Future Generation Computer Systems 156 (2024) 191–205

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Digital Twins-enabled Zero Touch Network: A smart contract and explainable AI integrated cybersecurity framework

Randhir Kumar [a], Ahamed Aljuhani [b], Danish Javeed [c], Prabhat Kumar [d,*], Shareeful Islam [e], A.K.M. Najmul Islam [d]

[a] Department of Computer Science and Engineering, SRM University AP, AP 522240, India
[b] Department of Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia
[c] Software College, Northeastern University, Shenyang 110169, China
[d] Department of Software Engineering, LUT University, 53850 Lappeenranta, Finland
[e] School of Computing and Information Science, Anglia Ruskin University, UK,

ARTICLE INFO

Keywords:
Blockchain
Digital Twins
Explainable AI
Intrusion Detection System
Zero Touch Network

ABSTRACT

Data-driven modeling using Artificial Intelligence (AI) is envisioned as a key enabling technology for Zero Touch Network (ZTN) management. Specifically, AI has shown huge potential for automating and modeling the threat detection mechanism of complicated wireless systems. The current data-driven AI systems, however, lack transparency and accountability in their decisions, and assuring the reliability and trustworthiness of the data collected from participating entities is an important obstacle to threat detection and decision-making. To this end, we integrate smart contracts with eXplainable AI (XAI) to design a robust cybersecurity framework for ZTN. The proposed framework uses a blockchain and smart contract-enabled access control and authentication mechanism to ensure trust among the participating entities. Additionally, with the collected data, we designed Digital Twins (DTs) for simulating the attack detection operation in the ZTN environment. Specifically, to provide a platform for analysis and the development of an Intrusion Detection System (IDS), the DTs are equipped with a variety of process-aware attack scenarios. A Self Attention-based Long Short Term Memory (SALSTM) network is used to evaluate the attack detection capabilities of the proposed framework. Furthermore, the explainability of the proposed AI-based IDS is achieved using the SHapley Additive exPlanations (SHAP) tool. The experimental results using N-BaIoT and a self-generated DTs dataset confirm the superiority of the proposed framework over some baseline and state-of-the-art techniques.

1. Introduction

The next-generation network systems and emerging technologies such as Software-Defined Networking (SDN) and Network Functions Virtualization (NFV) have transformed wireless communication systems and information technology to achieve not only efficiency, reliability, and scalability but also to meet the anticipated demands of increased traffic rate with increased number of connected devices [1–3]. Despite these benefits, these systems also pose significant challenges relating to increased complexity, operational expenses, and capital expenditure. Additionally, data is collected and transmitted over insecure channels within the network which is vulnerable to a wide range of cyberattacks that can pose any potential risks relating to service disruption and network resources depletion [4,5]. Human-driven approaches and individual service-based configurations are commonly used but not adequately supported to tackle some of these challenges [6]. In this context, closed-loop automation has emerged as a promising solution for fully automated network operations and management services. In particular, Zero-Touch Network (ZTN) aims to automate all management and operational processes (e.g., planning, deployment, provisioning, monitoring, and optimization), without any human intervention [7]. However, applications deployed in ZTN still become a target of several security threats. Protecting ZTN as a whole requires the testing of the functionality of protection mechanisms like blockchain and Intrusion Detection System (IDS). Additionally, the threat landscape is continuously evolving with sophisticated attack techniques and traditional IDSs cannot effectively support the identification of the anomalies to ensure the security of ZTN.

* Corresponding author.
E-mail addresses: randhir.honeywell@ieee.org (R. Kumar), a_aljuhani@ut.edu.sa (A. Aljuhani), 2027016@stu.neu.edu.cn (D. Javeed), prabhat.kumar@lut.fi (P. Kumar), shareeful.islam@aru.ac.uk (S. Islam), najmul.islam@lut.fi (A.K.M.N. Islam).

https://doi.org/10.1016/j.future.2024.02.015
Received 25 August 2023; Received in revised form 12 February 2024; Accepted 16 February 2024
Available online 20 February 2024

Artificial intelligence (AI) has been adopted in modern IDS to extract important features, identify anomalies, and classify attacks [8–10]. Machine learning and deep learning, in particular, have emerged as promising security solutions when combined with IDS to mitigate various types of cyberattacks. Over the last few years, deep learning-based IDS have been widely used as they provide high accuracy and a low false positive rate with better performance when working with large amounts of data [11,12]. However, deep learning-based IDS are still viewed as black boxes due to the complexity of detection models and lack of explanation of the overall decision-making process [13]. Explainable Artificial Intelligence (XAI) is a new AI paradigm that provides techniques for interpreting machine learning-based IDS, allowing such models to explain the reason behind their prediction [14–16]. Furthermore, blockchain has become another promising solution for managing security and privacy issues in ZTNs due to its functionalities relating to peer-to-peer (P2P) and decentralized networking [17]. Blockchain can greatly benefit ZTN applications as it provides a decentralized network structure that ensures reliability, trust, and transparency with full autonomy of operations and management. Additionally, entities in such a connected network need to prove their identities to prevent unauthorized access to ZTNs. Hence, blockchain-based authentication schemes have been proposed in different networking architectures [18,19]. However, despite significant attention to using explainable deep learning-based IDS and blockchain in various application domains, the adoption of ZTNs is still in the early stages.

As stated above, it is necessary to test security solutions before deploying into ZTN's, but this task is challenging on a live ZTN system as well as time-consuming. Multiple incidents could occur as a result of the testing process on live systems, resulting in significant physical damage and business disruption [20]. Digital Twin (DT) is a cutting-edge technology that represents a system or machine through simulation, emulation, and mirroring of the physical entity. It can be used to evaluate and analyze how a system responds to cyberattacks within a simulated environment [20,21]. To this end, DT has the potential to become an enabler for improving ZTN cybersecurity by implementing explainable deep learning-based IDS and blockchain-based authentication mechanisms. There are a number of existing contributions that consider DTs as an enabler to improve security [22–24], but they mainly focus on examining the underlying concepts and overall architecture design of DTs. There is a lack of focus on DTs-enabled cybersecurity for ZTNs.

Within this context, this paper proposes a novel cybersecurity framework that aims to enhance the overall security of ZTN. The proposed framework is enabled by Digital Twins and adopts blockchain-based smart contracts and XAI-based IDS for this purpose. The key contributions of this paper are summarized as follows:

- A robust DT-enabled cybersecurity framework for ZTN is proposed by integrating smart contracts with explainable AI.
- Blockchain-enabled key establishment and access control mechanism is proposed that authenticates the participating entities within ZTN and ensures secure communication. The temper-proof property of blockchain ensures high integrity of the data enrichment and builds trust between the participating entities of the blockchain network. A smart contract-enabled Proof-of-Authority (PoA) consensus mechanism is used to verify and validate the transactions or data.
- Adoption of DT for evaluating the security issues of the blockchain-enabled authentication scheme and XAI-based IDS within the ZTN environment. In particular, DT is set up with a range of process-aware attack scenarios to provide a platform for study and the creation of an IDS. A Self Attention-based Long Short Term Memory (SALSTM) network and SHapley Additive exPlanations (SHAP) tool is deployed for this purpose. This allows us to assess the proposed framework's ability to identify and defend the attacks.
- The proposed framework is evaluated through experiments using an actual DTs simulated dataset and state-of-the-art intrusion dataset (N-BaIoT). The outcomes are compared with existing baseline and state-of-the-art techniques to show the effectiveness of the proposed cybersecurity framework.

The rest of this paper is organized as follows. In Section 2, we have discussed the related study. In Section 3, we have discussed the network and threat model to design the proposed cybersecurity framework. The proposed framework and its key components are discussed in Section 4. In Section 5, we have discussed the experimental analysis and comparison with baseline and state-of-the-art techniques. Finally, Section 6 highlights conclusion and future research.

2. Related works

The realm of DTs-driven cybersecurity has been the subject of significant research over the past years. This section outlines the contributions of various studies and their implications for the field.

2.1. Digital Twins (DTs) and Intrusion Detection Systems (IDS)

Varghese et al. [22] concentrated on the integration of IDS with DTs. They emphasized the design of IDS and attack simulation modules that are integrated with DTs. However, their work is limited to only a few types of attacks. Similarly, Suhail et al. [25] extended the capabilities of DTs by incorporating XAI. Their work focuses on the explanation of features by XAI, providing more transparency and understanding for users. Despite its promising advantages, the study lacked a comprehensive evaluation of security detection. Eckhart et al. [20] utilized DTs to evaluate and analyze a system's response to cyberattacks. The primary advantage of this study is the estimation of potential damages and the design of mitigation mechanisms, thus safeguarding systems from attacks without having to perform tests on running systems. Yigit et al. [23] proposed DT-enabled cybersecurity framework to detect Distributed Denial of Service (DDoS) attacks. The proposed system architecture consists of physical objects, digital twins, and online learning. The authors investigated various feature selection methods and selected the best work for the proposed model. The proposed model used an artificial neural network (multilayer perceptron) for classification. The proposed approach achieved better results, although it can only identify DDoS attacks.

2.2. Deep learning and Explainable Artificial Intelligence (XAI)

Deep learning has significantly enhanced the capabilities of modern IDS. Javeed et al. [8] focused on the use of artificial intelligence in IDS to extract important features and identify anomalies. Their approach led to AI-based anomaly detection, even though the models used are often considered "black-boxes". To overcome, the blackbox nature of a deep learning-based IDS, Wang et al. [11] proposed an explainable framework for intrusion detection. However, they evaluated their proposed framework using an out-of-date dataset. Abou et al. [13] promotes the benefits of XAI in IDS. By making the machine learning-based models interpretable, the internal workings and decisions of the models become more transparent, instilling confidence in users. Oseni et al. [14] proposed an intrusion detection framework for Internet of Things networks. Their proposed approach employed deep learning techniques for attack classifications. Furthermore, they used the SHAP method to explain the decisions of their deep learning-based IDS. Their proposed work was validated using the ToN_IoT dataset, and the experimental results showed that the detection model was effective against various types of cyberattacks.

Table 1
A comparison of existing solutions in Digital Twins-driven cybersecurity.

Related work	Digital Twins	Blockchain	Explainable AI-based IDS	ZTN	Advantage	Limitation
Varghese et al. [22]	✓	N/A	N/A	N/A	IDS with DTs integration	Limited attacks
Suhail et al. [25]	✓	N/A	✓	N/A	XAI feature explanation	Security evaluation
Thakur et al. [19]	✓	✓	N/A	N/A	Defense against threats	Communication cost
Lu et al. [26]	✓	✓	N/A	N/A	Privacy via blockchain	No accuracy
Ferrag et al. [27]	✓	N/A	N/A	N/A	Security/privacy	No comparison
Javeed et al. [8]	N/A	N/A	✓	N/A	AI-based anomaly detection	Black-box models
Abou et al. [13]	N/A	N/A	✓	N/A	XAI in IDS	Complex workings
Eckhart et al. [20]	✓	N/A	N/A	N/A	DTs for attack response	Testing on live systems
Kobayashi et al. [24]	✓	N/A	✓	N/A	DTs with XAI	Early research stage
Bitton et al. [28]	✓	N/A	N/A	N/A	DTs for security	Unspecified

2.3. Blockchain in cybersecurity

Blockchain technologies have been widely investigated in the cybersecurity domain as such technology ensures the integrity and authenticity of a transaction [17]. Therefore, any alteration to chaining blocks will be detected and rejected. Thakur et al. [19] combined the strengths of DTs and blockchain technology. Their approach is notable for its effectiveness against multiple security threats. However, they pointed out that there is a computational cost associated with their methods. Lu et al. [26] integrated blockchain with DTs, focusing on preserving privacy using blockchain and federated learning. Despite its potential for upholding privacy, the model's accuracy was not detailed in their study.

