98.97% accuracy and an F1-score of 95.54% suggest some difficulty in classifying certain attack types. However, a small FNR of 5.43% and FPR of 0.97% indicate that some attacks were not correctly identified, though these values remain low. The efficiency remained competitive at 98.188 seconds, with an ER of just 1.03%.

Therefore, while the model exhibits high performance across most datasets, its effectiveness can vary depending on the dataset characteristics, particularly in terms of false positive and FNRs. As shown in Figures 6, 7, 8, 9 and 10. On the positive side, the model demonstrates strong generalisation capabilities, achieving consistently high precision and recall on most datasets. Its ability to maintain low ERs and high efficiency in terms of execution time further supports its potential for practical, real-world applications.

Figure 6 ToN_IoT confusion matrix (see online version for colours)

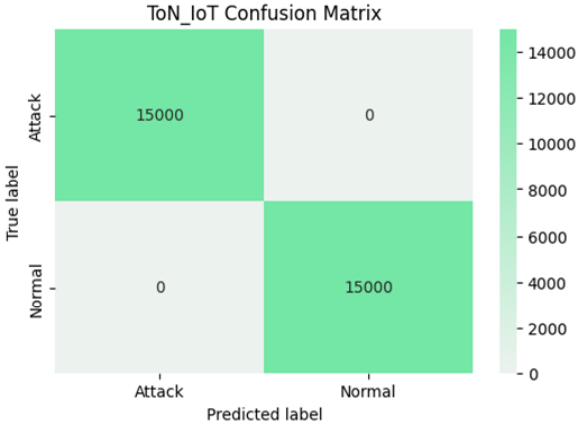


Figure 7 UNSW_NB15 confusion matrix (see online version for colours)

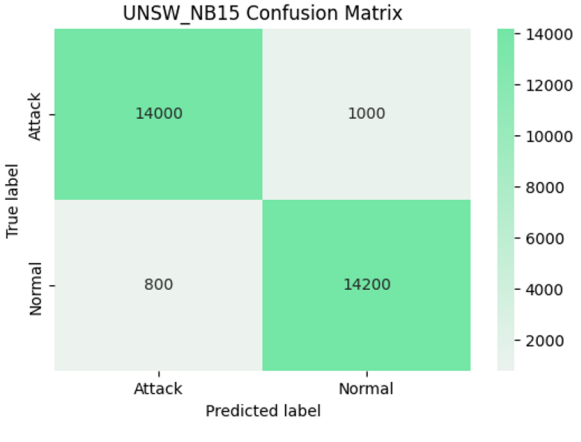


Figure 8 Bot_IoT confusion matrix (see online version for colours)

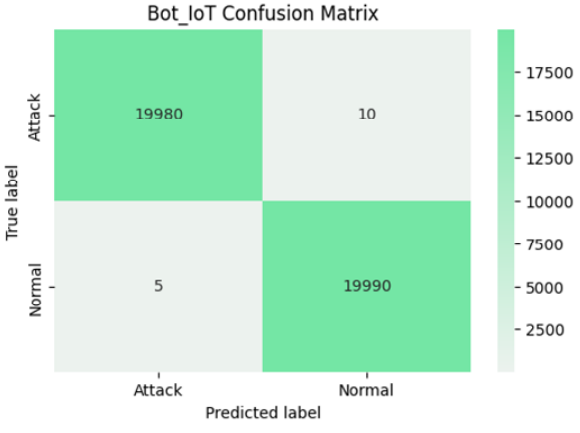


Figure 9 UNB_CIC_IoT2023 confusion matrix (see online version for colours)

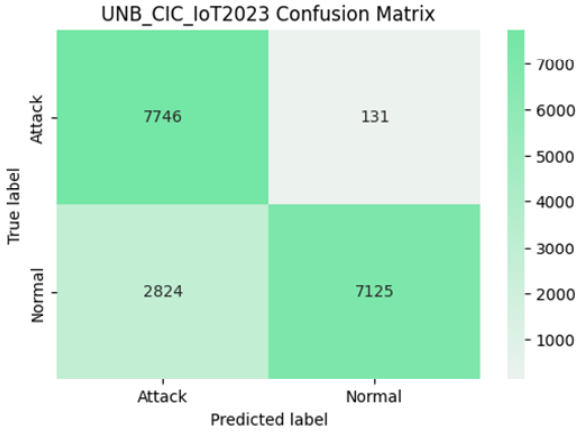


Figure 10 RT_IoT2022 confusion matrix (see online version for colours)

