Deep Learning and Onion Routing-based Collaborative Intelligence Framework for Smart Homes underlying 6G Networks

Nilesh Kumar Jadav, Rajesh Gupta, *Student Member, IEEE*, Mohammad Dahman Alshehri, Harsh Mankodiya, Sudeep Tanwar, *Senior Member, IEEE*, Neeraj Kumar, *Senior Member, IEEE*

Abstract

Sensor communication in the smart home environment is still in its infancy as the information exchange between sensors is vulnerable to security threats. Many traditional solutions use single-layer or multi-layer (i.e., onion routing protocol) encryption/decryption algorithms. But, in the traditional onion routing protocol, if the directory server is compromised, it may not track the malicious onion nodes within the onion network. It questioned the path anonymity of the onion routing protocol. Motivated by this, we proposed a blockchain and onion routing (OR)-based secure and trusted framework in the paper. The anonymity of the proposed OR network is maintained by storing and tracking the onion nodes threshold values through the blockchain network. A long short-term memory (LSTM) model is also utilized to classify the sensors data requests as malicious and non-malicious. The performance of the proposed system is evaluated with different performance metrics such as F1 score and accuracy. The LSTM model significantly improves the initial detection rate of malicious data requests from smart home sensors. Over these benefits, we considered the entire communication via 6G channel, reducing the

N. K. Jadav, R. Gupta, Mohammad Dahman Alshehri, H. Mankodiya, and S. Tanwar are with the Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad, Gujarat, India 382481. e-mails: (21ftphde53@nirmauni.ac.in, 18ftvphde31@nirmauni.ac.in, 19bce117@nirmauni.ac.in, sudeep.tanwar@nirmauni.ac.in).

Mohammad Dahman Alshehri is working with the Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia. (email: alshehri@tu.edu.sa)

N. Kumar is working with the Department of Computer Science Engineering, Thapar Institute of Engineering andTechnology, Deemed to be University, Patiala, Punjab, India, Department of Computer Science and Information Engineering, Asia University,Taiwan, and the School of Computer Science, University of Petroleum and Energy Studies, Dehradun, Uttarakhand, 248001, India. (email: neeraj.kumar@thapar.edu)

overall communication latency. Additionally, the OR network is simulated over the shadow simulator to analyze the OR network's performance considering parameters such as packet delivery ratio and malicious onion node detection rate.

Index Terms

6G, Artificial Intelligence, Anonymity, Blockchain, Deep Learning, LSTM, Onion Routing, Smart home.

I. Introduction

An indispensable IoT evolution, has enabled billions of smart devices to provide essential services in smart home system(SHS). It is a technology where home appliances are remotely monitored and controlled through an intelligent and cooperative network interface via different sensors deployed in the house. Its proliferation adheres to new facilities for user comforts, such as smart lighting, home entertainment system, smart fridge, garage door openers, and environmental controls. In addition, it facilitates high-level relief to the disabled and elderly people by providing on-demand services at a fingertip. However, in conjunction with its steady development, it is vividly disrupted by severe security incidents. For instance, unlocking doors, eavesdropping through their exploited cameras, manipulating the virtual assistant devices and smoke alarms. Furthermore, the smart sensors in SHS are resource-constrained with insufficient battery backup. A distributed denial-of-service (DDoS) attack influences the attackers to send multiple requests to the sensors and break off their resources. As a result, it deteriorates the energy efficiency of the SHS to violate the green communication between sensors.

Over the years, researchers have been proposing semantic solutions to tackle the security issues in the SHS. One of the promising approach is to use cryptographic solutions such as authentication, key generation and distribution that can relatively solve the aforementioned security issues. The authors of [1] presented an authentication scheme and provided a secure mechanism to send privacy-preserving queries for a SHS. However, the proposed work has a centralized system like a service provider and gateway, which has a high risk of being compromised. Next, Song *et al.* in [2] proposed a secure mechanism to stop traffic interception and analysis attacks by directing the network traffic from different router gateways. The adaptation to multi-hop gateways, brings traffic obfuscation which is challenging to analyze by the attackers. Later, [3] presented a secure message exchange between sensors by using symmetric key encryption and hash function to

preserve the confidentiality, integrity, and availability of the SHS. Nonetheless, the ciphertext is forwarded through the Internet, which uses conventional routing with single layer encryption such as transport layer security (TLS) or secure socket layer (SSL) and exchanges the routing table with each other. This approach is prone to network correlation attacks, where the attackers can analyze the route path information from the exploited routing tables.

Furthermore, it is relatively easy for an attacker to break off the cipher and extract the plaintext message with modern computing power. Therefore, there is a stringent requirement for an anonymous and multi-layer encrypted network, i.e., onion routing (OR) [4][5]. It is a pool of anonymous onion nodes (ONs), from which a few nodes are selected randomly to relay the message from source to destination [6][7]. Currently, the OR is integrated into various applications to preserve and secure their communication. For instance, the authors of [8] utilize the OR in data caching, where the content from the producer to consumer is protected from the attackers. For that, they created an OR-based hidden service to relay their content. Then, Sakai *et al.* in [9] examined the performance of the ON using data rates, path information, anonymity levels, and cost for the delay-tolerant network. It has been observed in the existing work that, the OR network has to process both the data that is malicious and non-malicious from the SHS which significantly raises the computational time of the OR network.

Hence, an efficient artificial intelligence (AI) model is required to classify malicious and non-malicious sensor data so that the OR network endures only non-malicious data from the SHS. The authors of [10] address the energy theft issues in smart home systems, to which they integrate AI schemes to filter out abnormalities and improve the decision capabilities of the sensors. Further, to enhance the security of low-power IoT devices in smart homes, [11] introduced an intrusion detection system to detect and classify multiple routing attacks in a single run. They utilize the signature and behavior-based detection rule to detect the anomaly in the network traffic. Reinforcement learning is also coming as an emerging technology that enhances the security of sensor communication by adopting better decision policies [12]. However, the aforementioned solutions are not confronting data integrity and data tampering issues that thwart the performance of smart homes. Therefore, there is a need for a conspicuous technology, i.e., blockchain, to securely store the correctly classified sensor data from the OR network. The authors of [13] proposed a combinatorial approach by combining the authentication scheme and smart contracts that adhere to the data security and privacy goals for smart homes. Their proposed method outperforms against tampering, denial-of-service (DoS), and data scraping

attacks. Further, to enhance the energy efficiency and data privacy of the SHS, [14] presented a distributed approach to optimally manage the energy via blockchain smart contracts. The result shows that their approach is computationally inexpensive. However, storing large transactions inside the blockchain can significantly increase the computational cost of the system.

Motivated by the challenges mentioned above, we introduce an amalgamation of AI and OR network to develop a collaborative intelligence framework for smart homes underlying the 6G network. Initially, an AI-based model, i.e., long short-term memory (LSTM), detect and classify non-malicious and malicious request from the sensors. Nevertheless, the AI model consumes a considerable amount of energy from resource-constrained devices that leave carbon footprints and dissatisfies the need for green communication. To bridge the gap, we incorporate edge nodes where all the AI processing transpires. Then, the correctly classified sensor data is multi-layered encrypted and passed to the OR network. Further, blockchain technology fused with smart contracts adroitly tackles the security threats of a SHS. Finally, encompassing the 6G network interface enhances the scalability and latency of the proposed framework.

A. Contributions

Following are the major contributions of the paper.

- We proposed a blockchain and OR-based secure and trusted data request exchange framework for sensor communication in smart home environment. The proposed framework also protects ONs from being compromised by tracking the threshold of ONs.
- We integrate LSTM model with the proposed blockchain and OR-based collaborative framework for malicious sensor data request classification.
- The proposed framework is evaluated against different deep learning metrics such as, F1-score and accuracy. Training accuracy and loss score graph are also illustrated to exhibit the significant performance of the intelligence.
- An onion routing network is simulated inside the shadow simulator to examine the malicious node by using their parameter threshold. Then, it is evaluated against the packet delivery ratio and detection rate in the presence of malicious ONs.