Table 1 provides a comparison of existing solutions in Digital Twin-driven cybersecurity. The examination of these existing solutions reveals a notable research gap in the comprehensive integration of Blockchain technology and Explainable AI (XAI) with Digital Twins (DTs) for enhanced cybersecurity. While several studies have individually explored the benefits of DTs for cybersecurity, the incorporation of Blockchain and XAI remains fragmented and underexplored. Specifically, the synergy between DTs and Blockchain for ensuring data integrity and secure transactions, alongside the deployment of XAI for transparent and understandable intrusion detection systems, is scarcely addressed. This gap indicates a significant opportunity for research in developing a unified cybersecurity framework that not only leverages the predictive and responsive capabilities of DTs but also incorporates Blockchain for security and trust, and XAI for clarity and accountability in decision-making processes. The existing literature's focus on isolated components underscores the potential for a holistic approach that could address the current limitations in trust, transparency, and explainability within the realm of Digital Twins-driven cybersecurity.

3. System model

In this section, we present the network and threat model that we have considered in designing the proposed framework. These models are explained briefly below:

3.1. Network model

The proposed framework is based on the network model illustrated in Fig. 1. In this model, we have considered a ZTN application that includes multiple layers for data sharing namely, the device layer, edge layer, cloud layer, and Digital Twins layer. Each layer has its own responsibility and task to perform and they are mentioned below:

- **Device layer:** In the ZTN environment, various connected and resource-constrained Smart Devices (SD) are used to monitor, collect, and transfer the environment readings.
- **Edge layer:** The edge servers are equipped with relatively higher computation devices for initial processing. The transaction or data from the device layer are relayed on a hop-by-hop forwarding basis to the edge layer.

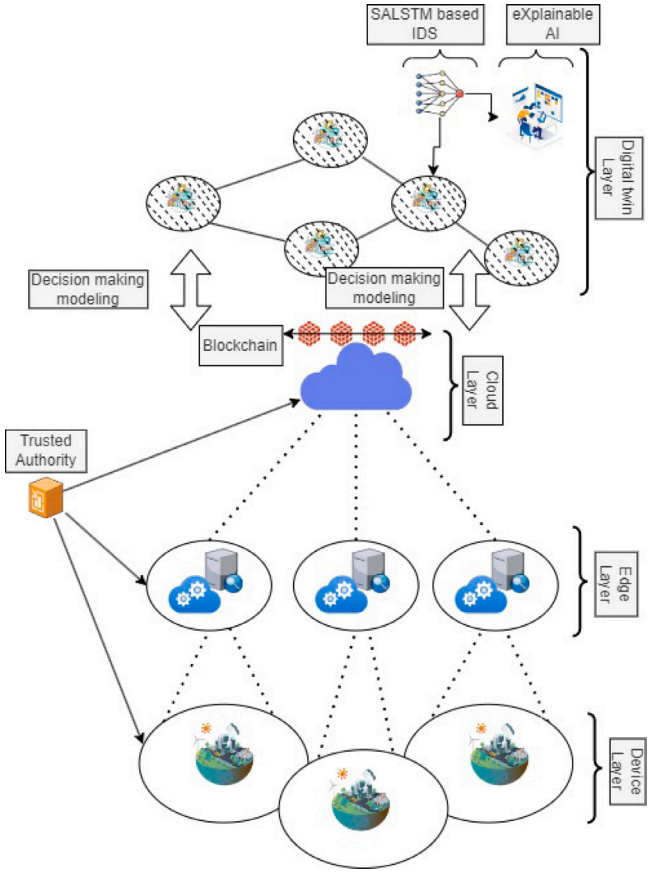


Fig. 1. Network model of proposed framework.

- **Cloud layer:** The cloud layer is used for long-term data storage and to perform various AI-based intelligent data analysis using the collected data for ZTN security and network management. This layer also uses a blockchain ledger to ensure secure decentralized data storage.
- **Digital Twins layer:** The data from the blockchain ledger is accessed by the Digital Twins (DTs) layer for generating and uploading various virtual models. In this paper, these virtual models are specifically various process-aware attack scenarios to train AI models for intrusion detection. Finally, an explainability element is added to the proposed attack detection technique in order to further explain the reasoning behind the higher accuracy in differentiating attacks from regular ZTN traffic patterns.

3.2. Threat model

Managing cybersecurity threats in the ZTN is a challenging task due to the integration of an increased number of devices, services, and technologies within the ZTN. In particular, advancement of IoT technology and mobile-based services, the connected devices within ZTN increased with potential attack surfaces that can pose any security threats [29]. It is necessary to understand threats within the network and take proactive security measures to safeguard the network, service, and data within ZTN. The proposed DT-enabled cybersecurity framework for ZTN considers four different layers of abstraction for formulating the network model. Threats that originated from the Device, Edge, Cloud, and Digital Twin layers need to be taken into consideration for threat modeling. Threat modeling provides an effective way to comprehensively understand and analyze the targeted attacks and identifies means to lower the impact of potential risks relating to the threats within ZTN. It allows one to identify weaknesses and shortcomings before a threat actor can exploit them. The potential attack surface of the ZTN is broad as vulnerabilities within the ZTN can be exploited from devices, services, and technologies. Therefore, threat modeling in ZTN needs to be carefully understood by each component within the layer-based abstraction of the proposed framework. The threats target ZTN focus on disruption of the network service delivery and capture data exchange within the network based on various types of attack including MITM, DoS, spoofing, tampering, and others. Additionally, Application Programming Interfaces(API) is one of the necessary components that interface between components and services within ZTN. Insecure APIs and parameters-based attacks can also be exploited that can pose unauthorized access, data manipulation, and DoS within the ZTN framework [30]. The adoption of ML and deep learning-based models in various parts of the ZTN such as IDS and intelligence network management can also introduce new attack vectors for potential poisoning attacks that tamper training data during the training phase or evasion attacks that bypass the learned model during the testing phase. There are also model inversion attacks that allow to capture of sensitive data using the model parameters [31]. Intent-based interfaces are the key component for the automation of ZTN which decouples from underlying technology and vendor-specific details and hides complexity from the user. There are several possible security threats such as Intercepting of application-related information from intent or malformed intent, which can violate the security of the ZTN. Therefore, it is challenging for traditional security measures to ensure security due to the growing number of connected devices, services, and the variety of APIs within the ZTN. The proposed framework adopts blockchain-based smart contracts and XAI-based IDS to enhance the overall security of ZTN.

4. Proposed framework

In this section, we have discussed the key components of the proposed cybersecurity framework.

4.1. Smart contract and blockchain-enabled authentication scheme

This subsection discusses mutual authentication, key agreement, and consensus methods based on smart contracts for secure, traceable, and transparent data storage.

4.1.1. System initialization phase

This section explains how network entities are initially set up. The initial configuration is carried out by the trusted authority *TPS*, who also distributes the necessary network settings for communications. The next section discusses the actions necessary for the network's initial setup.

The basic configuration of network entities is described in this section. The trustworthy entity *TPS* performs the initial configuration and distributes the required network configurations for communications. The steps required to set up the network initially are covered in the next section i.e.; registration and authentication. The description of the symbols is provided in Table 2.

Step-1: A non-singular elliptic curve is used by the *TPS* i.e., $E_n(k,m)$ which is in the form of $y^2 = x^3$+kx+m (mod e) over galois field *GF(e)*, where *e* signifies a large prime number over the condition $4k^3 + 27m^2 \neq 0$ (mod e) and also follows a non-singularity property over ω which includes either zero point and infinity point. Next, the *TPS* selects initial point $BSP \in E_n$(k,m) with closest order of *e*, say *n* i.e., $n \,.\, BSP = \omega$, where $n \,.\, BSP$ signifies the scalar multiplication point over the elliptic curve and $n \in Z_e$ i.e.; discrete algorithm over base point *BSP*.

Step-2: The *TPS* computes $hfn(.)$, a collision-resistant cryptographic hash function with one-way working principles. This is computed using the 256-bit message digest-providing secure hash technique (SHA-256) for security reasons.

Step-3: The *TPS* chooses the unique identity ID_{TPS} and perform computation of master key M_{TPS} and produces a private key randomly $PRK_{TPS} \in Z_e$, where $Z_e = \{1, 2, 3, 4, \ldots, e-1\}$. The *TPS* computes a public key i.e.; $PBK_{TPS} = PRK_{TPS} \,.\, BSP$.

Step-4: The *TPS* preserves a PRK_{TPS} and M_{TPS} preserves secret key and share the public parameter like { $E_n(k,m)$, *BSP*, PBK_{TPS}, $hfn(.)$ }.

4.1.2. Registration phase

This phase provides an illustration of the network entity registration process.

(i) ES Registration: The following processes, which are depicted below, are used by the *TPS* to register an edge server *ES*.

Step-1: The *TPS* picks an identity ID_{ES} and finds a pseudo identity SID_{ES}= hfn($ID_{TPS} \parallel M_{TPS} \parallel RT_{ES}$), where RT_{ES} signifies the timestamp of registration of *ES*. Further, *TPS* chooses a temporary identity $TIDM_{ES}$ and computes random secret value $PRK_{ES} \in Z_e$ and finds a public key $PBK_{ES} = PRK_{ES} \,.\, BSP$.

Step-2: The *TPS* generates a certificate of *ES* as $CRT_{ES} = M_{TPS}$ + hfn($PBK_{TPS} \parallel PBK_{ES} \parallel$) * PRK_{TPS} (mod g). Further, *TPS* stores the ES information i.e., ($TIDM_{ES}$, SID_{ES}, CRT_{ES}, PRK_{ES}, $E_n(k,m)$, *hfn(.)*) into the memory and distributes public key PBK_{ES} for further access.

(ii) SD Registration: The following processes, which are depicted below, are used by the *TPS* to register an edge server *SD*.

Step-1: The *TPS* picks an identity ID_{SD} and finds a pseudo identity SID_{SM}= hfn($ID_{TPS} \parallel M_{TPS} \parallel RT_{SD}$), where RT_{SD} signifies the timestamp of registration of smart device. Further, *TPS* chooses a temporary identity $TIDM_{SD}$ and computes random secret value $PRK_{SD} \in Z_e$ and finds a public key as $PBK_{SD} = PRK_{SD} \,.\, BSP$.

Step-2: The *TPS* generates a certificate of *SD* as $CRT_{SD} = M_{TPS}$ + hfn($PBK_{TPS} \parallel PBK_{SD} \parallel$) * PRK_{TPS} (mod g). Further, *TPS* stores the SM information i.e., ($TIDM_{SD}$, SID_{SD}, CRT_{SD}, PRK_{SD}, $E_n(k,m)$, *hfn(.)*) into the memory and distributes public key PBK_{SD} for further access.

*(iii) CS Registration:*The following processes, which are depicted below, are used by the *TPS* to register an edge server *CS*.

Step-1: The *TPS* picks an identity ID_{CS} and perform computation of pseudo identity SID_{CS}= hfn($ID_{TPS} \parallel M_{TPS} \parallel RT_{CS}$), where RT_{CS} signifies the timestamp of registration of cloud server. Further, *TPS* chooses a temporary identity $TIDM_{CS}$ and computes random secret value $PRK_{CS} \in Z_e$ and finds a public key $PBK_{CS} = PRK_{CS} \,.\, BSP$.

Step-2: The *TPS* generates a certificate of *CS* as $CRT_{CS} = M_{TPS}$ + hfn($PBK_{TPS} \parallel PBK_{CS} \parallel$) * PRK_{TPS} (mod g). Further, *TPS* stores the ES information i.e., ($TIDM_{CS}$, SID_{CS}, CRT_{CS}, PRK_{CS}, $E_n(k,m)$, *hfn(.)*) into the memory and distributes public key PBK_{CS} for further access.

Table 2
Symbol and description.

Symbol	Descriptions
TPS	Trusted authority
M_{TPS}, PRK_{TPS}, PRK_{TPS}	Master key, Private key, and Public key of Trusted authority
ID_{ES}, ID_{CS}, ID_{SD_i}	Identity of Edge server, Cloud server, and smart Device (SD)
SID_{ES}, SID_{CS}, SID_{SD_i}	Pseudo identity of Edge server, cloud Server, and SD
RT_{ES}, RT_{CS}, RT_{SD_i}	Registration Timestamp of Edge server, Cloud server, and SD
$TIDM_{ES}$, $TIDM_{CS}$, $TIDM_{SD_i}$	Temporary Identity of Edge server, Cloud server, and SD
CRT_{ES}, CRT_{CS}, CRT_{SD_i}	Certificate of Edge server, Cloud Server, and SD
PRK_{ES}, PRK_{CS}, PRK_{SD_i}	Private key of Edge server, Cloud server, and SD
PBK_{ES}, PBK_{CS}, PBK_{SD_i}	Public key of Edge server, Cloud server, and SD
$hfn(.)$, $E_n(k,m)$, BSP	Hash Function, Elliptical curve point (k,m), and curve base point
CTP_1, CTP_2, CTP_3	Current Timestamp of SD, Edge server, and cloud server
$SESV_{ES}$, $SESV_{CS}$, $SESV_{SD_i}$	Session key of Edge server, Cloud server, and SD

4.1.3. Key establishment and access control authentication phase

In this phase, the authentication procedures for smart devices (SD), Edge servers (ES), and cloud servers (CS) are discussed. Each entity in the authentication process keeps a session key establishment and access control, enabling secure interactions inside the framework. The authorization of each entity in the framework is developed through this method. The procedure for establishing key establishment and access control authentication and sharing session keys includes the stages below.