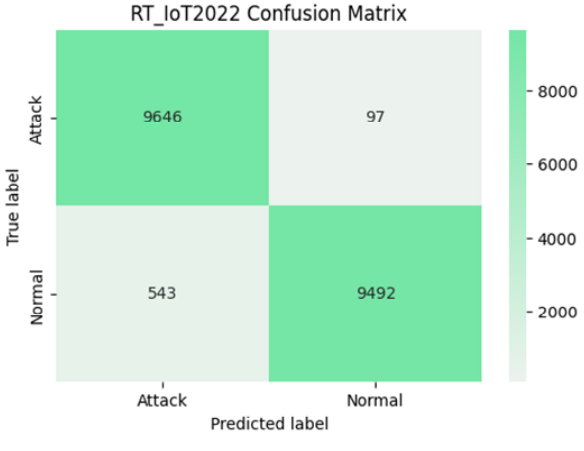


Figure 11 clearly shows that the model achieves perfect scores across most metrics for the ToN_IoT dataset, with the highest efficiency (execution time). UNSW_NB15 and Bot_IoT also demonstrate strong performance, particularly in terms of accuracy and precision, though efficiency is slightly lower for Bot_IoT due to its longer execution time. The UNB_CIC_IoT2023 dataset stands out with relatively lower recall and F1-scores, along with higher FPR and FNR, indicating more difficulty in identifying attacks in this dataset. The efficiency (execution time) for this dataset is also higher, suggesting a trade-off between accuracy and speed. Finally, the RT_IoT2022 dataset shows balanced performance, with slightly reduced recall and F1-scores but relatively low ERs and moderate efficiency.

The line chart in Figure 12 highlights the variability in ERs (FPR and FNR) across the datasets, reflecting the model's strengths and limitations in handling different challenges. The ToN_IoT and Bot_IoT datasets show negligible ERs, with both FPR and FNR close to 0%, demonstrating the model's exceptional performance. However, for UNSW_NB15, the FPR (6.07%) and FNR (3.4%) are moderately higher, suggesting room for improvement in reducing false positives and missed detections. The most significant challenge is seen in UNB_CIC_IoT2023, where the FNR spikes to 28.24%, indicating difficulty in detecting attacks within this dataset. The RT_IoT2022 dataset exhibits relatively low ERs, with a manageable FPR (0.97%) and FNR (5.43%), showcasing reliable performance. Overall, this visualisation underscores the model's robustness for most datasets but emphasises the need for further optimisation to address datasets with complex attack patterns.

Figure 11 Comparison of performance metrics across different datasets (see online version for colours)

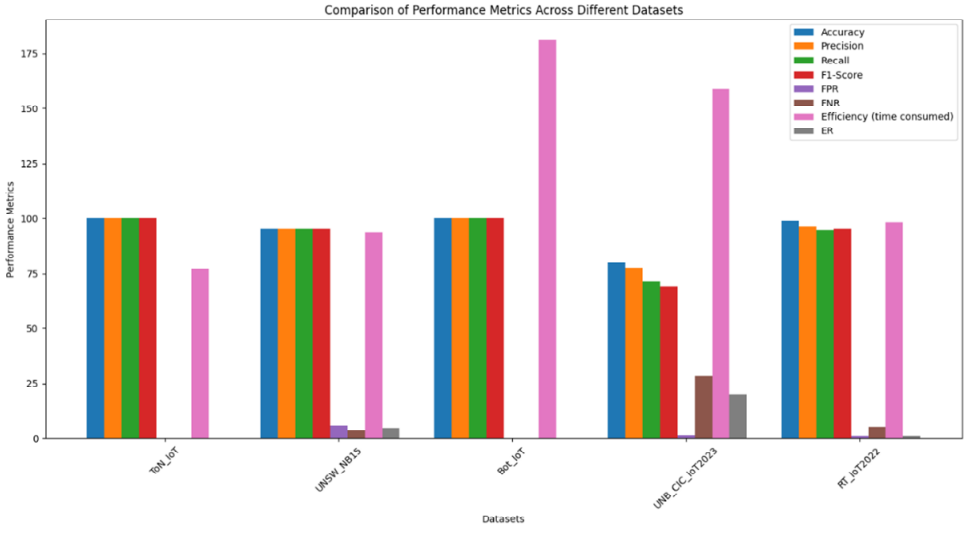


Figure 13 compares the accuracy of each dataset before and after FL. As shown, the accuracy either remains the same or improves slightly after FL. For example, the Ton_IoT dataset maintains a perfect 100% accuracy, while UNSW_NB15 shows a significant improvement from 88% to 95.45%. This indicates that FL does not compromise the model's ability to classify correctly and, in many cases, can improve the

overall performance. The improvements can be attributed to the distributed nature of FL, which aggregates local models trained on diverse datasets, allowing the model to generalise better across different scenarios. FL utilises a distributed architecture, where training is carried out across multiple devices, enabling parallel processing. This leads to a significant reduction in total training time, as the datasets are processed locally rather than centrally, avoiding bottlenecks.