B. Organization

The rest of the paper is organized as follows. Section 2 introduces the system model and problem formulation. Section 3 presents the proposed collaborative intelligence framework for

smart home systems. Section 4 presents the results and discussion. Finally, Section 5 gives the concluding remarks.

II. System Model and Problem Formulation

In this subsection, we formulate the problem considering the smart home scenario illustrated in Fig. 3. Its primary concern is to secure sensor ($\mathcal{S}$) communication data ($\mathcal{D}$) along with its routing path anonymity. In the proposed framework, $\{s_1, \ldots, s_i, \ldots, s_j, \ldots, s_S\} \in \mathcal{S}$ is the set of sensors deployed within the smart home scenario. Each sensor s_i has to communicate/share data request $\{d_1, \ldots, d_i, \ldots, d_j, \ldots, d_D\} \in \mathcal{D}$ (s.t. $D = S$) with other sensor s_j to perform specific sequence of tasks. Traditionally, symmetric/asymmetric encryption algorithms such as Elliptic curve, Rivest–Shamir–Adleman (RSA), Elgamal encryption, and many more can be used to secure the data exchange between smart home sensors s_i and s_j. The encryption process of the aforementioned traditional algorithms can be compromised with the high performance computing capability systems. This questions the security of data exchange between s_i and s_j. Thus, there is a need for a procedure or protocol that brings multi-layer encryption to the data d_i routing between s_i and s_j. So, the onion routing protocol is quite helpful in this regard that achieves end-to-end data security with high anonymity.

Fig. 1 shows the working of onion protocol with k layers of encryption between sensors s_i and s_j. Initially, the source sensor s_i encrypts the message with multiple layers of encryption (number of encryption layers depends upon the number of nodes in the onion network). The encryption process of both the traditional routing and onion routing protocol-based data exchange is elaborated as follows.

$$d_i' = \psi(d_i), \forall 1 \leq i \leq D \tag{1}$$

$$d_i' = \psi(\psi(\ldots(\psi(d_i)))), \forall 1 \leq i \leq D \tag{2}$$

where d_i' is the encrypted form of the original data request d_i from the i^{th} sensor. ψ shows the encryption process, which can be any one of the mentioned above. Eq. 1 depicts the traditional form of the message encryption, whereas Eq. 2 depicts the multi-layer encryption possessed in the onion routing protocol.

In onion routing, if any sensor s_i wants to share a data request d_i with other sensor s_j, then s_i need to encrypt the data request d_i with the ON's O_k public key $\mathbb{P}_k$. Then, O_k decrypt the received data request d_i from the sensor s_i with its secret key $\mathbb{S}_k$. Likewise, each ON has its own

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2022.3164715, IEEE Transactions on Network and Service Management

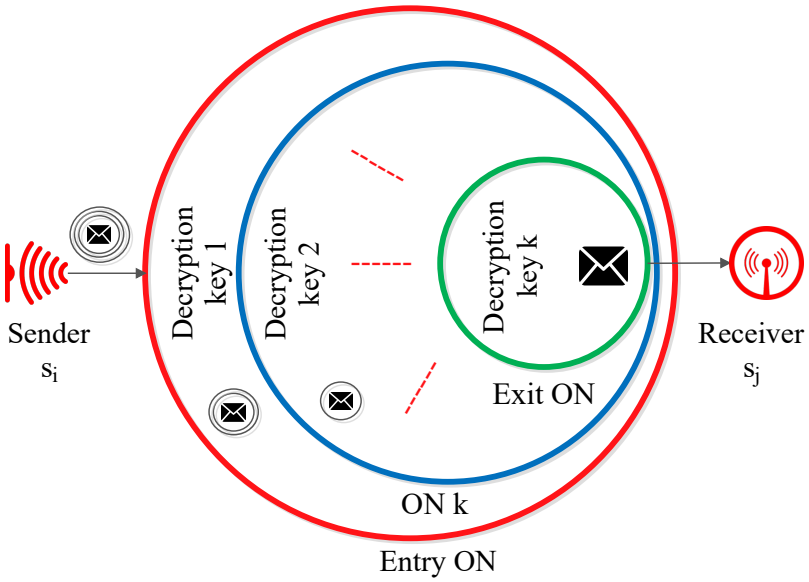


Fig. 1: Layout of onion protocol with k layers of encryption between s_i and s_j.

secret and public keys, which they have generated using ElGamal algorithm, which is shown as follows.

$$\mathbb{P} = \mathbb{P}_{x(1 \leq x \leq k)} = \{\mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_k\} \tag{3}$$

$$\mathbb{S} = \mathbb{S}_{x(1 \leq x \leq k)} = \{\mathbb{S}_1, \mathbb{S}_2, \ldots, \mathbb{S}_k\} \tag{4}$$

where Eq. 3 and 4 depicts the set of public and private keys of ONs respectively. Each O_k generates different set of public keys and private keys $\{\mathbb{P}_1^i, \mathbb{P}_2^i, \ldots, \mathbb{P}_k^i\}$ $\forall 1 \leq i \leq \mathcal{S}$ and $\{\mathbb{S}_1^i, \mathbb{S}_2^i, \ldots, \mathbb{S}_k^i\}$ $\forall 1 \leq i \leq \mathcal{S}$, respectively for different senors in $\mathcal{S}$. The sensor s_i shares a message d_i with s_j, which is encrypted using the public key of ONs is shown as follows.

$$d_i' = \mathbb{P}_1^i(\mathbb{P}_2^i(\ldots(\mathbb{P}_k^i(d_i)))) \tag{5}$$

Each onion layer O_k decrypts the original message with their private key $\mathbb{S}_k^i$.

Fig. 2 shows the conceptual view of the key exchange process between O_k and s_i. Here, O_k generate its own public $\mathbb{P}_k$ and private keys $\mathbb{S}_k$ from the cyclic group $\mathbb{G} = \langle \mathbb{Z}_\rho^*, \times \rangle$, where ρ is a large prime integer value. O_k then select a group member g_i from $\mathbb{G}$ with a condition that GCD(g_i, ρ) = 1 and $1 \leq g_i \leq \rho - 2$. From this, the O_k generate its public $\langle g_j, \alpha, \rho \rangle$ and private key $\mathbb{S}_k$. It then publish its public key with s_i. s_i encrypts the data request d_i with the public

Onion node (O_k)	**Source sensor** (s_i)
# Generate public and private keys 1. Select_large_prime_integer $\langle \rho \rangle$ 2. Generate_cyclic_group $\langle \mathbb{G} \rangle$ 3. Select $g_i \in \mathbb{G} = \langle \mathbb{Z}_\rho^*, \times \rangle$ $1 \leq g_i \leq \rho - 2$ GCD(g_i, ρ) = 1 4. Select g_j, the primitive root of $\langle \mathbb{Z}_\rho^*, \times \rangle$ 5. Calculate $\alpha \leftarrow g_j^{g_i} mod\ \rho$ 6. Private_key $\mathbb{S}_k \leftarrow g_i$ 7. Publish_public_key $\mathbb{P}_k$ $\mathbb{P}_k \leftarrow \langle g_j, \alpha, \rho \rangle$	
$\xrightarrow[\textbf{Share } \mathbb{P}_k \textbf{ with } s_i]{\langle g_j, \alpha, \rho \rangle}$	
	# Encrypt d_i using $\mathbb{P}_k$ 1. Select random no. $g_l \in \mathbb{G} = \langle \mathbb{Z}_\rho^*, \times \rangle$ s.t. GCD(g_l, ρ) = 1 2. Generate_ciphertexts α', α'' $\alpha' \leftarrow g_j^{g_l}\ mod\ \rho$ $\alpha'' \leftarrow (d_i \times \alpha^{g_l})\ mod\ \rho$ 3. Send $\langle \alpha', \alpha'' \rangle$ to O_k
	$\xleftarrow[\textbf{Sends to } O_k]{\langle \alpha', \alpha'' \rangle}$
# Decrypt the received message $d_i \leftarrow [\alpha''(\alpha'^{g_i})^{-1}]$ mod ρ	

Fig. 2: Secret data exchange between s_i and O_k.

key $\mathbb{P}_k$ of O_k. Now, O_k receives a message encrypted with its public key and decrypt with its private key $\mathbb{S}_k$. Let us consider a scenario, where the smart home sensor s_i or intermediate ON O_k or both are compromised. In this case, the malicious sensor/router will get the secret data request and anonymous path too. This bring insecurities in the onion routing protocol.