(i) Key Establishment and Access control Authentication between SD to SD

Step-1: SD_i randomly take a number $L_1 \in Z_e$ and also triggers a current timestamp CTP_1 and evaluates T_1= hfn($SID_{SD_i} \parallel TIDM_{SD_i} \parallel L_1 \parallel CTP_1$). Then SD_i makes encryption operation T_1 as T_2= $E_{PBK_{SD_j}}(T_1)$. Next, SD_i evaluates the parameter T_3= hfn($T_2 \parallel CRT_{SD_i} \parallel SID_{SD_i} \parallel TIDM_{SD_i} \parallel CTP_1$) and produces an request message M_1={SID_{SD_i}, $TIDM_{SD_i}$, CTP_1, T_2, T_3} and sends to another smart device (SD_j) by an open channels.

Step-2: After successful receiving of message M_1 on time CTP_1^*, the SD_j verifies the timestamp for checking delay of transmission using $|CTP_1^* - CTP_1| < \Delta T$. If the obtained timestamp is in the valid range, then SD_j checks for the certificate using $CRT_{SD_i}.BSP = PBK_{TPS}$ + hfn($PBK_{SD_i} \parallel PBK_{TPS}$), For every successful match, the SD_j fetches SID_{SD_i} corresponding to $TIDM_{SD_i}$ from the secure database storage and computes T_3^*= h ($T_2 \parallel SID_{SD_i} \parallel TIDM_{SD_i} \parallel CRT_{SD_i}$) to validate whether $T_3^* = T_3$. if valid, then SD_j applies decryption T_2 as T_1= $D_{PRK_{SD_j}}(T_2)$.

Step-3: Further, SD_j takes a number randomly $fr_1 \in Z_e$ and observes current timestamp CTP_2 and generates a new temporary identity $TIDM_{SD_i}^{new}$ and evaluates SD_j = hfn($SID_{SD_i} \parallel SID_{SD_j} \parallel fr_1 \parallel CTP_2$) and applies encryption SD_1 as SD_2= $E_{PBK_{SD_i}}(FGS_1)$. Next, SD_j generates a session key SES_{SD_j}= hfn($TIDM_{SD_i}^{new} \parallel T_1 \parallel FGS_1 \parallel CTP_1 \parallel CTP_2$), $TIDM_{SD_i}^* = TIDM_{SD_i}^{new} \oplus$ hfn($SID_{SD_j} \parallel TIDM_{SD_i} \parallel CTP_2$), and FG_3=hfn($TIDM_{SD_i}^* \parallel FGS_1 \parallel CRT_{SD_j} \parallel SID_{SD_j} \parallel CTP_2$) and makes a reply message M_2= {$TIDM_{SD_i}^*$, FG_2, CRT_{SD_j}, SID_{SD_j}, CTP_2} and shares with SD_i by an open channels.

Step-4: SD_j receives message (M_2) from SD_i on time CTP_2^*, then SD_i verify whether $|CTP_2^* - CTP_2| < \Delta T$ timestamp matches or not. if matches and it is within the range, then SD_i checks for the certificate i.e.; $CRT_{SD_j}.BSP = PBK_{TPS}$ + hfn($PBK_{SD_j} \parallel PBK_{TPS}$). Further, SD_i applies decryption FG_2 to get FGS_1= $D_{PRK_{SD_i}}(FG_2)$. Furthermore, SD_i evaluates FG_3^*= hfn($TIDM_{SD_i}^* \parallel FGS_1 \parallel CRT_{SD_j} \parallel SID_{SD_j} \parallel CTP_2$) and verify, if $FG_3^* = FG_3$ then SD_i it processes $TIDM_{SD_i}^{new} = TIDM_{SD_i}^* \oplus$ hfn($SID_{SD_j} \parallel TIDM_{SD_i} \parallel CTP_2$) and generates a session key SES_{SD_i}=hfn($TIDM_{SD_i}^{new} \parallel T_1 \parallel FGS_1 \parallel CTP_1 \parallel CTP_2$) and shares with SD_j. Next, SD_i picks a current timestamp CTP_3 and perform session key verification $SESV_{SD_i}$ using $SESV_{SD_i}$=hfn($SES_{SD_i} \parallel CTP_3$) and perform updation of $TIDM_{SD_i}$ and $TIDM_{SD_i}^{new}$ in the secure database storage. Further, SD_i generates an acknowledgment message M_3={$SESV_{SD_i}$, CTP_3} and shares with SD_j by open channels.

Step-5: After receiving an acknowledgment message M_3 on time CTP_3^*, then SD_j checks the current timestamp using $|CTP_3^* - CTP_3| < \Delta T$ to verify the validity of the timestamp. Next, SD_j verifies $SESV_{SD_i}$ = h ($SESV_{SD_j} \parallel CTP_3$). If it matches successfully, then SD_j shares the session key and performs mutual authentication $SESV_{SD_i}$ (= $SESV_{SD_j}$) with SD_i. Finally, SD_j makes updates to $TIDM_{SD_i}$ and $TIDM_{SD_i}^{new}$ in the secure database. The authentication between SD_i and SD_j is shown in Table 3.

(ii) Key Establishment and Access control Authentication between SD to ES

Step-1: SD_i takes a number randomly $L_1 \in Z_e$ and preserve a timestamp CTP_1 and perform computation of T_1= hfn($SID_{SD_i} \parallel TIDM_{SD_i} \parallel L_1 \parallel CTP_1$). Further, SD_i applies encryption T_1 as T_2= $E_{PBK_{ES}}(T_1)$. Furthermore, SD_i perform computation of T_3= hfn($T_2 \parallel CRT_{SD_i} \parallel SID_{SD_i} \parallel TIDM_{SD_i} \parallel CTP_1$) and sends a access request message M_1={SID_{SD_i}, $TIDM_{SD_i}$, CTP_1, T_2, T_3} and transmit to edger server (ES) by an open channels.

Step-2: After receive of message M_1 from SD_i on time CTP_1^*, then ES verifies current timestamp $|CTP_1^* - CTP_1| < \Delta T$. For successful timestamp it computes ES and checks for certificate using $CRT_{SD_i}.BSP = PBK_{TPS}$ + hfn($PBK_{SD_i} \parallel PBK_{TPS}$) if matches successfully, then ES keeps SID_{SD_i} of respective $TIDM_{SD_i}$ from the secure database storage and evaluates T_3^*= h ($T_2 \parallel SID_{SD_i} \parallel TIDM_{SD_i} \parallel CRT_{SD_i}$) to verify whether $T_3^* = T_3$. if matches successfully, then ES applies decryption T_2 as T_1= $D_{PRK_{ES}}(T_2)$.

Step-3: Further, ES takes a number randomly $fr_1 \in Z_e$ and preserve a timestamp CTP_2 and assign new temporary identity $TIDM_{SD_i}^{new}$ and evaluates ES = hfn($SID_{SD_i} \parallel SID_{ES} \parallel fr_1 \parallel CTP_2$) and applies encryption SD_1 as SD_2= $E_{PBK_{SD_i}}(FGS_1)$. Next, ES evaluates a session key SES_{ES}= hfn($TIDM_{SD_i}^{new} \parallel T_1 \parallel FGS_1 \parallel CTP_1 \parallel CTP_2$), $TIDM_{SD_i}^* = TIDM_{SD_i}^{new} \oplus$ hfn($SID_{ES} \parallel TIDM_{SD_i} \parallel CTP_2$), and FG_3=hfn($TIDM_{SD_i}^* \parallel FGS_1 \parallel CRT_{ES} \parallel SID_{ES} \parallel CTP_2$) and makes a reply message M_2= {$TIDM_{SD_i}^*$, FG_2, CRT_{ES}, SID_{ES}, CTP_2} and shares to SD_i by an open channels.

Step-4: After receive of reply message (M_2) from the ES on time CTP_2^*, then SD_i verify whether $|CTP_2^* - CTP_2| < \Delta T$ timestamp is valid or not. if validates successfully, then SD_i checks for certificate by $CRT_{ES}.BSP = PBK_{TPS}$ + hfn($PBK_{ES} \parallel PBK_{TPS}$). Further, SD_i applies decryption FG_2 to get FGS_1= $D_{PRK_{SD_i}}(FG_2)$. Furthermore, SD_i evaluates FG_3^*= hfn($TIDM_{SD_i}^* \parallel FGS_1 \parallel CRT_{ES} \parallel SID_{ES} \parallel CTP_2$) and verify, if $FG_3^* = FG_3$ then SD_i perform computation $TIDM_{SD_i}^{new} = TIDM_{SD_i}^* \oplus$ hfn($SID_{ES} \parallel TIDM_{SD_i} \parallel CTP_2$) and also makes evaluation of a session key SES_{SD_i}=hfn($TIDM_{SD_i}^{new} \parallel T_1 \parallel FGS_1 \parallel CTP_1 \parallel CTP_2$) and transmit to ES. Further, SD_i picks a current timestamp CTP_3 and perform computation of session key verification $SESV_{SD_i}$ using $SESV_{SD_i}$=hfn($SES_{SD_i} \parallel CTP_3$) and makes updation of the $TIDM_{SD_i}$ and $TIDM_{SD_i}^{new}$ in the secure database. Furthermore,

Table 3
Authentication process between SD to SD.

Smart Device (SD_i)	*Edge Server* (SD_j)
creates a unique random number $SDr_1 \in DS_r$ uses a current timestamp CTP_1 Evaluates LSD_1 = hfn ($SID_{SD_i} \| TIDM_{SD_i} \| SDr_1 \| CTP_1$) $LSD_2 = EN_{PBK_{SD_j}}(LSD_1)$ LSD_3 = hfn($LSD_2 \| CRT_{SD_i} \|$ $SID_{SD_i} \| TIDM_{SD_i} \| CTP_1$) $\xrightarrow[\text{(open channel)}]{MSG_1=\{SID_{SD_i}, TIDM_{SD_i}, CTP_1, LSD_2, LSD_3\}}$	
	Verify $\|CTP_1^* - CTP_1\| < \delta T$, if valid $CRT_{SD_i}.BSP = PBK_{TPS}$ + hfn($PBK_{SD_i} \| PBK_{TPS}$), if valid Fetch SID_{SD_i} with respect to $TIDM_{SD_i}$ from the database Computes LSD_3^*= hfn ($LSD_2 \| SID_{SD_i} \|$ $TIDM_{SD_i} \| CRT_{SD_i}$) Verify $LSD_3^* = LSD_3$, if validated successfully Decrypts $LSD_1 = D_{PRK_{SD_j}}(LSD_2)$ Picks a unique random number $CSr_1 \in DS_r$ and uses a current timestamp CTP_2 Computes CS_1= hfn($SID_{SD_i} \| SID_{SD_j} \|$ $CSr_1 \| CTP_2$) Encrypt CS_1 and store in $CS_2 = EN_{PBK_{SD_i}}(CS_1)$ produces a session key SES_{SD_j}= hfn($TIDM_{SD_i}^{new} \|$ $LSD_1 \| CS_1 \| CTP_1 \| CTP_2$), $TIDM_{SD_i}^* = TIDM_{SD_i}^{new} \oplus$ hfn($SID_{SD_j} \|$ $TIDM_{SD_i} \| CTP_2$), CSV_3= hfn($TIDM_{SD_i}^* \| CS_1 \|$ $CRT_{SD_j} \| SID_{SD_j} \| CTP_2$) $\xleftarrow[\text{(via open channel)}]{MSG_2=TIDM_{SD_i}^*, ES_2, CRT_{SD_j}, SID_{SD_j}, CTP_2}$
Verify $\| CTP_2^* - CTP_2 \| < \delta T$ Verify if, $CRT_{SD_j}.BSP = PBK_{TPS}$ + hfn($PBK_{SD_j} \| PBK_{TPS}$) Decrypts the CS_2 to receive $CS_1 = D_{PRK_{SD_i}}(CS_2)$ Computes CS_3^*= hfn($TIDM_{SD_i}^* \|$ $CS_1 \| CRT_{SD_j} \| SID_{SD_j} \| CTP_2$) if $CS_3^* = CS_3$ valid Computes $TIDM_{SD_i}^{new} = TIDM_{SD_i}^* \oplus$ hfn($SID_{SD_j} \| TIDM_{SD_i} \| CTP_2$) Evaluates session key SES_{SD_i}=hfn($TIDM_{SD_i}^{new} \|$ $LSD_1 \| CS_1 \| CTP_1 \| CTP_2$) Uses the current timestamp CTP_3 Performs verification of session key $SESV_{SD_i}$=hfn(SES_{SD_i} $\|CTP_3$) Change $TIDM_{SD_i}$ and $TIDM_{SD_i}^{new}$ in the database $\xrightarrow[\text{(via open channel)}]{MSG_3=SESV_{SD_i}, CTP_3}$	
	Verify $\| CTP_3^* - CTP_3 \| < \delta T$ Verify $SESV_{SD_i}$= hfn ($SESV_{SD_j} \| CTP_3$) if validated successfully Change $TIDM_{SD_i}$ and $TIDM_{SD_i}^{new}$ in the database. Verify both session keys SD_i and SD_j $SESV_{SD_i}$ ($=SESV_{SD_j}$)