Figure 12 Error rate across datasets (see online version for colours)

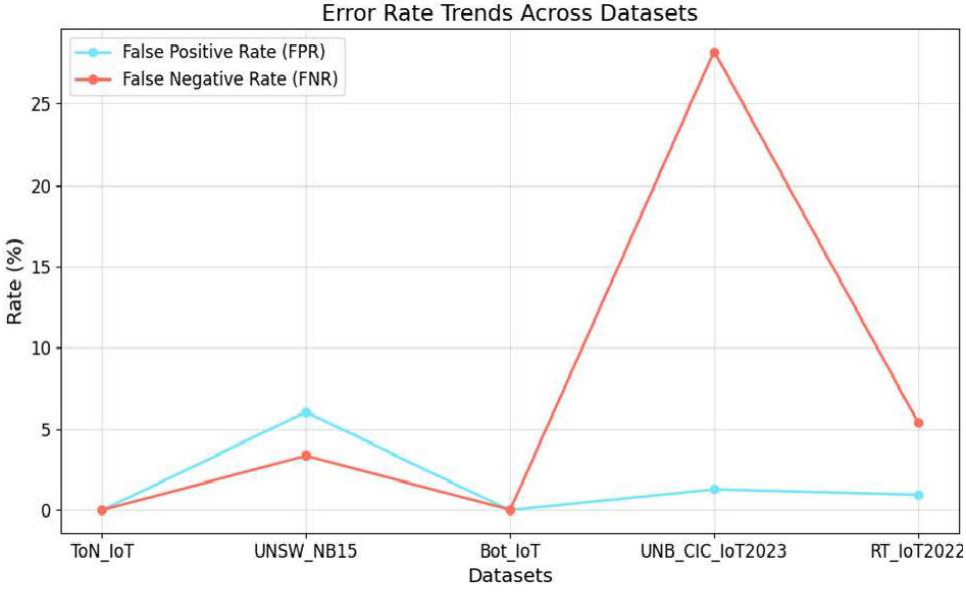
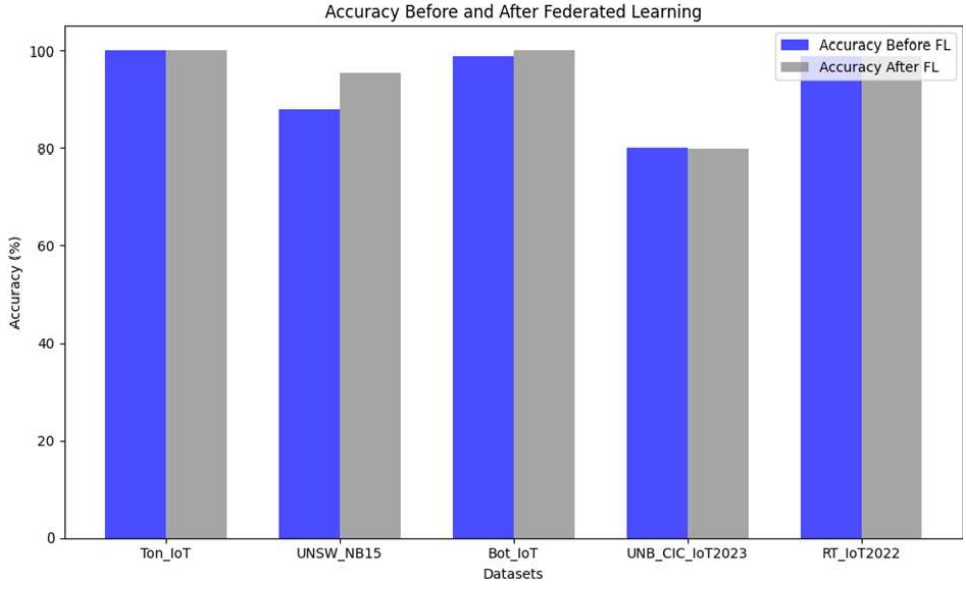
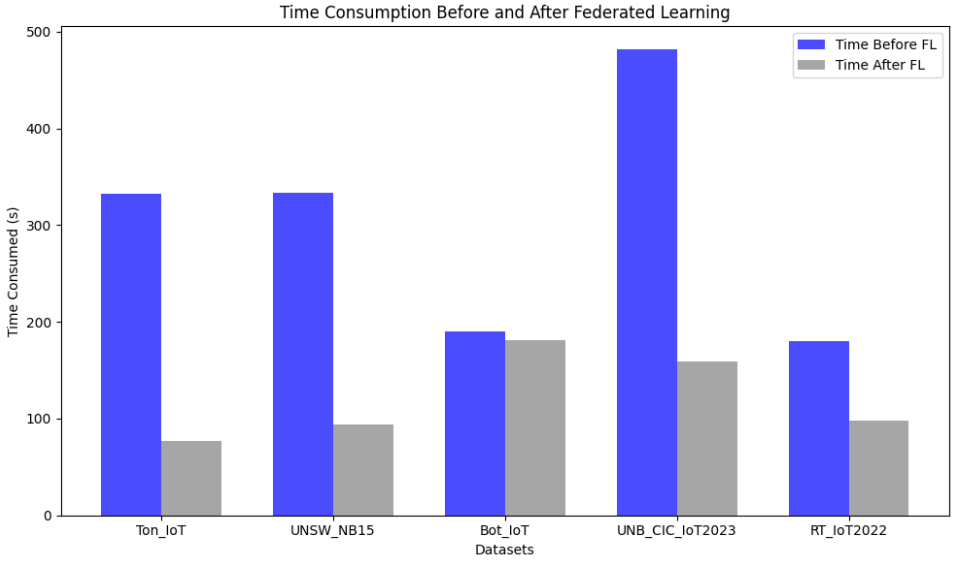