Based on the aforementioned facts, the dual objectives OF_1 and OF_2 of the paper is to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2022.3164715, IEEE Transactions on Network and Service Management

enhance the security in the onion routing protocol as well as the source sensor, which is defined as follows.

$$OF_1 = \max \sum_{i=1}^{S} \text{Secure}(s_i, d_i) \tag{6}$$

$$OF_2 = \max \sum_{x=1}^{k} \text{Secure}(O_x) \tag{7}$$

$$1 \leq i \leq S$$

$$1 \leq x \leq k$$

III. Proposed Collaborative Intelligence Framework

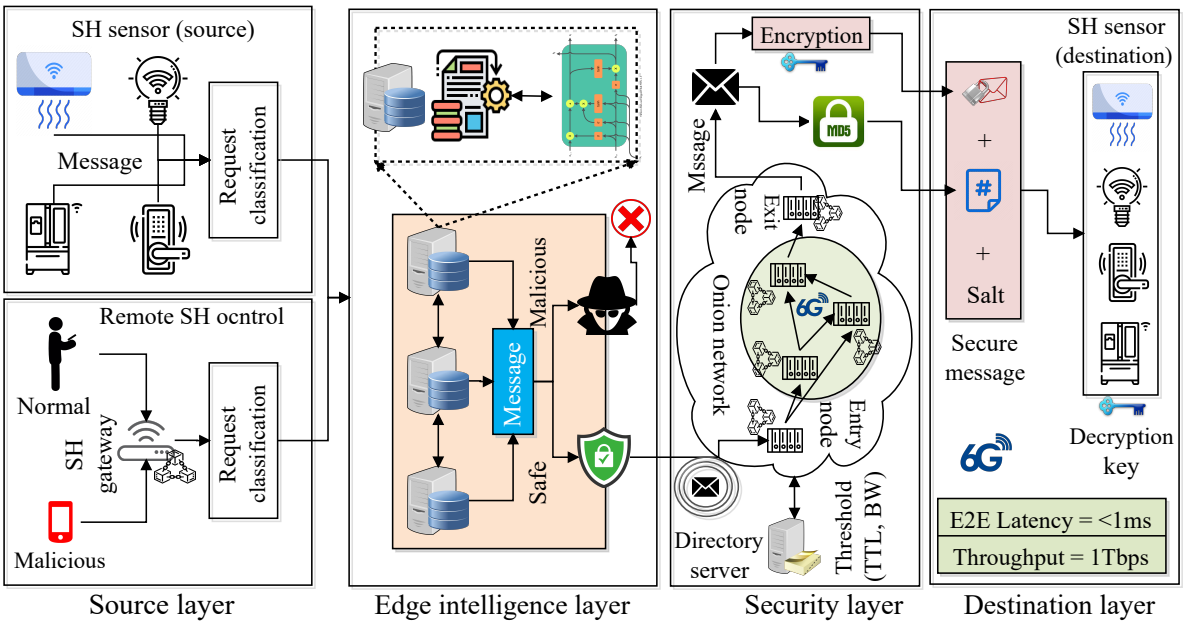


Fig. 3: Architecture

We illustrated the proposed framework in Fig. 3 with four distinct layers, i.e., source, edge intelligence, security, and destination layer. A thorough explanation of each layer is as follows.

A. Source Layer

The sensor layer in the proposed framework consists of various interconnected sensors with a separate set of functionalities. In the SHS, these sensors have shared goals and streamlined tasks

for which they have to communicate with each other to jointly complete the desired operation, such as turning the lights on or off for all rooms, multifarious environmental controls, etc. The attackers can disrupt the SHS sensor communication by resorting the device vulnerabilities, weak network interfaces, and poor configurations. An attacker needs a single compromising device through which he can proliferate several attacks in the SHS. Therefore, to preserve the security of the SHS, we analyze the sensor's message exchange in two ways. In the first scenario, we have multiple source sensors $\{s_1, \ldots, s_i, \ldots, s_j, \ldots, s_S\} \in \mathcal{S}$ ready to send a data request$\{d_1, \ldots, d_i, \ldots, d_j, \ldots, d_D\} \in \mathcal{D}$ (s.t. $D = S$) to the destination sensor s_j for smart home operation. An attacker K attempts to manipulate the data request d_i by maneuvering the passive attacks, such as MiTM, session hijacking, etc.

$$s_i \xrightarrow{\text{original } d_i} s_j \tag{8}$$

where, $0 < s_S \leq s_j$

$$s_i \xrightarrow{\text{original } d_i} K \xrightarrow{\text{malicious } d_i} s_j \tag{9}$$

The destination sensor obtains the malicious d_i, which deteriorates the goals of SHS. In another scenario, we have a remote user who monitors and controls the smart home devices of the SHS. The smart home gateway Υ obtains the d_i from the remote user Ψ, and then it is routed to the desired sensor for a specific operation. Nonetheless, there is a possibility of an intrusion by forwarding the malicious d_i to the smart home gateway by the attacker K.

$$\Psi \xrightarrow{\text{original } d_i} \Upsilon \xrightarrow{\text{original } d_i} s_j \tag{10}$$

$$K \xrightarrow{\text{malicious payload}} \Psi \xrightarrow{\text{malicious } d_i} \Upsilon \xrightarrow{\text{malicious } d_i} s_j \tag{11}$$

Consequently, there is a prerequisite for intelligence that profoundly detect and categorize malicious and non-malicious data request.

$$\exists\ \mathbb{I}\ \forall(d_i, d_i^{''}) \xrightarrow{\text{classifies}} d_i = 0 \text{ and } d_i^{''} = 1 \tag{12}$$

where, d_i = 0 and $d_i^{''}$ resembles original and malicious data request respectively and $\mathbb{I}$ represents the intelligence.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2022.3164715, IEEE Transactions on Network and Service Management

B. AI Layer

This section discusses in detail about the proposed AI layer to detect anomaly in the IoT based systems by analysing their transmitted data. Dataset description along with all the sub-steps which constitutes up the whole prediction pipeline that includes data preparation and preprocessing is also discussed further. Fig. 4 shows the refined AI layer diagram to detect intrusion on a IoT network.