SD_i produces an acknowledgment message M_3={$SESV_{SD_i}$, CTP_3} and shares to ES by an open channels.

Step-5: After receive of an acknowledgment message M_3 on time CTP_3^*, then ES check for the timestamp using $| CTP_3^* - CTP_3 | < \Delta T$ to check the whether timestamp is valid or not. Further, ES checks $SESV_{SD_i}$= h ($SESV_{ES} \| CTP_3$). If matches successfully, then ES shares the session key and establish a mutual authentication $SESV_{SD_i}$ ($=SESV_{ES}$) with SD_i. Finally, ES makes updation of $TIDM_{SD_i}$ and $TIDM_{SD_i}^{new}$ in the secure database. The authentication between SD to ES is shown in Table 4.

(iii) Key Establishment and Access control Authentication between ES to CS

Step-1: ES_i takes a number randomly $L_1 \in Z_e$ and preserve a timestamp CTP_1 and perform computation of T_1= hfn($SID_{ES_i} \| TIDM_{ES_i} \| L_1 \| CTP_1$). Further, ES_i applies encryption T_1 as $T_2 = E_{PBK_{CS}}(T_1)$. Furthermore, SD_i perform computation of T_3= hfn($T_2 \| CRT_{ES_i} \| SID_{ES_i} \| TIDM_{ES_i} \| CTP_1$) and sends a access request message M_1={SID_{ES_i}, $TIDM_{ES_i}$, CTP_1, T_2, T_3} and transmit to smart respective meter (CS) by an open channels.

Step-2: After receive of message M_1 on time CTP_1^*, then CS verifies current timestamp $|CTP_1^* - CTP_1| < \Delta T$. For successful timestamp, CS checks for certificate using $CRT_{ES_i}.BSP = PBK_{TPS}$ + hfn($PBK_{ES_i} \| PBK_{TPS}$). For successful verification of certificates CS keeps SID_{ES_i} of respective $TIDM_{ES_i}$ in the database and evaluates T_3^*= h ($T_2 \| SID_{ES_i} \| TIDM_{ES_i} \| CRT_{ES_i}$) to verify whether $T_3^* = T_3$. For successful matches, CS perform decryption T_2 as $T_1 = D_{PRK_{CS}}(T_2)$.

Step-3: Further, CS takes a number randomly $fr_1 \in Z_e$ and preserve timestamp CTP_2 and assign new temporary identity $TIDM_{ES_i}^{new}$ and

Table 4
Authentication process between SD and ES.

Smart Device (SD)	*Edge Server (ES)*
creates a unique random number $SDr_1 \in DS_r$ uses a current timestamp CTP_1 Evaluates LSD_1= hfn (SID_{SD} || $TIDM_{SD}$ || SDr_1 || CTP_1) LSD_2= $EN_{PBK_{ES}}(LSD_1)$ LSD_3= hfn(LSD_2 || CRT_{SD} || SID_{SD} || $TIDM_{SD}$ || CTP_1) $\xrightarrow[\text{(open channel)}]{MSG_1=\{SID_{SD},TIDM_{SD},CTP_1,LSD_2,LSD_3\}}$	
	Verify $\|CTP_1^*$- $CTP_1\| < \delta T$, if valid $CRT_{SD}.BSP$= PBK_{TPS} + hfn(PBK_{SD} || PBK_{TPS}), if valid Fetch SID_{SD} with respect to $TIDM_{SD}$ from the database Computes LSD_3^*= hfn (LSD_2 || SID_{SD} || $TIDM_{SD}$ || CRT_{SD}) Verify LSD_3^*= LSD_3, if validated successfully Decrypts LSD_1= $D_{PRK_{ES}}(LSD_2)$ Picks a unique random number $CSr_1 \in DS_r$ and uses a current timestamp CTP_2 Computes CS_1= hfn(SID_{SD} || SID_{ES} || CSr_1 || CTP_2) Encrypt CS_1 and stored in CS_2= $EN_{PBK_{SD}}(CS_1)$ produces a session key SES_{ES}= hfn($TIDM_{SD}^{new}$ || LSD_1 || CS_1 || CTP_1 || CTP_2), $TIDM_{SD}^*$= $TIDM_{SD}^{new}$ ⊕ hfn(SID_{ES} || $TIDM_{SD}$ ||CTP_2), CSV_3= hfn($TIDM_{SD}^*$ || CS_1 || CRT_{ES} || SID_{ES} || CTP_2) $\xleftarrow[\text{(via open channel)}]{MSG_2=TIDM_{SD}^*,ES_2,CRT_{ES},SID_{ES},CTP_2}$
Verify | CTP_2^* - CTP_2 | $< \delta T$ Verify if, $CRT_{ES}.BSP$=PBK_{TPS} + hfn(PBK_{ES} || PBK_{TPS}) Decrypts the CS_2 to receive CS_1= $D_{PRK_{SD}}(CS_2)$ Computes CS_3^*= hfn($TIDM_{SD}^*$ || CS_1 || CRT_{ES} || SID_{ES} || CTP_2) if CS_3^*= CS_3 valid Computes $TIDM_{SD}^{new}$= $TIDM_{SD}^*$ ⊕ hfn(SID_{ES} || $TIDM_{SD}$ || CTP_2) Evaluates session key SES_{SD}=hfn($TIDM_{SD}^{new}$ || LSD_1 || CS_1 || CTP_1 || CTP_2) Uses the current timestamp CTP_3 Performs verification of session key $SESV_{SD}$=hfn(SES_{SD} ||CTP_3) Change $TIDM_{SD}$ and $TIDM_{SD}^{new}$ in the database $\xrightarrow[\text{(via open channel)}]{MSG_3=SESV_{SD},CTP_3}$	
	Verify | CTP_3^* - CTP_3 | $< \delta T$ Verify $SESV_{SD}$= hfn ($SESV_{ES}$ || CTP_3) if validated successfully Change $TIDM_{SD}$ and $TIDM_{SD}^{new}$ in the database. Verify both session key ES and ES $SESV_{SD}$ (=$SESV_{ES}$)

evaluates CS = hfn(SID_{ES_i} ‖ SID_{CS} ‖ fr_1 ‖ CTP_2) and applies encryption SD_1 as SD_2= $E_{PBK_{ES_i}}(FGS_1)$. Next, CS evaluates a session key SES_{CS}= hfn($TIDM_{ES_i}^{new}$ ‖ T_1 ‖ FGS_1 ‖ CTP_1 ‖ CTP_2), $TIDM_{ES_i}^*$= $TIDM_{ES_i}^{new}$ ⊕ hfn(SID_{CS} ‖ $TIDM_{ES_i}$ ‖CTP_2), and FG_3=hfn($TIDM_{ES_i}^*$ ‖ FGS_1 ‖ CRT_{CS} ‖ SID_{CS} ‖ CTP_2) and makes a reply message M_2= {$TIDM_{ES_i}^*$, FG_2, CRT_{CS}, SID_{CS}, CTP_2} and shares to SD_i by an open channels.

Step-4: After receive of reply message (}M_2) from the CS on time CTP_2^*, then SD_i verify whether | CTP_2^* - CTP_2 | < ΔT timestamp is valid or not. if validates successfully, then SD_i checks for certificate by $CRT_{CS}.BSP$=PBK_{TPS} + hfn(PBK_{CS} ‖ PBK_{TPS}). Further, SD_i applies decryption FG_2 to get FGS_1= $D_{PRK_{ES_i}}(FG_2)$. Furthermore, SD_i evaluates FG_3^*= hfn($TIDM_{ES_i}^*$ ‖ FGS_1 ‖ CRT_{CS} ‖ SID_{CS} ‖ CTP_2) and verify, if FG_3^*= FG_3 then SD_i perform computation $TIDM_{ES_i}^{new}$= $TIDM_{ES_i}^*$ ⊕ hfn(SID_{CS} ‖ $TIDM_{ES_i}$ ‖ CTP_2) and also makes evaluation of a session key SES_{ES_i}=hfn($TIDM_{ES_i}^{new}$ ‖ T_1 ‖ FGS_1 ‖ CTP_1 ‖ CTP_2) and transmit to CS. Further, SD_i picks a current timestamp CTP_3 and perform computation of session key verification $SESV_{ES_i}$ using $SESV_{ES_i}$=hfn(SES_{ES_i} ‖CTP_3) and makes updation of the $TIDM_{ES_i}$ and $TIDM_{ES_i}^{new}$ in the secure database. Furthermore, SD_i produces an acknowledgment message M_3={$SESV_{ES_i}$, CTP_3} and shares to CS by an open channels.

Step-5: After receive of an acknowledgment message M_3 on time CTP_3^*, the CS check for the timestamp using | CTP_3^* - CTP_3 | < ΔT to check the whether timestamp is valid or not. Further, CS checks $SESV_{ES_i}$= h($SESV_{CS}$ ‖ CTP_3). If matches successfully, then CS shares the session key and establish a mutual authentication $SESV_{ES_i}$ (=$SESV_{CS}$) with ES_i. Finally, CS makes updation of $TIDM_{ES_i}$ and $TIDM_{ES_i}^{new}$ in the secure database. The authentication between ES to CS is shown in the Table 5.

4.2. Consensus mechanism for block creation and validation

This phase summarizes the consensus mechanism of verification and block creation. The verification and block creation are performed by

Table 5
Authentication process between ES and CS.