Figure 13 accuracy before and after federated learning (see online version for colours)



In Figure 14, the chart focuses on time efficiency. Across all datasets, FL dramatically reduces the time required for training. For instance, the ToN_IoT dataset sees a reduction from 332.865 s to 77.168 s (76.82% improvement), while UNSW_NB15 experiences a drop from 333.794 s to 93.85 s (71.88% improvement). The other datasets also show notable time savings, with a 4.85% improvement in Bot_IoT, 67.00% in UNB_CIC_IoT 2023, and 45.58% in RT_IoT2022. This reduction in time can be attributed to distributed training and the use of local data. FL allows training to happen locally on multiple devices, reducing the need for data to be uploaded to a central server. The local processing of data results in quicker computations, avoiding the network overhead associated with centralised models. Training on local data in each device reduces the need for data transfers to a central server, significantly cutting down on time while ensuring that updates are applied more quickly.

Figure 14 Time consumption before and after federated learning (see online version for colours)



5.3 Addressing real-world deployment challenges

While the proposed hybrid deep learning model with FL has demonstrated high accuracy across multiple datasets, real-world deployment introduces challenges such as latency, device heterogeneity, class imbalance, and real-time processing constraints. One of the key concerns in FL-based IoT security is latency, as devices with varying network conditions and computational capabilities may delay global model updates. To mitigate this, asynchronous federated learning (AFL) can be utilised, allowing devices to contribute updates at different times rather than following a strict synchronisation schedule. Additionally, edge computing can be integrated to process data locally, reducing reliance on cloud-based computations and minimising network delays. Another significant challenge is device heterogeneity, where IoT environments consist of diverse hardware, ranging from low-power sensors to high-performance computing nodes. To accommodate these variations, techniques like adaptive model aggregation, knowledge

distillation, and selective client participation can be applied, ensuring that resource-constrained devices contribute meaningfully without being overburdened.

Another critical issue is class imbalance, where certain attack types, such as scanning or ransomware, have significantly fewer instances compared to others like normal or backdoor traffic. This imbalance can bias the model toward dominant classes, leading to misclassification of minority attack types. To address this, data augmentation, oversampling techniques (e.g., SMOTE), and cost-sensitive learning approaches can be integrated into the training process. Additionally, federated re-weighting can be employed to ensure that underrepresented classes are adequately learned without compromising overall model stability. Lastly, real-time constraints pose a challenge for IoT security applications, as attack detection must occur with minimal delay. Optimising inference time using lightweight architectures, model pruning, and quantisation can significantly reduce computational overhead while maintaining accuracy. Implementing priority-based anomaly detection, where high-risk alerts are processed first, can further enhance real-time responsiveness. By addressing these challenges, the proposed model can better adapt to real-world IoT environments, ensuring both robust attack detection and practical deployment feasibility.

The trade-off between detection accuracy and efficiency is a critical factor in assessing the practicality of deep learning models, particularly for real-world IoT security applications. The results demonstrate that the proposed hybrid deep-federated learning framework achieves an effective balance, sustaining high accuracy while keeping execution time within practical limits across different datasets. Unlike conventional deep learning models that often incur excessive delays as computational demands grow, the proposed model shows a well-optimised structure that delivers robust detection performance without prohibitive computational cost.