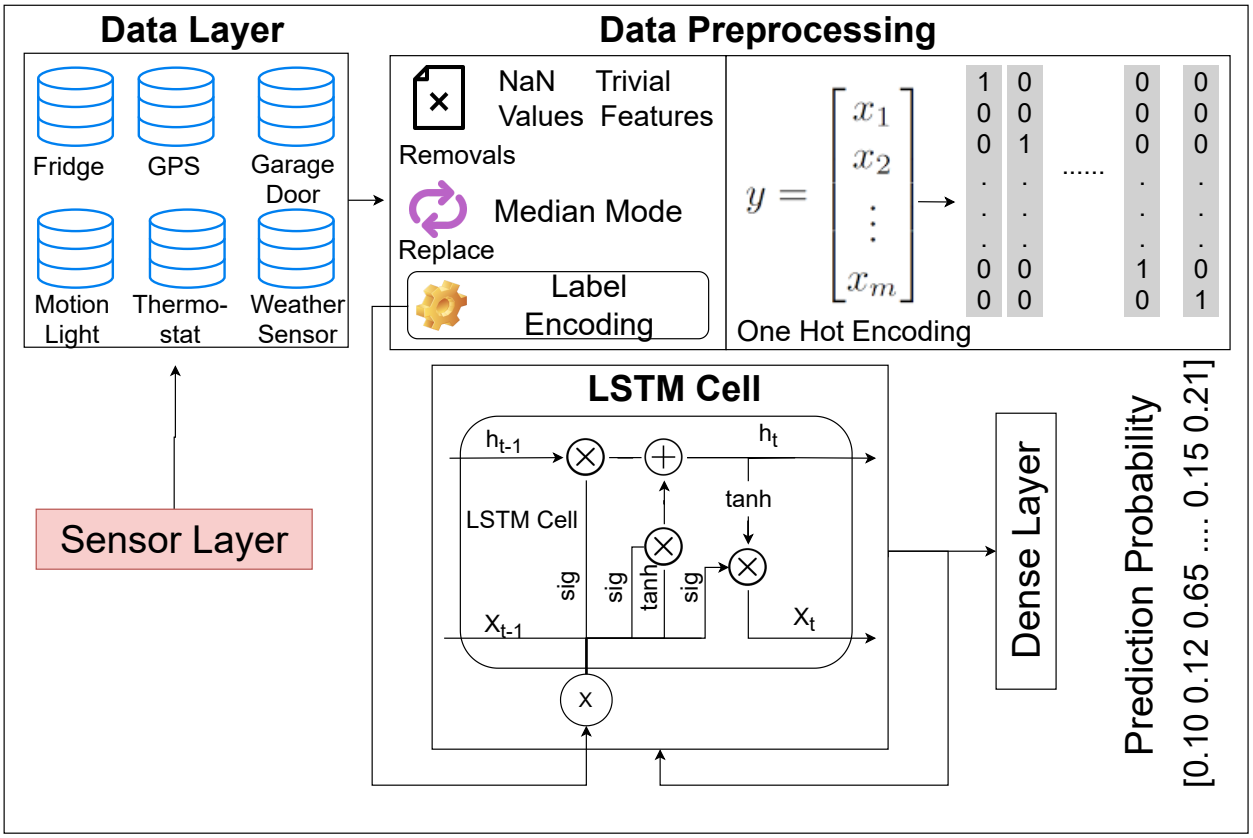


Fig. 4: AI Layer

a) Dataset Description: To train the LSTM [15] based intrusion detection system for IoT devices, TON_IoT datasets available at [16] have been used. This dataset consists of seven different files containing the sensor data from various IoT appliances namely fridge, garage door, motion light, thermostat, modbus, global positioning system (GPS) tracker and weather sensors. In each sensor file, corresponding data collected at some timestamp is present. Detailed description about each of the sensor data files can be obtained from [17]. In this paper, only the data transmitted from fridge, garage door, motion light, thermostat, GPS tracker and weather

sensors is taken into consideration for training and predictions. A tabular representation of the dataset and its sub-files can be seen in TABLE I.

TABLE I: Individual sensor dataset configuration with total instances, features, and target attack classes.

	Total Instances	**Total Features**	**Target Attack Classes**
Fridge Sensor	587076	6	normal, injection, ddos, password, xss, ransomware, backdoor
GPS Tracker	595686	6	normal, scanning, injection, ddos, password, xss, ransomware, backdoor
Garage Door	541177	6	normal, scanning, injection, ddos, password, xss, ransomware, backdoor
Motion Light	452262	6	normal, injection, ddos, scanning, password, xss, ransomware, backdoor
Thermostat	437607	6	normal, scanning, injection, password, xss, ransomware, backdoor
Weather Sensor	650242	7	normal, scanning, injection, ddos, password, xss, ransomware, backdoor

b) Data preparation and preprocessing: The dataset obtained is further preprocessed to make it ready to train on the sensor data. By analysing all the sub-files of the dataset, several columns with NaN/NULL values were found. NULL and NaN values in specific columns are then replaced with column median values. Furthermore, trivial columns such as date and timestamps are dropped from each training dataset. The target other columns with string as datatype along with target labels are label encoded to convert them into numerical format. To put it down mathematically, consider a set D representing the total overall dataset.

$$D = \{d_1, d_2, d_3, ..., d_s, ..., d_S\} \tag{13}$$

$$d_s \implies m_s \times n_s \tag{14}$$

$$\forall 1 \leq s \leq S$$

where $S = 6$ are total sensors/sensor data files taken into consideration and m_s = total instances for sensor data file s and n_s = total features for sensor data file s. Consider transformation function T to preprocess the input data. Applying T to the sensor data d_s yields transformed dataset D'

$$D' = T(D) \implies d'_s = T(d_s) \tag{15}$$

$$d'_s \implies m'_s \times n'_s \tag{16}$$

For each dataset d'_s, one hot encoding transformation ζ is applied to the target class column $y_{s_{m'_s \times 1}}$.

$$y_{s_{m'_s \times C_s}} := \zeta(y_{s_{m'_s \times 1}}) \tag{17}$$

where C_s is total number of unique target class for sensor data file s. Where $:=$ operation represents replacement operator. The set Y for dependent features for all the sensors can be obtained as $Y = \bigcup_S y_{s_{m'_s \times C_s}}$.

$$y_{s_{m'_s \times C_s}} = \begin{bmatrix} 0 \\ 1 \\ . \\ . \\ . \\ 0 \\ 0 \end{bmatrix} \cdots \begin{bmatrix} 1 \\ 0 \\ . \\ . \\ . \\ 0 \\ 0 \end{bmatrix} \cdots \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ . \\ 1 \\ 0 \end{bmatrix} \tag{18}$$

The final prepared dataset with independent feature set D' and dependent transformed feature set Y can be represented as follows.

$$X = \bigcup_S (d'_s, y_s) \tag{19}$$

c) Model Training and Prediction: As there are total S=6 total sensor data files, six LSTM models are trained on the individual sensor data. Predictions for each sensor is done on corresponding trained model to detect and classify the attack. Fig. 5 show the LSTM model constructed for the fridge sensor dataset. All the other architectures are constructed by considering the same overall structure with the only difference being the input and output shapes which depends upon the dataset. TABLE II shows the training configuration for the proposed LSTM models.

Activation function $tanh$ is used in the LSTM layer with a softmax activation with C_s neurons at the dense layer depending upon the sensor data that is being trained. The processed data

Algorithm 1 Data preparation algorithmic flow.

Input: Sensor data D

Initialization: S = D.length

Output: X

procedure T(d)

$d \leftarrow$ drop(d, columns=[date, timestamps])

$d_{column} \leftarrow d_{column}$.fillNull($d_{column}$.median())

$d_{column_{dtype=string}} \leftarrow$ labelEncoder($d_{column_{dtype=string}}$)

end procedure

procedure ONEHOTENCODING(y_s)

index $\leftarrow y_s$

zeros $\leftarrow$ numpy.zeros()

zeros[index] $\leftarrow$ 1

$y_s \leftarrow$ zeros

end procedure

for `s in range(S)` **do**

$d'_s \leftarrow$ T(d_s)

$y_s \leftarrow$ oneHotEncoding(y_s)

end for

D' $\leftarrow \bigcup_S d'_s$

X $\leftarrow (D', Y)$

TABLE II: Training Configuration for the prediction models

Epochs	10
Learning Rate	0.001
Batch Size	100
Loss Function	Categorical Crossentropy
Optimizer	Adam

$X = (D', Y)$ is passed on to the model for training. Trainable weights θ and biases ϕ can be