Edge Server (ES)	*Cloud Server (CS)*
creates a unique random number $ESr_1 \in DS_r$ uses a current timestamp CTP_1 Evaluates LES_1= hfn (SID_{ES} || $TIDM_{ES}$ || ESr_1 || CTP_1) LES_2= $EN_{PBK_{CS}}(LES_1)$ LES_3= hfn(LES_2 || CRT_{ES} || SID_{ES} || $TIDM_{ES}$ || CTP_1) $\xrightarrow[\text{(open channel)}]{MSG_1=\{SID_{ES},TIDM_{ES},CTP_1,LES_2,LES_3\}}$	
	Verify $\|CTP_1^*$- $CTP_1\| < \delta T$, if valid $CRT_{ES}.BSP$= PBK_{TPS} + hfn(PBK_{ES} || PBK_{TPS}), if valid Fetch SID_{ES} with respect to $TIDM_{ES}$ from the database Computes LES_3^*= hfn (LES_2 || SID_{ES} || $TIDM_{ES}$ || CRT_{ES}) Verify LES_3^*= LES_3, if validated successfully Decrypts LES_1= $D_{PRK_{CS}}(LES_2)$ Picks a unique random number $CSr_1 \in DS_r$ and uses a current timestamp CTP_2 Computes CS_1= hfn(SID_{ES} || SID_{CS} || CSr_1 || CTP_2) Encrypt CS_1 and stored in CS_2= $EN_{PBK_{ES}}(CS_1)$ produces a session key SES_{CS}= hfn($TIDM_{ES}^{new}$ || LES_1 || CS_1 || CTP_1 || CTP_2), $TIDM_{ES}^*$= $TIDM_{ES}^{new}$ $\oplus$ hfn(SID_{CS} || $TIDM_{ES}$ ||CTP_2), CSV_3= hfn($TIDM_{ES}^*$ || CS_1 || CRT_{CS} || SID_{CS} || CTP_2) $\xleftarrow[\text{(via open channel)}]{MSG_2=TIDM_{ES}^*,CS_2,CRT_{CS},SID_{CS},CTP_2}$
Verify | CTP_2^* - CTP_2 | $< \delta T$ Verify if, $CRT_{CS}.BSP$=PBK_{TPS} + hfn(PBK_{CS} || PBK_{TPS}) Decrypts the CS_2 to receive CS_1= $D_{PRK_{ES}}(CS_2)$ Computes CS_3^*= hfn($TIDM_{ES}^*$ || CS_1 || CRT_{CS} || SID_{CS} || CTP_2) if CS_3^*= CS_3 valid Computes $TIDM_{ES}^{new}$= $TIDM_{ES}^*$ $\oplus$ hfn(SID_{CS} || $TIDM_{ES}$ || CTP_2) Evaluates session key SES_{ES}=hfn($TIDM_{ES}^{new}$ || LES_1 || CS_1 || CTP_1 || CTP_2) Uses current timestamp CTP_3 Performs verification of session key $SESV_{ES}$=hfn(SES_{ES} ||CTP_3) Change $TIDM_{ES}$ and $TIDM_{ES}^{new}$ in the database $\xrightarrow[\text{(via open channel)}]{MSG_3=SESV_{ES},CTP_3}$	
	Verify | CTP_3^* - CTP_3 | $< \delta T$ Verify $SESV_{ES}$= hfn ($SESV_{CS}$ || CTP_3) if validated successfully Change $TIDM_{ES}$ and $TIDM_{ES}^{new}$ in th database. Verify both session key ES and CS $SESV_{ES}$ (=$SESV_{CS}$)

the $\mathcal{CS}^i$. To verify and creation of block, a PoA consensus mechanism is applied. The Algorithm 1 consists of two different methods namely BlockSign() and CreateBlock(). The BlockSign() module is responsible for verifying and performing block signatures by the mining nodes i.e.; $\mathcal{CS}^i$. The underlying methods check for the current timestamp $\mathcal{CTP}$ and also check for maximum mining nodes are participating in the block verification process. Next, for successful verification block weight is assigned to 1, and for every unsuccessful verification weight is assigned to 0. Further, the CreateBlock() module takes the block weight, block-sign, and mining node information. For successful authentication of miners and related block information block creation is performed and shared with the peer mining node to append the block to the current ledger. The algorithm consists of two functions namely Blocksign() and reateBlock(). In the presented algorithm we have included CS_i as a mining node where *ith* represent individual mining node. However, $\{i = 1, 2, \ldots n\}$. The mining node sign the block which is added to next of the *nth* block. If the block is sign successfully by the mining node then block weight gets value as BZ.W=1 Else BZ.W=0. However, each mining node has two option either choose the block or not choose the block. So, as per the mining option complexity of mining and generating block hash of a block is O(2^n). Finally, Each blocks are added as a chronological order and thus it must traverse the entire lists of block which takes O(n) as a time complexity. The total time complexity of the algorithm is O(2^n) + O(n).

4.3. Explainable deep learning-based intelligent Intrusion Detection System

The proposed IDS includes three main stages: (1) Data collection and Pre-processing, (2) Self Attention-based LSTM, and (3) eXplainable AI. The proposed IDS working is shown in Fig. 2. Each of them is explained below in detail:

4.3.1. Data collection and pre-processing

This process includes two parts mentioned below:

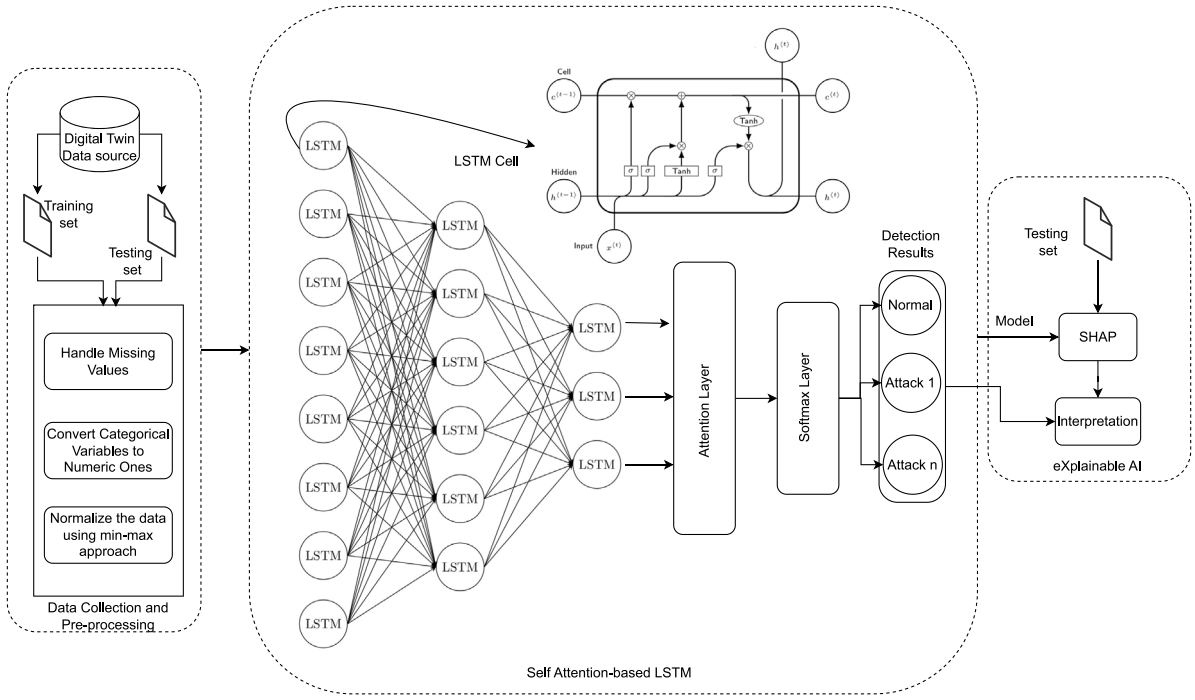


Fig. 2. Proposed explainable deep learning-based intelligent intrusion detection system.

Algorithm 1 Block Verification and Creation using Proof-of-Authority Consensus Approach

Input: Set of $\mathcal{CS}_i$, $\mathcal{BZ}$ ← Block Details, $\mathcal{PRB}$ ← parent block, CS_{ID_i} ← who made block signature, $\mathcal{BSI}$ ← Block Index Number, $\mathcal{W}$ ← Weight of Signed Block, $\mathcal{BD}$ ← Block Delay Time Commitment, $\mathcal{PRM}$ ← Majority of Peers($(\mathcal{CS}_i/2)+1$).
Output: *Block_Verification* and *Propagation*
function BlockSign()($\mathcal{CS}_i$, $\mathcal{CTP}$) :
  β ← Maximum $\mathcal{CS}_i$
  Result= False
  for do $\mathcal{CS}$ = {$\mathcal{CTP}$-β, … , $\mathcal{CTP}$}
    if then $\mathcal{BZ}.\mathcal{BSI}$ mod $|\mathcal{CS}_i| == i^{th}$ then Result= True
      return Result
    end if
  end for
end function
function CreateBlock(:)
  while true **do**
    CTP ← previous-block(CS_i).BSI
    wait until *function BlockSign*($\mathcal{CS}_i$, $\mathcal{N}$)
    CTP ← receive-timestamp($\mathcal{CS}_i$)
    wait until clock >= CTP + $\mathcal{BZ}$
    if (**then** $\mathcal{CTP}+1$) mod ($\mathcal{CS}_i$) == ith then
      BZ.W=1
    else
      BZ.W=0
    end if
    BZ.BI=$\mathcal{CTP}+1$
    BZ.PRB= ParentBlock($\mathcal{BZ}_i$)
    BZ.$\mathcal{CS}_i$ ← BlockSign()
    $\mathcal{BZ}_i$ ← { $\mathcal{BZ}_i$ U {BZ},$\mathcal{PRB}_i U\{BZ.PRB\}$ }
    Distribute Block $\mathcal{BZ}_i$
  end while
end function

- Collecting data from the digital twin: This involves capturing the virtual behaviors and interactions within the simulated environment. Considerations for intrusion detection using digital twin data include: (1) Monitor the behavioral patterns of digital entities within the virtual environment. This includes tracking interactions between components, data flows, and communication protocols. (2) Intentionally introduce anomalies and security threats into the digital twin to simulate potential intrusion scenarios. This enables the training of intrusion detection algorithms on a diverse set of scenarios. (3) Emulate virtual sensors within the digital twin to capture simulated data reflecting potential security breaches. These sensors mimic network traffic, system logs, and user activities in ZTN. (4) Implement detailed event logging within the digital twin to record all activities and changes. This log data serves as the basis for intrusion detection analysis.
- Pre-processing of collected data: (1) Identifying and addressing missing values is essential for maintaining the integrity and accuracy of the dataset. In this research article, we have used the mean imputation approach. This involves identifying features with missing data, calculating the mean for each of these features based on available values, and replacing the missing entries with their respective means. (2) categorical variables were converted into numeric format. (3) Normalization using min–max scaling, where numerical features were converted to a specific range, typically between 0 and 1.

4.3.2. Self attention-based LSTM

The pre-processed data is being used to design self self-attention-based LSTM model. The RNN has been widely used in the last few years in a variety of areas, such as translation, speech recognition, language modeling, etc. It was specifically designed for the time-series data. Unlike traditional CNN-based models, i.e., VGG and AlexNet, the RNN have the ability to capture the temporal information in sequence. However, it has gradient vanishing problems. LSTM is considered to be an advanced version of the traditional RNN [32], which is specially designed to solve the gradient vanishing problems by using its gating mechanism. The LSTM contains 3 gates; an Input gate ($\mathcal{N}_t$), Output gate ($\mathcal{M}_t$), and a Forget gate ($\mathcal{P}_t$). The last $\mathcal{P}_t$ is used to discard and choose to eliminate the extraneous data from the input ($\mathcal{S}_t$) and the preceding output ($\mathcal{J}_{t-1}$). The mathematical equations for computing these gates are as follows [33]:

$$\mathcal{N}_t = \sigma(w_n \mathcal{S}_t + w_n \mathcal{J}_{t-1} + b_n) \quad (1)$$

$$\mathcal{P}_t = \sigma(w_p \mathcal{S}_t + w_p \mathcal{J}_{t-1} + b_p) \tag{2}$$

$$\mathcal{M}_t = \sigma(w_m \mathcal{S}_t + w_m \mathcal{J}_{t-1} + b_m) \tag{3}$$

$$\mathcal{C}_t = \mathcal{C}_{t-1} * \mathcal{P}_t + \tanh(w_c \mathcal{S}_t + w_c \mathcal{J}_{t-1} + b_c) \tag{4}$$

$$\mathcal{Z}_t = \tanh(w_z \mathcal{S}_t + w_z \mathcal{J}_{t-1} + b_z) \tag{5}$$

Where σ is sigmoid function, $\mathcal{S}_t$ is the input vector, $\mathcal{W}_n$, $\mathcal{W}_p$, $\mathcal{W}_m$, $\mathcal{W}_c$ and $\mathcal{W}_z$ are the weight matrices for $\mathcal{N}_t$, $\mathcal{P}_t$, $\mathcal{M}_t$, $\mathcal{C}_t$ and $\mathcal{Z}_t$. Furthermore the Bias (b) for the weight matrices (w) are represented by b_n, b_p, b_m, b_c and b_z.