It is important to note that the framework does not always achieve both the highest accuracy and the lowest execution time simultaneously, as these metrics naturally vary across datasets. However, it consistently maintains a practical balance: for example, on the Bot_IoT dataset it achieves near-perfect accuracy (99.98%) despite higher execution time (181.342 s), while on ToN_IoT it combines perfect accuracy (100%) with significantly lower execution time (77.168 s). These complementary outcomes highlight the adaptability of the model, proving that it can remain both accurate and efficient under different conditions. Such flexibility is essential for real-world IoT environments, where data characteristics vary and maintaining reliable performance with feasible cost is critical.

5.4 Comparative analysis of model results with previous studies

In comparison to previous studies as presented in Table 3, our proposed hybrid CNN-LSTM model optimised with PSO and FL demonstrates competitive performance across several key metrics. Koroniotis et al. (2020) achieved 99% accuracy using a multi-layer perceptron (MLP) with PSO, showing high precision and recall scores. Similarly, Avanija et al. (2023) also reached 99.9% accuracy using a DNN with PSO, achieving near-perfect precision, recall, and F1-score values, which indicate the robustness of PSO as an optimiser in deep learning models.

Saba et al. (v) employed a voting classifier, achieving 99.7% accuracy and slightly higher precision and recall scores (99.8% and 99.8%, respectively), outperforming the

proposed model by a small margin. However, it is important to note that while their model excelled in precision, our model offers a more balanced trade-off between precision (96.46%) and recall (95.45%), making it a robust choice for diverse scenarios.

Wiyono and Cahyani (2020) achieved a comparable accuracy of 97.62% using the decision tree C4.5, though their recall was significantly higher at 99.99%. This suggests that decision tree models may perform well in terms of recall, but they may not generalise as effectively across varying datasets compared to our hybrid approach.

Sangher et al. (2022) implemented a CNN-based FL model with significantly lower accuracy (81.69%) and suboptimal precision and recall scores (34.93% and 69.38%, respectively). This highlights the improvement in performance that can be achieved by incorporating PSO as an optimisation technique within a FL framework, as seen in our model's results. Djenna et al. (2023), who used a hybrid LSTM-CNN, reported an accuracy of 98.74%, which is slightly higher than our model, but their recall was lower at 88.6%, suggesting that their model may be prone to missing some attack types.

Lastly, Zhang et al. (2023) utilised a CNN model and reported a strong performance, with a precision of 98.71% and recall of 96.22%. Although their results are slightly better in precision and recall than ours, the FL and PSO combination in our approach offers a more balanced and flexible framework that can be deployed across different datasets while maintaining high efficiency and reliability.

Overall, while some of the previous models have marginally higher accuracy and precision rates, our model strikes a balance between accuracy, precision, recall, and efficiency, making it well-suited for environments where computational resources are a concern. The integration of FL and PSO further allows for scalable, real-time learning across decentralised systems, which is not addressed in many of the previous models.

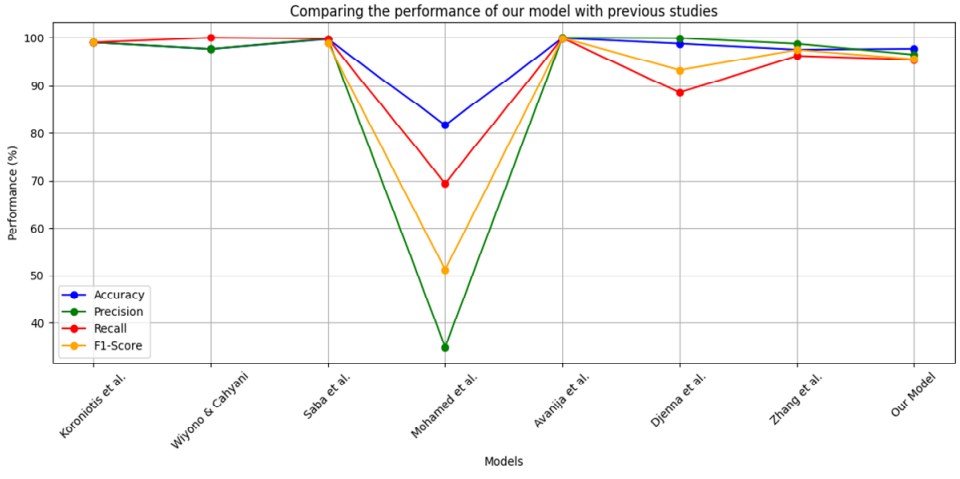
Table 3 Comparing the performance of our model with previous studies