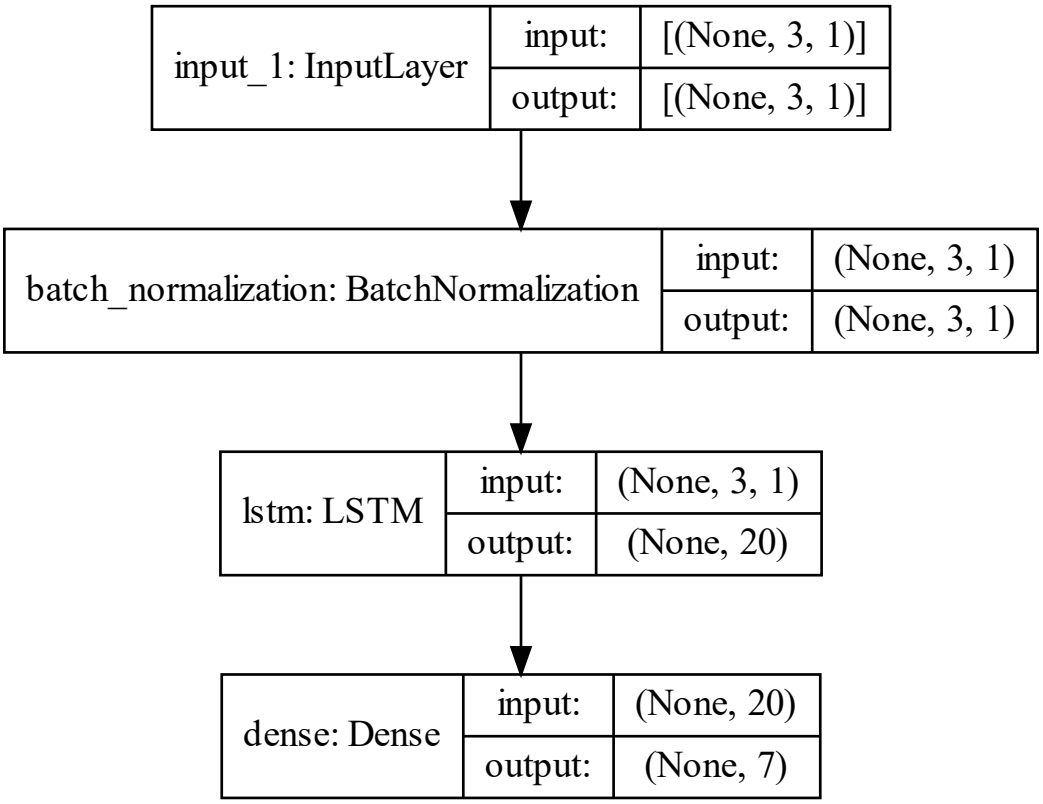


Fig. 5: LSTM Sequence model for fridge sensor data.

learnt by minimizing the categorical crossentropy loss function $L(\theta, \phi)$.

$$\hat{y_s} = \theta_s^T d_s' + \phi_s \tag{20}$$

$$Min\ L(\theta, \phi) = -\sum_{C_s} y_s log(Z(\hat{y_s})) \tag{21}$$

$$Z(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{22}$$

Z represents the tanh activation function [18]. The output probability of the trained model is be calculated using the softmax activation function [19] at the dense layer.

$$softmax(\hat{y_s}) = \frac{e^{\hat{y_{s_l}}}}{\sum\limits_{l \epsilon C_s} e^{\hat{y_{s_l}}}} \tag{23}$$

where l are the predicted target class labels for the sensor data file s.

C. Security Layer

In the edge intelligence layer, the LSTM model trains on six different types of security attacks in the smart home sensor communication. This training helps the LSTM model to classify any future attacks. With the acquired intelligence, the model discards the malicious data request d_i'' from the further sensor communication. Contrary, the original data request d_i is forwarded to the destination sensor to accomplish the shared task of SHS. However, it uses the conventional routing protocols to deliver the d_i, without any defensive mechanism, the enroute d_i is prone to severe security threats. Hence, adopting a multi-layer encrypted and anonymous network, i.e., an onion routing network, is inevitable in this layer [20]. Here, the source sensor first connects to the directory server (DS) of the onion routing network to obtain information about the ONs. Then, this information assists him in getting compatible ONs to form an onion routing circuit between the source and destination sensor. The DS is an entity that observes the behavior of all ONs by acquiring information such as bandwidth, time to live (TTL), response time, etc.

$$\mathcal{O} = ON_1, ON_2, \ldots, ON_n \tag{24}$$

$$ON_i \xrightarrow{\text{posses}} \omega_i,\ \chi_i \ and\ \tau \tag{25}$$

where ω_i, χ_i, τ represents bandwidth, TTL and response time of each ON. Based on the received information, the DS sets the thresholds to verify the maliciousness of the ON. Additionally, each ON and DS is connected to the blockchain, where it stores the data, which is decentralized and immutable. It is evident that the attackers can target the DS due to its centralized nature, but the approach is trivial. To justify this, we have taken two scenarios to check the performance of DS in the presence of an attacker. First, if DS and all ONs are connected to the blockchain, all threshold data is directly stored inside the blockchain. The DS voluntarily neglects the ONs whose threshold value is readily fluctuating, showing the malicious behavior. Secondly, if the DS is compromised and the attacker attempts to manipulate the threshold values, the integrity of the original threshold value remains the same. This is because the values are stored inside the immutable blockchain, where it is not possible for anyone to tamper with them. Additionally, the DS is compromised and cannot further ascertain the ONs threshold value; therefore, the smart contract is implemented to verify the malicious ONs and remove them from the onion routing network. The smart contract is, for the time being, until the DS comes back to its normal state because it cannot manage the OR network. After providing the ONs to the source sensor, it

generates the public-private key pair using Elgamal key encryption algorithm to encapsulate the data request d_i in multi-layer encryption. The triple encrypted data request $d_i^{'}$ is represented as,

$$d_i^{'} = \psi_3(\psi_2(\psi_1(d_i))) \tag{26}$$

where ψ_3, ψ_2, ψ_1 are encryption layers at each ON, which is decrypted by his private key $\mathbb{S}\neg$. Each ON in the onion routing network has its own private key and the address of next ON to forward the $d_i^{'}$. This way onion routing network acts as a one-way directed network where each entity only knows the next node information. where, $\mathcal{O}$ is the pool of all onion routing nodes, $dk1$ is the decryption key of each ON, and ζ represents the address of the next ON.

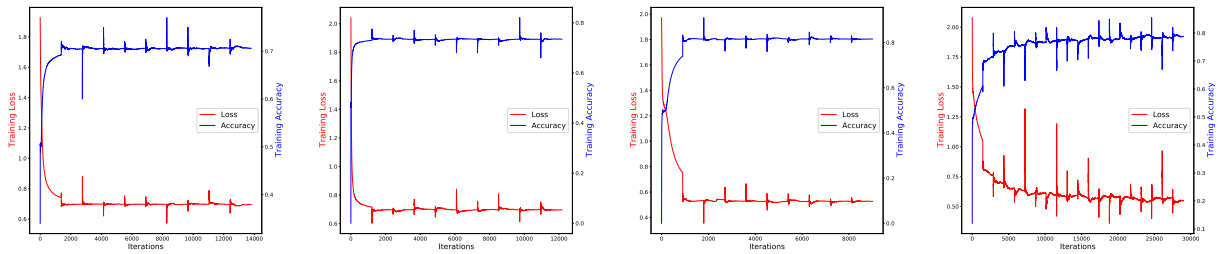
$$\mathcal{O} = ON_1 \xrightarrow{\zeta_2} ON_2 \xrightarrow{\zeta_3} \ldots \xrightarrow{\zeta_k} ON_k \tag{27}$$

where, $\mathcal{O}$ is the pool of all onion routing nodes, and ζ represents the address of the next ON. Hence, backpropagation is challenging, which leads to concrete anonymity in the onion routing network. Likewise, the encrypted data request $d_i^{'}$ is directed to the first ON, i.e., the guard node, where it uses it's private key $\mathbb{S}_3$ to decrypt the first layer of encryption, that is, $\psi_3(d_i)$. Similarly $\psi_2(\psi_1(d_i)$ is decrypted at subsequent ONs. Finally the exit ON, decrypts the last layer of encryption, $\psi_1(d_i)$ to forward the decrypted data request d_i to the destination sensor. Moreover, we want to mention that the exit node of the onion routing gateway is acting as a smart home gateway in the proposed framework. A summarized representation of the aforementioned process is as follows,

$$\begin{aligned}
&\underbrace{\{ON_1, ON_2, \ldots, ON_k\}}_{\text{Onion Nodes}} \xrightarrow{\text{shares } \mathbb{P}_k, \mathbb{S}_k} \\
&\underbrace{s_i(d_i' = \mathbb{P}_1(\ldots(\mathbb{P}_k(d_i)), O_1)}_{\text{Source sensor}} \xrightarrow{\psi_3(\psi_2(\psi_1(d_i)))} \\
&\underbrace{O_1(\mathbb{S}_1, \mathbb{P}_2(\ldots(\mathbb{P}_k(d_i))), O_2)}_{\text{Entry node}} \\
&\xrightarrow{\psi_2(\psi_1(d_i))} \\
&\underbrace{O_{k-1}(\mathbb{S}_{k-1}, \mathbb{P}_{k-1}(d_i), O_k)}_{\text{Intermediate onion node}} \xrightarrow{\psi_1(d_i)} \\
&\ldots \underbrace{O_k(\mathbb{S}_k, d_i, s_j)}_{\text{Last onion node}} \xrightarrow{d_i^{dec}} \underbrace{s_j}_{\text{Destination sensor}}
\end{aligned} \tag{28}$$

The last onion node, i.e. the smart home gateway owns the private key $\mathbb{S}_k$, the decrypted data request d_i^{dec} and the address of destination sensor s_j.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2022.3164715, IEEE Transactions on Network and Service Management



(a) Fridge Sensor (b) Garage Door Sensor (c) Thermostat (d) Weather Sensor

Fig. 6: Loss and accuracy curves for different models which were trained on the respective variants of the dataset.