Moreover, we have employed an attention mechanism to learn and select the relevant information from the $\mathcal{N}$ hidden states ($\mathcal{J}_t$) using the weights (γ). Eq. (6) depicts the calculation of the Self-attention Vector ($\mathcal{SAV}$).

$$\mathcal{SAV} = \sum_{e=1}^{\mathcal{N}} \gamma_t \mathcal{J}_t \tag{6}$$

where the weighting factor of the γ_t is determined as

$$\gamma_t = \frac{exp(\mathcal{L}_t^T)\mathcal{L}_w}{\sum_t exp(\mathcal{L}_t^T)\mathcal{L}_w} \tag{7}$$

The $\mathcal{L}_w$ represents the weight matrix respectively.

4.3.3. eXplainable AI

The recent advances in the communication sector have yielded phenomenal concepts for developing a rational understanding between technologies and consumers. Artificial intelligence (AI) is revealing miraculous manifestations in every dimension of the digital world and has become an accelerating choice for smart communication systems such as ZTNs. The flourishing trends of AI applications in ZTN environments stimulate the need for some explainable approaches to ensure transparency in methodological frameworks. Explainable AI (XAI) is mapped over the elaboration concepts to visualize the typical processing of conventional integration frameworks.

In this article, we have used SHapley Additive exPlanations (SHAP) for interpreting the importance of features behind the attack detection capabilities [34]. The SHAP explanation method from coalitional game theory is used to derive Shapley values. First, calculate the SHAP values for each feature i in the set by evaluating the difference in predictions when including and excluding the feature. The SHAP value for feature i on input x is given by:

$$\phi_i(x) = \sum_{S \subseteq N \setminus \{i\}} \left[\frac{|S|!(|N| - |S| - 1)!}{|N|!} \right] \times \left[f_\Theta(S \cup \{i\}, x) - f_\Theta(S, x) \right]. \tag{8}$$

where N is the set of all feature in the set, S is a subset of feature excluding feature i, $f_\Theta(S, x)$ is the model's prediction with feature in S, $f_\Theta(S \cup \{i\}, x)$ is the model's prediction with feature in S plus feature i. For global interpretation, we calculate global SHAP values for each feature i by aggregating the contributions across the entire dataset: $\Phi_i = \frac{1}{m} \sum_{k=1}^{m} \phi_i(x_k)$, where Φ_i represents the global SHAP value for feature i and m is the number of data points in the dataset. Finally, we analyze the global SHAP values Φ_i to understand the relative importance of each feature i across the entire dataset. Larger absolute values of Φ_i indicate more influential features.

5. Performance analysis

5.1. Experimental setup

The deep learning algorithms are implemented using Tensorflow version 2.5 and Python programming language. The concept of explainable AI is implemented using the SHapley Additive exPlanations (SHAP) library version 0.39.0. This approach uses DeepExplainer for SALSTM model and plots the SHAP summary plot with mean SHAP values on the x-axis, and feature names on the y-axis. The simulations are carried out on PowerEdge R940xa Rack Server having 2x Intel® Xeon® Gold 6240 2.6G, 256 GB RAM, and NVIDIA Ampere A100, 80 GB Passive GPU installed on Microsoft Server 2019 Standard. The proposed XAI-SALSTM is trained with 3 layers having 128, 64, and 32 neurons, a "Relu" optimizer with a learning rate of 0.001 over 15 epochs, and "categorical cross-entropy" loss. The softmax activation function is used at the last layer to perform multi-class attack detection. The blockchain-based experimental analysis is performed over the Ethereum Goerli Test Network-enabled PoA consensus mechanism. The smart contract is implemented using solidity version 0.8.21. The Goreli etherscan blockchain explorer is used to store the transactions. To access the ethersacn services and smart contract functionality Web3 python is used. To setup the Ethereum node and receive goreli faucet Alchamey Web3 provider services are used.

Table 6
Statistics of instances of the datasets.

Dataset	Category	Instances
N-BaIoT	Benign	49 433
	Ack	3370
	Scan	3277
	SYN	2260
	UDP	3377
	UDP Plain	3383
	Combo	3307
	Junk	3367
	TCP	3390
DTs	Benign	1930
	NDoS	313
	NMM	27
	CI	163
	CMM	97

5.2. Dataset description

This work employed IoT-based datasets, i.e., N-BaIoT [35] and a self-generated DTs dataset by [22] for experimentation. The datasets contain a Benign class along with numerous attack classes, such that Ack, Scan, SYN, UDP, TCP, Junk, Combo, UDP Plain, Calculated Measurement Modification (CMM), Naive Measurement Modification (NMM), Command Injection (CI), and Network DoS (NDoS). The complete details about the instances if these datasets are given in 6. Further, we divided the dataset into the traditional 70 and 30 percent ratios. Finally, we employed the pre-processing based on [36].

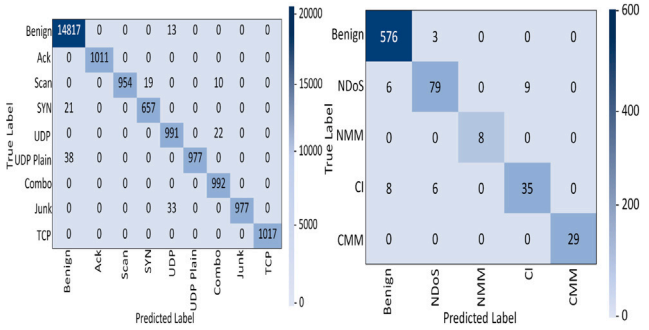
5.3. Evaluation metrics

In this work, we employed standard evaluation metrics to thoroughly evaluate the performance of our proposed framework, i.e., accuracy, precision, recall, and F1-score. Furthermore, we used the Shap summary plot of the XAI to interpret the features that contributed the most to the model's decision.

5.4. Numerical results for explainable DL-based IDS

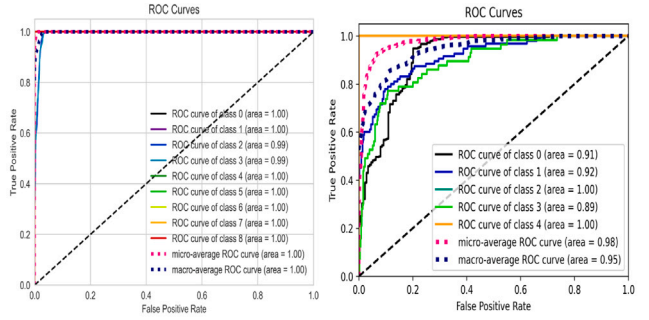
In this subsection, we discussed the performance of the proposed eXplainable DL-based IDS. Figs. 3(a) and 3(b) depict the confusion matrix of the proposed IDS under N-BaIoT and DTs datasets. The figures are evidence that the proposed IDS identified all the classes correctly and categorized the instances into their specific class.

Further, The ROC Curve of the proposed IDS is given in Figs. 4(a) and 4(b) for both datasets, where it depicts the performance between the true positive rate and false positive. The proposed IDS learns all



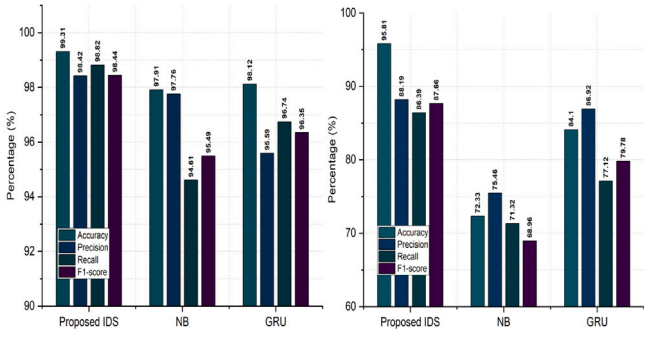
(a) Under N-BaIoT Dataset (b) Under DTs Dataset

Fig. 3. Confusion matrix.



(a) Under N-BaIoT Dataset (b) Under DTs Dataset

Fig. 4. ROC curve.



(a) N-BaIoT Dataset (b) DTs Dataset

Fig. 5. Overall performance comparison of the proposed IDS with existing threat detection schemes (Naive Bayes (NB) and Gated Recurrent Unit (GRU)).

the classes in an efficient manner. We designed the IDS for multi-class detection, and it has the ability to identify each unique threat independently, which is useful for the real-time deployment of the protection mechanism. Despite the fact that multi-class detection is a tricky, difficult, and very challenging task to attain high detection accuracy, the proposed IDS was able to attain an efficient accuracy. The overall comparison is provided in Fig. 5(a) under the N-BaIoT dataset, where the model accomplished accuracy, precision, recall, and F1-score of 99.31%, 98.42%, 98.82%, and 98.44%, respectively. Further, the comparison under the DTs dataset is provided in Fig. 5(b), where the proposed IDS achieved accuracy, precision, recall, and F1-score of 95.81%, 88.19%, 86.39%, and 87.66%. Also, in order to provide a more comprehensive performance of the proposed IDS, we provide the per-class performance analysis in Tables 7 and 8 for the N-BaIoT and DTs datasets respectively.

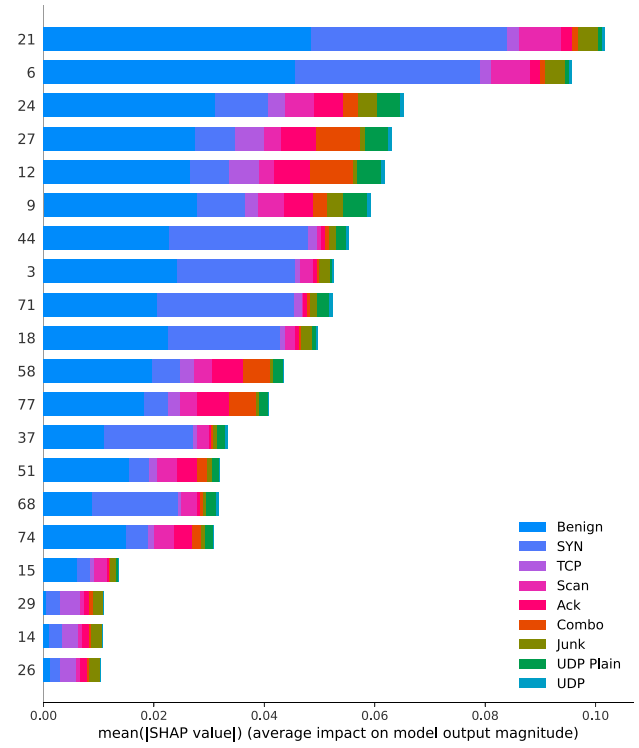


Fig. 6. Shap summary plot under N-BaIoT dataset.

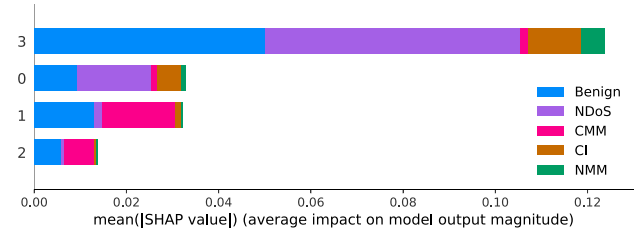


Fig. 7. Shap summary plot under DTs dataset.

Furthermore, we provide the Summary plot to show each feature's contribution in the model decision. Fig. 6 depicts the top 20 features of each class of the N-BaIoT dataset, that contributed the most to the proposed model output, while Fig. 7 illustrates the features of the DTs dataset. The different colors in these figures depict the magnitude of the features and the x-axis shows their influence on the classification of each distinct class. These features and associated Shapley values allow for the evaluation of the veracity of the projected result.

Further, to assess the statistical significance of the performance differences between models on the N-BaIoT and DTs dataset, we utilized the Wilcoxon rank-sum test, with a significance level set at $\alpha = 0.05$. The results of this test are summarized in Table 9. As can be discerned from the table, all the p-values are notably below the 0.05 threshold. This suggests that the differences in performance between the models are statistically significant across both datasets.