Model reference	*Technique*	*Accuracy*	*Precision*	*Recall*	*F1-score*
Koroniotis et al. (2020)	MLP, PSO	99%	99%	99%	99%
Wiyono and Cahyani (2020)	Decision tree C4.5	97.62%	97.63%	99.99%	NA
Saba et al. (2022)	Voting classifier	99.7%	99.8%	99.8%	98.9%
Mohamed et al. (2023)	CNN based on federated learning	81.69%	34.93%	69.38%	51.18%
Avanija et al. (2023)	DNN, PSO	99.9%	100%	99.9%	99.9%
Djenna et al. (2023)	Hybrid LSTM, CNN	98.74%	99.9%	88.6%	93.3%
Zhang et al. (2023)	CNN	97.48%	98.71%	96.22%	97.45%
Our model	Hybrid CNN, LSTM with PSO and federated learning	97.66%	96.46%	95.45%	95.54%

The line graph in Figure 15 illustrates the comparative performance of the proposed hybrid CNN-LSTM model alongside other referenced models, highlighting the consistency of our model across all four key metrics. Notably, the graph emphasises areas where our model, though slightly lower in precision and recall compared to some, strikes a strong balance in overall performance. Additionally, the significant drop in accuracy

and recall seen in models like Mohamed et al. underscores the improvements gained by incorporating PSO into our FL framework. This visual representation reinforces how our model remains competitive while offering flexibility and generalisability across diverse datasets.

Figure 15 Comparing the performance of our model with previous studies (see online version for colours)



Our proposed hybrid CNN-LSTM model, optimised with PSO and enhanced by FL, demonstrates a significant advantage over existing models due to its efficiency and lightweight nature. These characteristics make it particularly suitable for real-time, resource-constrained environments.

One of the primary reasons our model outperforms others is its superior time efficiency. After applying FL, training times across datasets were dramatically reduced. For example, the ToN_IoT dataset saw a drop from 332.865 seconds to 77.168 seconds, and UNSW_NB15 decreased from 333.794 seconds to 93.85 seconds. This substantial improvement in time efficiency comes without any compromise in performance, as accuracy remained consistently high. The distributed nature of FL allows local training, avoiding network bottlenecks and making the overall process faster compared to centralised models.

In addition to its time efficiency, the model's lightweight design ensures competitive performance without the need for extensive computational resources. By integrating CNN-LSTM with PSO, our model achieves optimal hyperparameter tuning, resulting in high accuracy, precision, and recall. For instance, the model achieved 100% accuracy on the ToN_IoT dataset and 99.98% on Bot_IoT. These results show the model's ability to generalise well across different datasets while maintaining low resource consumption, which is critical for applications like IoT systems.

Furthermore, the model benefits from FL's scalability, enabling it to handle decentralised and dynamic data in real-time. This feature allows the model to update quickly and adapt to new data, making it highly practical for large-scale or distributed systems where traditional centralised models struggle.

In comparison to other models, our approach excels by maintaining a strong balance between performance metrics and resource efficiency. While some models may achieve marginally higher accuracy, they often do so at the cost of longer training times or greater resource demands. Our model, on the other hand, offers a practical solution that combines high accuracy with significantly improved time efficiency and lower computational overhead, making it well-suited for real-world, resource-limited environments.

6 Conclusions and future work

Where the IoT is reshaping our lives, this research introduces a groundbreaking hybrid framework that harnesses the power of deep learning and FL to detect anomalous behaviours in IoT devices. By combining these advanced AI techniques, our model not only excels in identifying cyber threats but also champions data privacy – an essential consideration in today's interconnected world. The proposed framework demonstrates remarkable accuracy and efficiency, showcasing its potential to revolutionise digital forensics and cybersecurity. This study presents a hybrid CNN-LSTM model optimised with PSO and FL, achieving superior efficiency in IoT forensics. By decentralising data with FL, the model ensures improved privacy, and PSO optimally tunes parameters for peak accuracy. As IoT devices proliferate, so do the complexities of securing them, making this innovative approach not just timely but vital.

Moving forward, improvements will be taken into consideration: first, exploring the framework's applicability across various domains, such as healthcare, and industrial IoT, to highlight its versatility and adaptability. Second, the development of advanced visualisation tools will allow forensic investigators to intuitively analyse detected anomalies, thereby enhancing decision-making processes and improving incident response times. Future research could explore this framework's application in healthcare and industrial IoT to assess its adaptability and efficiency across domains.

Acknowledgements

The authors gratefully acknowledge Qassim University, represented by the Deanship of Graduate Studies and Scientific Research, on the financial support for this research under the number (QU-J-PG-2-2025-53012) during the academic year 1446 AH/2024 AD.