D. Destination layer

In this layer, the data request d_i^{dec} exited from the smart home onion gateway (exit node) O_k routed to the destination sensor s_j. The single-hop communication between O_k and s_j needs to be secured from the malicious intent. In this regard, we use the ElGamal algorithm to generate and exchange public and private key pairs between O_k and s_j. The destination sensors $\mathcal{S}^d = \{s_1, \ldots, s_j, \ldots, s_S\}$, where $s_i \notin \mathcal{S}^d$. It generates a public $\mathbb{P}_j$ and private $\mathbb{S}_j$ keys. Likewise, each destination senor generates their public keys as $\mathbb{P}_j (1 \leq j \leq \mathcal{S},\ i \notin \mathcal{S}^d)$ and private keys as $\mathbb{S}_j (1 \leq j \leq \mathcal{S},\ i \notin \mathcal{S}^d)$. Then, the s_j exchanges his $\mathbb{P}_j$ with the O_k to encrypt thed_i^{dec}, which is represented as d_i^{enc}. Further, to enhance the secure communication between O_k and s_j we utilize the hash and salt function along with the encrypted d_i^{enc}. The salt function is a random alphanumeric string (Δ) that is adjoin with the d_i^{enc} to obfuscate the attackers.

$$\Gamma = d_i^{enc} + \Delta \tag{29}$$

where, Γ represents the salted data request. Then, we have used the MD5 algorithm for hashing, where each character of salted data request Γ is converted into a suitable binary number (ξ).

$$\xi = \xi_i | \ \Gamma \tag{30}$$

where ξ_i is the binary number of each character of Γ. We have calculated the length of ξ and added it to the padded bits to produce $\{\eta_1, \eta_2, \ldots, \eta_i\} \in \Gamma$, where each $\eta_i \in 512$ bit blocks. Moreover, the MD5 algorithm has an iterative approach where each η_i is concatenated with its four initialized buffers to form a hashed message $\mathcal{H}(\Gamma)$. The inclusion of hash and salt function

makes the data request d_i more robust against integrity attacks. The $\mathcal{H}(\Gamma)$ received at s_j using a single hop, yet encrypted communication.

$$\mathcal{H}(\Gamma) \xrightarrow{\text{reaches}} s_j \tag{31}$$

$$\mathcal{H}(\Gamma) = \mathcal{H}(\Gamma^{'}) \tag{32}$$

$$d_i^{enc} = \Gamma - \Delta \tag{33}$$

The s_j calculates the hash as $\mathcal{H}(\Gamma^{'})$ at his side that must satisfy the Eq. (32) to absolve from the data integrity attacks. Next, the salt string (Δ) has to be removed from the salted data request Γ to get the encrypted data request d_i^{enc} as represented in Eq. (33). Finally, the s_j uses his private key $\mathbb{S}_j$ to decrypt the d_i^{enc} to get original data request d_i from source sensor s_i.

IV. Result Discussion

This section details down the experimentally LSTM based generated results for the proposed architecture. These results are obtained by applying various common performance measures. Further, it also discusses the result of the security layer and its implications.

A. AI-based Results

The performance of the AI layer specifically can be tested using various evaluation metrics. Common measures like F1-score and accuracy scores are considered. Loss and accuracy curves are also plotted to visualize the overall convergence of the models. As discussed in the proposed architecture, the dataset consists of six different variants of incoming sensor readings. For each variant of the sensor dataset coming from different appliances, a model is trained and used for prediction.

Mathematical notations for F1 and accuracy scores are expressed as below. We have used different acronyms to represent the equations mathematically. Acronym TP, FP, and FN stands for true positives, false positives and false negatives, respectively. Average F1 and accuracy scores are computed as follows.

$$\mu^s_{precision} = \frac{\sum\limits_{C_s} \frac{TP_{c'}}{TP_{c'}+FP_{c'}}}{C_s} \qquad \mu^s_{recall} = \frac{\sum\limits_{C_s} \frac{TP_{c'}}{TP_{c'}+FN_{c'}}}{C_s}$$

$$\mu^s_{F1} = \frac{2 \times \mu_{precision} \times \mu_{recall}}{\mu_{precision} + \mu_{recall}} \tag{34}$$

$$Accuracy_s = \frac{\sum_{C_s} TP}{\sum_{C_s} TP + FP + FN} \tag{35}$$

Where C_s signifies the collection of total unique classes for a given sensor s_i such that $\forall c' \epsilon C_s$. $\mu^s_{precision}$ and μ^s_{recall} respectively signifies the mean precision and recall for a given IoT sensor s_i with total classes C_s. The purpose of all the individually trained neural networks over the separate sensor data is to increase the overall mean accuracy and F1-score . Higher values of these scores allows security layer to efficiently block malicious requests.

Fig. 6 shows the loss and training curves for the separately trained LSTM architectures. Loss vs iterations and accuracy vs iterations depicts the overall convergence of the model during training. As each dataset variant has different training examples, total iterations to achieve convergence also varies. It is important to note that, all the data variants were trained on a constant models architecture, optimizers, batch size and epochs. It can be observed that the loss value for each model plateaus out at the end of training showing the point of convergence has been reached.

TABLE III: Sensor wise model F1 and accuracy scores for the trained prediction models

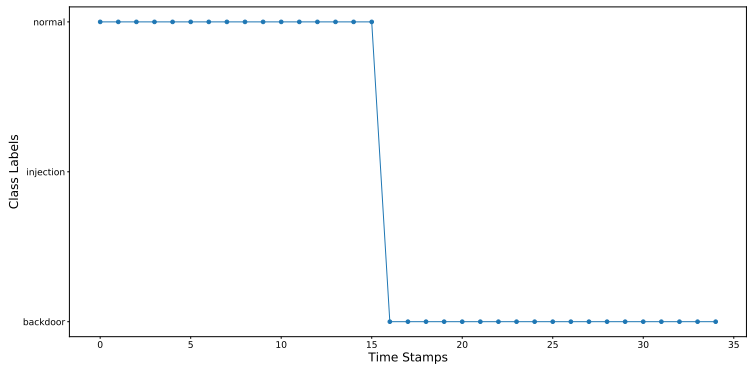
Sensor Type	F1-Score	Accuracy
Fridge Sensor	0.62	0.70
Garage Door	0.64	0.73
Thermostat	0.74	0.81
Weather Sensor	0.84	0.85

TABLE III shows the generated F1 and accuracy scores for the proposed LSTM based architectures. These results are obtained on the individual test sets of the individual preprocessed sensor data. Accuracy and F1 scores provides overall insight on model, by giving a fair outlook on their training performance.