5.5. Numerical results for smart contract and blockchain-enabled authentication scheme

Fig. 8 shows execution time analysis of transaction (Tx) off-chain upload and block mining using the PoA consensus mechanism. Fig. 9 illustrates the block creation time and storage size of transactions over the off-chain layer. Figs. 8(b) and 9(a), illustrate block mining and block creation time over different peers and transactions. The

Table 7
Per-class performance analysis of the proposed IDS on N-BaIoT dataset.

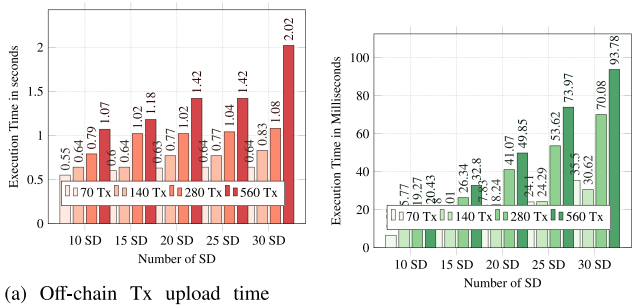
Parameters	Benign	Ack	Scan	SYN	UDP	UDP Plain	Combo	Junk	TCP
Accuracy	99.94	100.00	97.06	97.04	99.97	99.99	99.99	99.99	99.99
Precision	99.93	100.00	89.21	97.34	99.39	100.00	100.00	100.00	99.90
Recall	99.97	100.00	98.86	92.26	100.00	99.69	99.89	99.69	100.00
False Positive Rate	0.00306	0.00	0.03010	0.00045	0.00027	0.00	0.00	0.00	0.00004

Table 8
Per-class performance analysis of the proposed IDS on DTs dataset.

Parameters	Benign	NDoS	NMM	CI	CMM
Accuracy	91.74	92.98	100.00	94.33	100.00
Precision	90.72	85.86	92.30	72.09	100.00
Recall	97.48	72.20	100.00	62.29	100.00
False Positive Rate	0.27467	0.01601	0.00	0.01349	0.00

Table 9
P-values of Wilcoxon test for accuracy results on N-BaIoT and DTs dataset.

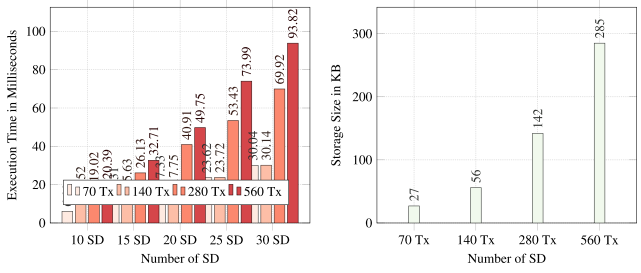
Comparison	N-BaIoT	DTs dataset
Proposed IDS vs NB	0.001953	0.002543
Proposed IDS vs GRU	0.002853	0.006606
NB vs GRU	0.003267	0.003658



(a) Off-chain Tx upload time Analysis

(b) Block Mining Time

Fig. 8. Analysis of blockchain authentication in terms of block mining using PoA consensus and Transaction (Tx) upload Time.



(a) Analysis of Block Creation Time

(b) Analyis of off-chain storage size of transaction (Tx) in KB

Fig. 9. Analysis of blockchain authentication in terms of block creation and storage size of transaction (Tx)

execution time analysis gradually increments as more peers participate in the network for the verification and mining process. Fig. 9(b) shows the size of shared transactions over off-chain storage leveraged with different sets of transactions and participating peers. It can be noticed that storage size increases as participating nodes and transactions are added to the network.

Table 10
Comparison with existing techniques.

Ref	Model	Dataset	DTs	BC	XAI	Accuracy
[39]	CNN+RNN	N-BaIoT	×	×	×	86.47%
[22]	Stacked	DTs	✓	×	×	92.70%
[37]	SOM-DAGMM	N-BaIoT	×	×	×	96.00%
[38]	SVM	N-BaIoT	×	×	×	95.90%
[10]	Transformer	N-BaIoT	×	×	×	96.33%
Proposed	XAI-SALSTM	N-BaIoT DTs	✓	✓	✓	99.31% 95.81%

Terms & Abbreviations: DTs: Digital Twins; BC: Blockchain; XAI: Explainable Artificial Intelligence; SOM-DAGMM: Self-organizing Map-Deep Autoencoder Gaussian Mixture Model; SVM: Support Vector Machine; CNN: Convolutional Neural Networks; RNN: Recurrent Neural Network.

Table 11
Comparison between proposed work and other related frameworks.

Authors	Year	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	$\mathcal{D}$	$\mathcal{E}$	$\mathcal{F}$	$\mathcal{G}$	$\mathcal{H}$	$\mathcal{I}$	$\mathcal{J}$
Varghese et al. [22]	2022	✓	×	✓	×	×	×	×	×	×	×
Suhail et al. [25]	2023	✓	×	✓	✓	×	×	✓	×	✓	✓
Javeed et al. [8]	2023	✓	×	✓	×	×	×	×	×	×	×
Wang et al. [11]	2020	✓	×	✓	✓	×	×	✓	×	✓	×
Houda et al. [13]	2022	✓	×	✓	✓	×	×	✓	×	✓	×
Thakur et al. [19]	2023	✓	✓	×	×	×	✓	×	×	✓	✓
Lu et al. [26]	2020	✓	✓	×	×	×	×	×	×	✓	✓
Proposed Framework	2023	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

$\mathcal{A}$: Security; $\mathcal{B}$: Privacy; $\mathcal{C}$: Intrusion Detection System; $\mathcal{D}$: eXplainable AI; $\mathcal{E}$: Ledger Distribution; $\mathcal{F}$: Smart Contracts; $\mathcal{G}$: Transparency; $\mathcal{H}$: Decentralized; $\mathcal{I}$: Trust; $\mathcal{J}$: Digital Twins.

5.6. Comparative analysis

Finally, we compare the performance of the proposed framework with some recent threat detection techniques from the current literature, i.e., [37,38], and [22]. Table 10 depicts the thorough comparison of the proposed approach against these techniques. It can be seen that the authors of [37] evaluated their proposed framework using the N-BaIoT dataset and achieved 96% accuracy. Similarly, the authors in [38] also used the N-BaIoT dataset for training and evaluation and achieved an accuracy of 95.90%. Further, in [22] the authors employed a self-generated DTs dataset to train and evaluate their proposed IDS and attained 92.70% detection accuracy. The comparison further proves that the proposed framework outperformed the other schemes by achieving an accuracy of 99.31% under the N-BaIoT dataset and 95.81% under the DTs dataset.

Finally, we have compared the proposed framework with some recent state-of-the-art based on various factors. Table 11 shows this comparison of the proposed work with [8,11,13,19,22,25,26]. For instance, In the context of ZTN management, Security ($\mathcal{A}$) is paramount as it guards the system from cyber threats, ensuring that the automation in ZTN remains uncompromised. Privacy ($\mathcal{B}$) is essential because, with ZTN's autonomous operations, maintaining user data confidentiality becomes more critical to prevent unauthorized data access. The Intrusion Detection System (IDS) ($\mathcal{C}$), utilizing Digital Twins ($\mathcal{J}$), offers a powerful mechanism for ZTN: these virtual replicas of the network to allow for real-time simulations, enabling the IDS to proactively identify and counter potential threats without disrupting the actual network. Explainable AI ($\mathcal{D}$) is vital for ZTN as stakeholders need to understand AI-driven decisions in an environment that thrives

on automation. Ledger Distribution ($\mathcal{E}$) offers a decentralized record-keeping mechanism, ensuring that data within ZTN is tamper-proof and verifiable. Smart Contracts ($\mathcal{F}$) bring in efficiency and trust in ZTN, enabling automated agreements that execute when conditions are met. Transparency ($\mathcal{G}$) in ZTN ensures stakeholders can monitor and validate the automated processes, ensuring accountability. Decentralization ($\mathcal{H}$) is a foundational principle of ZTN, eliminating single points of failure and promoting robustness. Trust ($\mathcal{I}$) is especially vital in ZTN's decentralized landscape, ensuring that automated processes and interactions are credible.

5.7. Discussion

The proposed SALSTM aligns with the single-step prediction paradigm where the model predicts the classification of the current packet or sequence based on the immediate past data. The self-attention mechanism allows the model to weigh the importance of different packets or features in a sequence, helping it to identify patterns that are indicative of malicious behavior. By focusing on crucial segments of the input, the model can make a more accurate single-step prediction, especially when considering sequential data where long-term dependencies exist. On the other hand, the proposed IDS has an impact on DTs and the physical world. (1) DTs Impact: A DTs is a digital replica of a physical system. Any prediction made by the SALSTM can be first tested and visualized on the DTs. For intrusion detection, if an attack is predicted, the DTs can simulate the effects of that attack on the digital system, allowing for risk-free testing and validation. (2) Physical World Impact: Once predictions are validated on the Digital Twins, they can inform decisions in the real-world system. For instance, If an imminent threat is detected, immediate protective measures can be initiated in real-time to prevent potential harm.

6. Conclusion

In this paper, we have proposed a robust cybersecurity framework by integrating smart contracts and eXplainable AI. Specifically, first, we have designed a smart contract and blockchain-enabled authentication scheme to ensure the trustworthiness of data acquisition from different participating sources. Then, a DT was set up with different attack scenarios to design and test the effectiveness of an intrusion detection system in the ZTN environment. To detect attacks, a self-attention-based long short-term memory was designed and deployed. We also used the SHAP tool to understand the reasoning and effects of the features contributing towards higher accuracy of the proposed IDS. The experiment result shows that the proposed framework significantly outperforms the baselines and state-of-the-art techniques in terms of explainability, accuracy, and other crucial security parameters but has few limitations. For instance, the proposed approach's reliance on specific datasets could affect broader applicability, and the sole use of SHAP might not capture every interpretative nuance. Future research will include implementing the proposed framework in a federated ZTN scenario with other explainability methods.

CRediT authorship contribution statement

Randhir Kumar: Writing – review & editing, Writing – original draft, Validation, Methodology, Conceptualization. **Ahamed Aljuhani:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Data curation. **Danish Javeed:** Writing – review & editing, Writing – original draft, Validation, Investigation, Data curation, Conceptualization,. **Prabhat Kumar:** Writing – review & editing, Writing – original draft, Validation, Methodology, Formal analysis, Data curation, Conceptualization. **Shareeful Islam:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Formal analysis. **A.K.M. Najmul Islam:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the European Union's Horizon Europe research and innovation program under grant agreement No 101120779, for the CyberSecDome project — An innovative Virtual Reality based intrusion detection, incident investigation, and response approach for enhancing the resilience, security, privacy, and accountability of complex and heterogeneous digital systems and infrastructures. This work was also partially supported by the Research Council of Finland with CHIST-ERA, grant agreement no - 359790, Di4SPDS-Distributed Intelligence for Enhancing Security and Privacy of Decentralised and Distributed Systems.

References

[1] M. Liyanage, Q.-V. Pham, K. Dev, S. Bhattacharya, P.K.R. Maddikunta, T.R. Gadekallu, G. Yenduri, A survey on zero touch network and service (ZSM) management for 5G and beyond networks, J. Netw. Comput. Appl. (2022) 103362.

[2] J. Gallego-Madrid, R. Sanchez-Iborra, P.M. Ruiz, A.F. Skarmeta, Machine learning-based zero-touch network and service management: A survey, Digit. Commun. Netw. 8 (2) (2022) 105–123.

[3] R. Kumar, P. Kumar, M. Aloqaily, A. Aljuhani, Deep learning-based blockchain for secure zero touch networks, IEEE Commun. Mag. (2022).

[4] R. Kumar, A. Aljuhani, P. Kumar, A. Kumar, A. Franklin, A. Jolfaei, Blockchain-enabled secure communication for unmanned aerial vehicle (UAV) networks, in: Proceedings of the 5th International ACM Mobicom Workshop on Drone Assisted Wireless Communications for 5G and beyond, 2022, pp. 37–42.

[5] A. Aljuhani, Machine learning approaches for combating distributed denial of service attacks in modern networking environments, IEEE Access 9 (2021) 42236–42264.

[6] L.M. Contreras, J. Serrano, L. Mamatas, G. Bernini, P. Monti, M. Antunes, U. Atmojo, E. Tocker, I. Val, A. Sgambelluri, et al., Modular architecture providing convergent and ubiquitous intelligent connectivity for networks beyond 2030, 2022.