References

Almutairi, W. and Moulahi, T. (2023) 'Joining federated learning to blockchain for digital forensics in IoT', *Computers*, August, Vol. 12, No. 8, p.157, MDPI AG, https://doi.org/10.3390/computers12080157.

Arshad, M.Z. et al. (2022) 'Digital forensics analysis of IoT nodes using machine learning', *Journal of Computing & Biomedical Informatics*, December, Vol. 4, No. 1, https://doi.org/10.56979/401/2022/107.

Avanija, J. et al. (2023) 'Enhancing network forensic and deep learning mechanism for internet of things networks', *Journal of Scientific & Industrial Research*, May, Vol. 82, No. 5, https://doi.org/10.56042/jsir.v82i05.1084.

Balushi, Y.A. et al. (2023) 'The use of machine learning in digital forensics: review paper', *Proceedings of the 1st International Conference on Innovation in Information Technology and Business (ICIITB 2022)*, pp.96–113, Atlantis Press International BV, https://doi.org/10.2991/978-94-6463-110-4_9.

Breitinger, F. and Jotterand, A. (2023) 'Sharing datasets for digital forensic: a novel taxonomy and legal concerns', *Forensic Science International: Digital Investigation*, July, Vol. 45, p.301562, https://doi.org/10.1016/j.fsidi.2023.301562.

Brotsis, S. et al. (2019) 'Blockchain solutions for forensic evidence preservation in IoT environments', *2019 IEEE Conference on Network Softwarization (NetSoft)*, IEEE, June, https://doi.org/10.1109/netsoft.2019.8806675.

Djenna, A. et al. (2023) 'Unmasking cybercrime with artificial-intelligence-driven cybersecurity analytics', *Sensors*, July, Vol. 23, No. 14, p.6302, MDPI AG, https://doi.org/10.3390/s23146302.

Elhoseny, M. et al. (2020) 'Optimal deep learning based convolution neural network for digital forensics face sketch synthesis in internet of things (IoT)', *International Journal of Machine Learning and Cybernetics*, July, Vol. 12, No. 11, pp.3249–60, Springer Science and Business Media LLC, https://doi.org/10.1007/s13042-020-01168-6.

Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep Learning*, MIT Press, Cambridge, MA.

Hazarika, B. and Medhi, S. (2016) 'Survey on real-time security mechanisms in network forensics', *International Journal of Computer Applications*, Vol. 151, No. 2, pp.1–4.

Hossain, M., Karim, Y. and Hasan, R. (2018) 'FIF-IOT: a forensic investigation framework for IoT using a public digital ledger', *2018 IEEE International Congress on Internet of Things (ICIOT)*, IEEE, pp.33–40.

Kaur, R. and Kaur, A. (2012) 'Digital forensics', *International Journal of Computer Applications*, Vol. 50, No. 5, pp.5–9.

Koroniotis, N. et al. (2019) 'Forensics and deep learning mechanisms for botnets in internet of things: a survey of challenges and solutions', *IEEE Access*, Institute of Electrical and Electronics Engineers (IEEE), Vol. 7, pp.61764–85.

Koroniotis, N. et al. (2019) 'Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset', *Future Generation Computer Systems*, Vol. 100, No. C, pp.779–796.

Koroniotis, N. et al. (2020) 'A new network forensic framework based on deep learning for internet of things networks: a particle deep framework', *Future Generation Computer Systems*, September, Vol. 110, pp.91–106, Elsevier BV, https://doi.org/10.1016/j.future.2020.03.042.

Liu, Y. et al. (2023) 'Enhancing IoT security with federated learning: a novel approach', *Journal of Network and Computer Applications*.

Mazhar, M.S. et al. (2022) 'Forensic analysis on internet of things (IoT) device using machine-to-machine (M2M) framework', *Electronics*, April, Vol. 11, No. 7, p.1126, MDPI AG, https://doi.org/10.3390/electronics11071126.

Meffert, C. et al. (2017) 'Forensic state acquisition from internet of things (FSAIOT): a general framework and practical approach for IoT forensics through IoT device state acquisition', *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ACM, p.56.

Mohamed, H. et al. (2023) 'Digital forensics based on federated learning in IoT environment', *2023 Australasian Computer Science Week*, USA, ACM, January, https://doi.org/10.1145/3579375.3579387.

Moustafa, N. (2024) *ToN_IoT and UNSW15 Datasets*, UNSW Research, 16 February [online] https://research.unsw.edu.au/projects/toniot-datasets (accessed 16 February 2024).

Neaimi, M.A. et al. (2020) 'Digital forensic analysis of files using deep learning', *2020 3rd International Conference on Signal Processing and Information Security (ICSPIS)*, IEEE, November, https://doi.org/10.1109/icspis51252.2020.9340141.