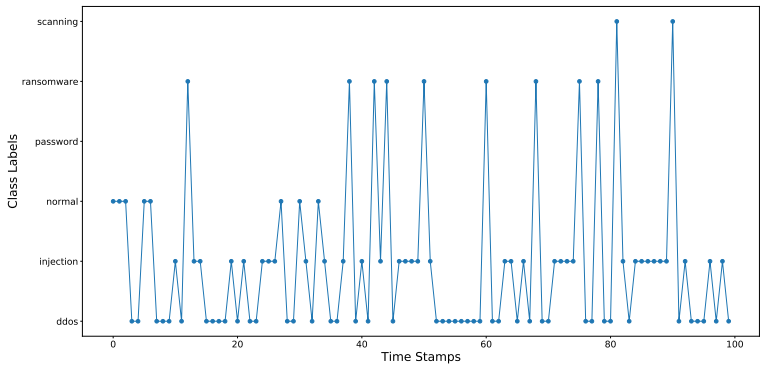
B. Onion Network Performance Evaluation

The onion routing network in the proposed framework is created inside the shadow simulator, which is an open-source network simulator to implement and execute applications like onion routing and blockchain. We simulated the onion routing topology inside the shadow simulator to test the performance of malicious ONs against our proposed framework. We configure the ONs

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2022.3164715, IEEE Transactions on Network and Service Management



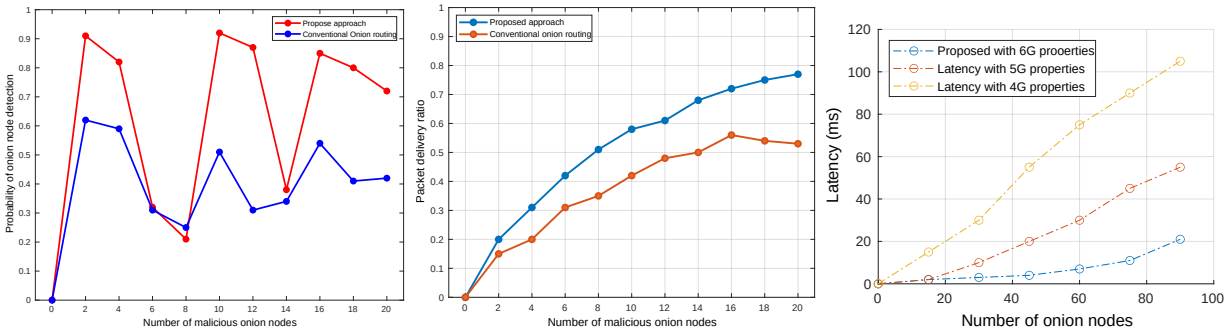
(a) Thermostat sensor reading predictions



(b) Garage Door sensor reading predictions

Fig. 7: Attack prediction on trained individual sensor models at different timestamps.

topology by manually specifying the parameter values to the ω_i, χ_i, τ for each ON. Then, the ONs utilize these parameters to process the message in the onion routing. Hence, the parameter values are important to understand the behavior of an ON. To extract the values of ω_i, χ_i, τ, we will use the log files maintained by each ON in the simulator. Next, we will set the threshold for each parameter of the ON based on their best performance. Then, we presented a few attacks, such as DoS and flooding, which directly target the resources of the ONs; this results in fluctuation in their parameters threshold values. Formally, the DS is the sole authority that manages the ONs by observing their threshold values. However, if the DS is compromised, it neglects to check the malicious ONs, which breaks the anonymity. Therefore, we stored these threshold values inside the immutable blockchain that preserves anonymity. The problem persists if a few smart



(a) Malicious ON detection rate (b) Packet loss rate (c) Latency comparison

Fig. 8: Loss and accuracy curves for different models which were trained on the respective variants of the dataset.

ONs are subtle and not showing any malicious behavior, even though they are malicious. It is a challenging task for a DS to find such ONs from the onion routing network.

1) Packet Delivery Ratio: Fig. 8b illustrates the performance of the security layer of the proposed framework. It shows the probability of the packet delivery in the presence of malicious ON in the onion routing network. As the number of malicious nodes increases, the packet delivery ratio of conventional onion routing decreases gradually. However, due to the involvement of blockchain, we can adroitly remove the malicious ONs from the onion routing and enhance the security of the proposed framework. Once the malicious node removes from the onion routing, it is evident that the packets reach successfully to the destination. Therefore, the packet delivery ratio of the proposed onion routing outperforms conventional onion routing.

2) Malicious ON Detection Rate: Fig. 8a shows the performance of the onion routing-based security layer against the detection rate of the malicious ON. From the graph, we can mention that our proposed framework seamlessly detects the malicious ON, even when the DS is compromised. This is because the threshold values are stored inside the immutable blockchain. Further, we have used smart contracts that can utilize the values to detect the malicious ONs, until the DS recovers from the attack. We also want to remark that in the scenario where ONs impersonate their malicious behavior, the proposed security layer detects them and performs equivalently to the conventional onion routing

3) Latency Comparison: The latency in the proposed onion routing-based framework has significantly reduced with the integration of 6G communication network. Fig. 8c shows the latency

comparison of the proposed framework by considering different latency properties based on 4G, 5G, and 6G networks. It shows that the proposed framework with 6G network outperformed compared to other networks. The consideration of 6G channel is between the onion nodes as well as the exit node and destination sensor s_j.

V. Conclusion

In this paper, we presented deep learning and onion routing-based secure sensor communication for smart home systems by adopting blockchain and a 6G network interface. We first discussed sensor communication and its intricacy regarding security in the SHS. The sensor communication is passed through a collaborative intelligence, where it is first forwarded to the LSTM model and then to the onion routing network. The LSTM classifies the sensor's malicious and non-malicious requests, and only non-malicious requests are passed to the onion routing network; this satisfies the green communication in the smart home. Further, in the onion routing, we involved blockchain technology to store the threshold of each ONs parameter to enhance its security and anonymity. Once the message exited from the onion routing network, we utilized Elgamal cryptosystem with hash and salt functions to achieve the end-end security. The proposed framework is assessed against deep learning metrics such as F1- score and accuracy. Additionally, the onion routing is simulated inside the shadow simulator to evaluate the detection rate of the malicious ONs and packet delivery ratio. Finally, we want to mention that in future work, we will improve the performance of the proposed onion routing in the presence of impersonating malicious ONs.

Acknowledgement

Taif University Researchers Supporting Project number (TURSP-2020/126), Taif University, Taif, Saudi Arabia

References

[1] G. S. Poh, P. Gope, and J. Ning, "PrivHome: Privacy-Preserving Authenticated Communication in Smart Home Environment," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1095–1107, 2021.

[2] J. Liu, C. Zhang, and Y. Fang, "EPIC: A Differential Privacy Framework to Defend Smart Homes Against Internet Traffic Analysis," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1206–1217, 2018.

[3] T. Song, R. Li, B. Mei, J. Yu, X. Xing, and X. Cheng, "a privacy preserving communication protocol for iot applications in smart homes," *IEEE Internet of Things Journal*.

[4] M. Backes, I. Goldberg, A. Kate, and E. Mohammadi, "Provably Secure and Practical Onion Routing," in *2012 IEEE 25th Computer Security Foundations Symposium*, pp. 369–385, 2012.

[5] R. Gupta, S. Tanwar, and N. Kumar, "B-IoMV: Blockchain-Based Onion Routing Protocol for D2D Communication in an IoMV Environment beyond 5G," *Veh. Commun.*, vol. 33, jan 2022.

[6] M. Sayad Haghighi and Z. Aziminejad, "Highly Anonymous Mobility-Tolerant Location-Based Onion Routing for VANETs," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2582–2590, 2020.

[7] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous connections and onion routing," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 482–494, 1998.

[8] K. Kita, Y. Koizumi, T. Hasegawa, O. Ascigil, and I. Psaras, "Producer Anonymity Based on Onion Routing in Named Data Networking," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 2420–2436, 2021.

[9] K. Sakai, M.-T. Sun, W.-S. Ku, J. Wu, and F. S. Alanazi, "Performance and Security Analyses of Onion-Based Anonymous Routing for Delay Tolerant Networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 12, pp. 3473–3487, 2017.

[10] W. Li, T. Logenthiran, V.-T. Phan, and W. L. Woo, "A Novel Smart Energy Theft System (SETS) for IoT-Based Smart Home," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5531–5539, 2019.

[11] A. Agiollo, M. Conti, P. Kaliyar, T.-N. Lin, and L. Pajola, "DETONAR: Detection of Routing Attacks in RPL-Based IoT," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1178–1190, 2021.

[12] K. Li, W. Ni, M. Abolhasan, and E. Tovar, "Reinforcement Learning for Scheduling Wireless Powered Sensor Communications," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 2, pp. 264–274, 2019.