[7] C. Benzaid, T. Taleb, AI-driven zero touch network and service management in 5G and beyond: Challenges and research directions, Ieee Netw. 34 (2) (2020) 186–194.

[8] D. Javeed, M.S. Saeed, I. Ahmad, P. Kumar, A. Jolfaei, M. Tahir, An intelligent intrusion detection system for smart consumer electronics network, IEEE Trans. Consum. Electron. (2023).

[9] S. Mane, D. Rao, Explaining network intrusion detection system using explainable AI framework, 2021, arXiv preprint arXiv:2103.07110.

[10] Y. Luo, X. Chen, N. Ge, W. Feng, J. Lu, Transformer-based device-type identification in heterogeneous IoT traffic, IEEE Internet Things J. 10 (6) (2022) 5050–5062.

[11] M. Wang, K. Zheng, Y. Yang, X. Wang, An explainable machine learning framework for intrusion detection systems, IEEE Access 8 (2020) 73127–73141.

[12] D. Javeed, T. Gao, M.S. Saeed, M.T. Khan, FOG-empowered augmented intelligence-based proactive defensive mechanism for IoT-enabled smart industries, IEEE Internet Things J. (2023).

[13] Z. Abou El Houda, B. Brik, L. Khoukhi, "Why should i trust your ids?": An explainable deep learning framework for intrusion detection systems in internet of things networks, IEEE Open J. Commun. Soc. 3 (2022) 1164–1176.

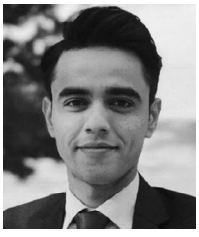
[14] A. Oseni, N. Moustafa, G. Creech, N. Sohrabi, A. Strelzoff, Z. Tari, I. Linkov, An explainable deep learning framework for resilient intrusion detection in IoT-enabled transportation networks, IEEE Trans. Intell. Transp. Syst. (2022).

[15] M.M. Alani, E. Damiani, U. Ghosh, DeepIIoT: An explainable deep learning based intrusion detection system for industrial IOT, in: 2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops, ICDCSW, IEEE, 2022, pp. 169–174.

[16] S. Roy, J. Li, V. Pandey, Y. Bai, An explainable deep neural framework for trustworthy network intrusion detection, in: 2022 10th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud, IEEE, 2022, pp. 25–30.
[17] D. Yaga, P. Mell, N. Roby, K. Scarfone, Blockchain technology overview, 2019, arXiv preprint arXiv:1906.11078.
[18] T.R. Gadekallu, Q.-V. Pham, D.C. Nguyen, P.K.R. Maddikunta, N. Deepa, B. Prabadevi, P.N. Pathirana, J. Zhao, W.-J. Hwang, Blockchain for edge of things: Applications, opportunities, and challenges, IEEE Internet Things J. 9 (2) (2021) 964–988.
[19] G. Thakur, P. Kumar, S. Jangirala, A.K. Das, Y. Park, et al., An effective privacy-preserving blockchain-assisted security protocol for cloud-based digital twin environment, IEEE Access 11 (2023) 26877–26892.
[20] M. Eckhart, A. Ekelhart, Digital twins for cyber-physical systems security: State of the art and outlook, in: Security and Quality in Cyber-Physical Systems Engineering: With Forewords by Robert M. Lee and Tom Gilb, Springer, 2019, pp. 383–412.
[21] B.R. Barricelli, E. Casiraghi, D. Fogli, A survey on digital twin: Definitions, characteristics, applications, and design implications, IEEE Access 7 (2019) 167653–167671.
[22] S.A. Varghese, A.D. Ghadim, A. Balador, Z. Alimadadi, P. Papadimitratos, Digital twin-based intrusion detection for industrial control systems, in: 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events, PerCom Workshops, IEEE, 2022, pp. 611–617.
[23] Y. Yigit, B. Bal, A. Karameseoglu, T.Q. Duong, B. Canberk, Digital twin-enabled intelligent ddos detection mechanism for autonomous core networks, IEEE Commun. Stand. Mag. 6 (3) (2022) 38–44.
[24] K. Kobayashi, B. Almutairi, M.N. Sakib, S. Chakraborty, S.B. Alam, Explainable, interpretable & trustworthy AI for intelligent digital twin: Case study on remaining useful life, 2023, arXiv preprint arXiv:2301.06676.
[25] S. Suhail, M. Iqbal, R. Hussain, R. Jurdak, ENIGMA: An explainable digital twin security solution for cyber–physical systems, Comput. Ind. 151 (2023) 103961.
[26] Y. Lu, X. Huang, K. Zhang, S. Maharjan, Y. Zhang, Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks, IEEE Trans. Ind. Inform. 17 (7) (2020) 5098–5107.
[27] M.A. Ferrag, B. Kantarci, L.C. Cordeiro, M. Debbah, K.-K.R. Choo, Poisoning attacks in federated edge learning for digital twin 6G-enabled IoTs: An anticipatory study, 2023, arXiv preprint arXiv:2303.11745.
[28] R. Bitton, T. Gluck, O. Stan, M. Inokuchi, Y. Ohta, Y. Yamada, T. Yagyu, Y. Elovici, A. Shabtai, Deriving a cost-effective digital twin of an ICS to facilitate security evaluation, in: Computer Security: 23rd European Symposium on Research in Computer Security, ESORICS 2018, Barcelona, Spain, September 3–7, 2018, Proceedings, Part I 23, Springer, 2018, pp. 533–554.
[29] R. Dave, IoT security and authentication schemes based on machine learning: Review, 2021, arXiv:2109.02695.
[30] P. Porambage, G. Gür, D.P.M. Osorio, M. Liyanage, A. Gurtov, M. Ylianttila, The roadmap to 6G security and privacy, IEEE Open J. Commun. Soc. 2 (2021) 1094–1122, http://dx.doi.org/10.1109/OJCOMS.2021.3078081.
[31] T.R. Gadekallu, M. M K, S.K. S, N. Kumar, S. Hakak, S. Bhattacharya, Blockchain-based attack detection on machine learning algorithms for IoT-based e-health applications, IEEE Internet Things Mag. 4 (3) (2021) http://dx.doi.org/10.1109/iotm.1021.2000160.
[32] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[33] D. Javeed, T. Gao, M.S. Saeed, P. Kumar, An intrusion detection system for edge-envisioned smart agriculture in extreme environment, IEEE Internet Things J. (2023).
[34] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, vol. 30, 2017.
[35] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, Y. Elovici, N-baiot—network-based detection of iot botnet attacks using deep autoencoders, IEEE Pervasive Comput. 17 (3) (2018) 12–22.
[36] D. Javeed, T. Gao, P. Kumar, A. Jolfaei, An explainable and resilient intrusion detection system for industry 5.0, IEEE Trans. Consum. Electron. (2023).
[37] K. Saha, M.M.R. Fakir, M.A. Hashem, An unsupervised self-organizing map assisted deep autoencoder gaussian mixture model for IoT anomaly detection, in: 2021 5th International Conference on Electrical Information and Communication Technology, EICT, IEEE, 2021, pp. 1–6.
[38] T.B. Seong, V. Ponnusamy, N. Jhanjhi, R. Annur, M. Talib, A comparative analysis on traditional wired datasets and the need for wireless datasets for IoT wireless intrusion detection, Indonesian J. Electr. Eng. Comput. Sci. 22 (2) (2021) 1165–1176.
[39] D.T. Rahmantyo, B. Erfianto, G.B. Satrya, Deep residual cnn for preventing botnet attacks on the internet of things, in: 2021 4th International Conference of Computer and Informatics Engineering, IC2IE, IEEE, 2021, pp. 462–466.

Randhir Kumar: received his Ph.D. in the department of Information Technology from NIT Raipur, India, and he is working as Postdoctoral Research Fellow in the department of CSE, IIT Hyderabad. He has published his research article in leading journal and conferences from IEEE, Elsevier, Springer, and John Wiley. His paper has been published in some of the high impact factor journals such as — IEEE Internet of Things, IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Network Science and Engineering, IEEE Transactions on Green Communications and Networking, IEEE Transactions on Industrial Informatics, IEEE COMSNETs, IEEE ICC, Computer Networks, JPDC, and Transactions on Emerging Telecommunications Technologies (ETT Wiley). He has qualified UGC-NET in the year of 2018. He has been awarded as Best Engineer Trainee by Honeywell Technology, India in the year 2009. His research interest includes cryptographic techniques, information security, blockchain technology, and web mining. He is also an IEEE Member.

Ahamed Aljuhani is currently an assistant professor and the chair of the Department of Computer Engineering, College of Computing and Information Technology, University of Tabuk, Tabuk, Saudi Arabia. His research interests include information security, network security and privacy, secure system design, and system development. Aljuhani received his Ph.D. degree in computer science/information security track from The Catholic University of America, Washington, DC, USA. Contact him at a_aljuhani@ut.edu.sa.



Danish Javeed is currently pursuing a Ph.D. degree in Software Engineering, specializing in Information Security with the Software College, Northeastern University, China under the prestigious fellowship of Ministry of Education funded by the Government of China. He got his M.E degree in Computer Applied Technology from Changchun University of Science and Technology, China, under the same fellowship in 2020. He is also working on various research projects with researchers from the LUT School of Engineering Science, LUT University, Lappeenranta, Finland. He has many research contributions in the areas of Deep Learning, Cybersecurity, Intrusion Detection and Prevention Systems, the Internet of Things, Software-defined Networking, and Edge Computing. He has authored or co-authored over 20+ publications in high-ranked journals and conferences.

Prabhat Kumar received his Ph.D. degree in Information Technology, National Institute of Technology Raipur, Raipur, India, under the prestigious fellowship of Ministry of Human Resource and Development (MHRD) funded by the Government of India in 2022. Thereafter, he worked with Indian Institute of Technology Hyderabad, India as a Post-Doctoral Researcher under project" Development of Indian Telecommunication Security Assurance Requirements for IoT devices". He is currently working as Post-Doctoral Researcher with the Department of Software Engineering, LUT School of Engineering Science, LUT University, Lappeenranta, Finland. He has many research contributions in Machine Learning, Deep Learning, Federated Learning, Big Data Analytics, Cybersecurity, Blockchain, Cloud Computing, Internet of Things and Software Defined Networking. He has authored or co-authored over 35+ publications in high-ranked journals and conferences, including 13+ IEEE TRANSACTIONS paper. One of his Ph.D. publications was recognized as a top cited article by WILEY in 2020-21. He is IEEE Consumer Technology Society (CTSoc) Technical Committee member in Machine learning, Deep learning, and AI in Consumer Electronics. He has served as a program co-Chair, and a Technical Program Committee member for major conferences, including IEEE ICCE and ACM CCS. He is also an IEEE Member.

Shareeful Islam is currently working at the School of Computing and Information Science, Anglia Ruskin University, UK. He was the visiting researcher at the National Institute of Informatics (NII), Japan and SBA research, Austria. His research interests lie in the areas of cybersecurity, risk management, requirement engineering and information systems. He has pioneered work in developing risk assessment and treatment methods using business and technical goals, modeling language for cybersecurity risk management. The works are implemented in various application domains including cloud migration, critical infrastructure, and healthcare sector cybersecurity. He has published more than 70 papers (h-index 26) and he has led and/or participated in projects funded by the European Union (FP7), Innovate UK, FwF, and DAAD. He has experience of acting as evaluator for national and international funding bodies including the EPSRC, FwF, and CHIST-ERA. His e-mail address is Shareeful.islam@aru.ac.uk.

A.K.M. Najmul Islam is a Full Professor at LUT University, Finland. He conducts cross-disciplinary research in digitalization and its impact on citizens, organizations, and society. He is a docent of Information Systems at Tampere University. Islam's publication has appeared in top Information Systems outlets such as Journal of Strategic Information Systems, European Journal of Information Systems and Information Systems Journal. He has published in other highly ranked interdisciplinary journals such as IEEE Access, Computers & Education, Technological Forecasting and Social Change, International Journal of Information Management, Information Technology & People, Computers in Human Behavior, Computers in Industry, Internet Research, Communications of the AIS, among others. He is currently serving as a Senior Editor for Information Technology & People journal.