Neto, E.C.P. et al. (2023) 'CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment', *Journal of Sensors*.

Nwafor, E. and Olufowobi, H. (2019) 'Towards an interactive visualization framework for IoT device data flow', *2019 IEEE International Conference on Big Data (Big Data)*, IEEE, December, https://doi.org/10.1109/bigdata47090.2019.9006450.

Pollitt, M. (1995) 'Computer forensics: an approach to evidence in cyberspace', *Proceedings of the National Information Systems Security Conference*, Vol. 2, pp.487–491.

Raval, H. (2020) 'Artificial intelligence forensics, machine learning forensics and digital forensics', *Digital Forensics (4N6) Journal*, November, https://doi.org/10.46293/4n6/2020.02.04.05.

Ronen, E. et al. (2017) 'IoT goes nuclear: creating a Zigbee chain reaction', *2017 IEEE Symposium on Security and Privacy (SP)*, IEEE, pp.195–212.

Sharmila, B.S. and Nagapadma, R. (2023) *RT-IoT2022*, UCI Machine Learning Repository, https://doi.org/10.24432/C5P338.

Saba, T. et al. (2022) 'Securing the IoT system of smart city against cyber threats using deep learning', in Gong, D. (Ed.): *Discrete Dynamics in Nature and Society*, June, pp.1–9, Hindawi Limited, https://doi.org/10.1155/2022/1241122.

Salih, K. and Dabagh, N. (2023) 'Digital forensic tools: a literature review', *Journal of Education and Science*, March, Vol. 32, No. 1, pp.109–24, University of Mosul, https://doi.org/10.33899/edusj.2023.137420.1304.

Sangher, K.S. et al. (2022) 'Implementation of threats detection modelling with deep learning in IoT botnet attack environment', *IoT with Smart Systems*, Springer Nature, Singapore, October, pp.585–92, https://doi.org/10.1007/978-981-19-3575-6_57.

Sokolova, M. and Lapalme, G. (2009) 'A systematic analysis of performance measures for classification tasks', *Information Processing & Management*, Vol. 45, No. 4, pp.427–437.

SonicWall (2023) *2023 SonicWall Cyber Threat Report*, SonicWall, July [online] http://www.sonicwall.com/2023-mid-year-cyber-threat-report (acceessed 15 July 2024).

Sun, J-R., Shih, M-L. and Hwanga, M-S. (2015) 'A survey of digital evidences for forensic and cybercrime investigation procedure', *IJ Network Security*, Vol. 17, No. 5, pp.497–509.

Tageldin, L. and Venter, H. (2023) 'Machine-learning forensics: state of the art in the use of machine-learning techniques for digital forensic investigations within smart environments', *Applied Sciences*, September, Vol. 13, No. 18, p.10169, MDPI AG, https://doi.org/10.3390/app131810169.

Verma, R. et al. (2019) 'DF 2.0: an automated, privacy preserving, and efficient digital forensic framework that leverages machine learning for evidence prediction and privacy evaluation', *The Journal of Digital Forensics, Security and Law*, ERAU Hunt Library – Digital Commons Journals, https://doi.org/10.15394/jdfsl.2019.1606.

Wang, D., Tan, D. and Liu, L. (2018) 'Particle swarm optimization algorithm: an overview', *Soft Computing*, Vol. 22, No. 2, pp.387–408.

Wiyono, R.T. and Cahyani, N.D.W. (2020) 'Performance analysis of decision tree C4.5 as a classification technique to conduct network forensics for botnet activities in internet of things', *2020 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, August, https://doi.org/10.1109/icodsa50139.2020.9212932.

Yang, K. et al. (2022) 'Federated learning for IoT: a comprehensive survey', *IEEE Internet of Things Journal*.

Yaqoob, I. et al. (2019) 'Internet of things forensics: recent advances, taxonomy, requirements, and open challenges', *Future Generation Computer Systems*, March, Vol. 92, pp.265–75, Elsevier BV, https://doi.org/10.1016/j.future.2018.09.058.

Yusoff, Y., Ismail, R. and Hassan, Z. (2011) 'Common phases of computer forensics investigation models', *International Journal of Computer Science & Information Technology*, Vol. 3, No. 3, pp.17–31.

Zhang, S. et al. (2023) 'A malware detection approach based on deep learning and memory forensics', *Symmetry*, March, Vol. 15, No. 3, p.758, MDPI AG, https://doi.org/10.3390/sym15030758.

Zhu, W. et al. (2019) *Federated Heavy Hitters Discovery with Differential Privacy*, 22 February, DOI: 10.48550/arXiv.1902.08534.