[13] A. Qashlan, P. Nanda, X. He, and M. Mohanty, "Privacy-Preserving Mechanism in Smart Home Using Blockchain," *IEEE Access*, vol. 9, pp. 103651–103669, 2021.

[14] Q. Yang and H. Wang, "Privacy-Preserving Transactive Energy Management for IoT-Aided Smart Homes via Blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11463–11475, 2021.

[15] G. Chen, J. Wu, W. Yang, A. K. Bashir, G. Li, and M. Hammoudeh, "Leveraging Graph Convolutional-LSTM for Energy-Efficient Caching in Blockchain-Based Green IoT," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1154–1164, 2021.

[16] "Ton iot dataset." https://research.unsw.edu.au/projects/toniot-datasets. Accessed: 2021.

[17] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON IoT datasets," *Sustainable Cities and Society*, vol. 72, p. 102994, 2021.

[18] M. Ma and Z. Mao, "Deep-convolution-based LSTM network for remaining useful life prediction," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1658–1667, 2020.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[20] R. Gupta, M. M. Patel, S. Tanwar, N. Kumar, and S. Zeadally, "Blockchain-Based Data Dissemination Scheme for 5G-Enabled Softwarized UAV Networks," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 4, pp. 1712–1721, 2021.

Nilesh Kumar Jadav is a Full-Time Ph.D. Research Scholar in the Computer science and Engineering Department at Nirma University, Ahmedabad, Gujarat, India. He received his Bachelor as well as Master of Technology from the Gujarat Technological University (GTU), Gujarat, India in 2014 and 2018, respectively. He has authored/co-authored publications (including papers in SCI Indexed Journal and IEEE ComSoc sponsored International Conference). Some of his research findings are published in top-cited journals and conferences such as IEEE INFOCOM, IEEE ICC, IJCS and many more. His research interest includes, Artificial Intelligence, Network security, 5G Communication Network, Blockchain Technology. He is also an active member of ST Research Laboratory (www.sudeeptanwar.in).

Rajesh Gupta (*Student Member, IEEE*) is a Full-Time Ph.D. Research Scholar in the Computer science and Engineering Department at Nirma University, Ahmedabad, Gujarat, India. He received his Bachelor of Engineering in 2008 from the University of Jammu, India and Master's in Technology in 2013 from Shri Mata Vaishno Devi University, Jammu, India. He has authored/co-authored some publications (including papers in SCI Indexed Journals and IEEE ComSoc sponsored International Conferences). Some of his research findings are published in top-cited journals and conferences such as IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, IEEE NETWORK MAGAZINE, IEEE IOT MAGAZINE, COMPUTER COMMUNICATIONS, COMPUTER AND ELECTRICAL ENGINEERING, IJCS WILEY, ETT WILEY, PHYSICAL COMMUNICATION, IEEE ICC, IEEE INFOCOM, IEEE GLOBECOM, IEEE CITS, and many more. His research interest includes Device-to-Device Communication, Network Security, Blockchain Technology, 5G Communication Network, and Machine Learning. He is also a recipient of Doctoral Scholarship from the Ministry of Electronics and Information Technology, Govt. of India under the Visvesvaraya Ph.D. Scheme. He is a recipient of student Travel Grant from WICE-IEEE to attend IEEE ICC 2021 held in Canada. He has been awarded best research paper awards from IEEE ECAI 2021, IEEE ICCCA, and IEEE IWCMC 2021. His name has been included in the list of Top 2% scientists worldwide published by the Stanford university, USA. He was felicitated by Nirma University for their research achievements in 2021. He is also an active member of ST Research Laboratory (www.sudeeptanwar.in). He is a student member of IEEE since 2018.

Dr. Mohammad Dahman Alshehri is an Assistant Professor at Computer Science Department, Taif University, Saudi Arabia and Visiting Professor at School of Computer Science at the University of Technology Sydney (UTS), Australia. He received his PhD in Artificial Intelligence of Cybersecurity for Internet of Things (IoT) from the University of Technology Sydney, Australia. He developed 6 smart novel algorithms for IoT to reinforcement Cybersecurity with AI that be able to detect the various behaviours of cyber-attacks and provide full secure and protection platform for the IoT from the most harm cyber-attacks. Furthermore, he published several publications in high ranked international journals, top-tier conferences and chapter of books, also he received number of international and national awards and prizes. His main current research interest lies in the areas of Cybersecurity, Artificial Intelligence, Internet of Things (IoT), Trust and Reputation

Harsh Mankodiya is currently pursuing Bachelors of Technology in Computer Science Engineering at the Institute of Technology, Nirma University, Ahmedabad, Gujarat, India, and is anticipated to graduate in the year 2023. He is also an undergraduate researcher at Sudeep Tanwar's Research Group (www.sudeeptanwar.in), Nirma University. His research interests mainly lie in the field of Machine Learning, Deep Learning, and Artificial Intelligence.

Sudeep Tanwar (*Senior Member, IEEE*) is currently working as a Professor with the Computer Science and Engineering Department, Institute of Technology, Nirma University, India. He is also a Visiting Professor with Jan Wyzykowski University, Polkowice, Poland, and the University of Pitesti in Pitesti, Romania. He received B.Tech in 2002 from Kurukshetra University, India, M.Tech (Honor's) in 2009 from Guru Gobind Singh Indraprastha University, Delhi, India and Ph.D. in 2016 with specialization in Wireless Sensor Network. He has authored two books and edited 13 books, more than 250 technical articles, including top journals and top conferences, such as IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, IEEE WIRELESS COMMUNICATIONS, IEEE NETWORKS, ICC, GLOBECOM, and INFOCOM. He initiated the research field of blockchain technology adoption in various verticals, in 2017. His H-index is 45. He actively serves his research communities in various roles. His research interests include blockchain technology, wireless sensor networks, fog computing, smart grid, and the IoT. He is a Final Voting Member of the IEEE ComSoc Tactile Internet Committee, in 2020. He is a Senior Member of IEEE, Member of CSI, IAENG, ISTE, and CSTA, and a member of the Technical Committee on Tactile Internet of IEEE Communication Society. He has been awarded the Best Research Paper Awards from IEEE IWCMC-2021, IEEE GLOBECOM 2018, IEEE ICC 2019, and Springer ICRIC-2019. He has served many international conferences as a member of the Organizing Committee, such as the Publication Chair for FTNCT-2020, ICCIC 2020, and WiMob2019, a member of the Advisory Board for ICACCT-2021 and ICACI 2020, a Workshop Co-Chair for CIS 2021, and a General Chair for IC4S 2019, 2020, and ICCSDF 2020. He is also serving the editorial boards of Computer Communications, International Journal of Communication System, and Security and Privacy. He is also leading the ST Research Laboratory, where group members are working on the latest cutting-edge technologies.

Neeraj Kumar (*Senior Member, IEEE*) received the Ph.D. degree in CSE from Shri Mata Vaishno Devi University, Katra, Jammu and Kashmir, India. He was a Postdoctoral Research Fellow with Coventry University, Coventry, U.K. He is currently a Full Professor with the Department of Computer Science and Engineering, Thapar University, Patiala, Punjab, India. He is leading the research group Sustainable Practices for Internet of Energy and Security (SPINES), where group members are working on the latest cutting-edge technologies. He is a Visiting Professor at Coventry University. He has published more than 300 technical research papers in leading journals and conferences from IEEE, Elsevier, Springer, and John Wiley. Some of his research findings are published in top cited journals, such as the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, IEEE NETWORK, IEEE Communications Magazine, IEEE WIRELESS COMMUNICATIONS, the IEEE INTERNET OF THINGS JOURNAL, the IEEE SYSTEMS JOURNAL, FGCS, JNCA, and ComCom. He has guided many Ph.D. and M.E./M.Tech. His research is supported by fundings from Tata Consultancy Service, the Council of Scientific and Industrial Research (CSIR), and the Department of Science and Technology. He is a TPC member and a reviewers of many international conferences across the globe. He has awarded best research paper awards from IEEE ICC 2018 and the IEEE SYSTEMS JOURNAL 2018.