Blockchain-based Privacy Preservation Scheme for Misbehavior Detection in Lightweight IoMT Devices

Rahmadika, Sandi; Astillo, Philip Virgil; Choudhary, Gaurav; Duguma, Daniel Gerbi; Sharma, Vishal; You, Ilsun

Published in:
IEEE Journal of Biomedical and Health Informatics

Link to article, DOI:
10.1109/JBHI.2022.3187037

Publication date:
2022

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Blockchain-based Privacy Preservation Scheme for Misbehavior Detection in Lightweight IoMT Devices

Sandi Rahmadika, Philip Virgil Astillo, Gaurav Choudhary, Daniel Gerbi Duguma, Student Member, IEEE, Vishal Sharma, Senior Member, IEEE, and Ilsun You, Senior Member, IEEE

Abstract—The Internet of Medical Things (IoMT) has risen to prominence as a possible backbone in the health sector, with the ability to improve quality of life by broadening user experience while enabling crucial solutions such as near real-time diagnosis. However, privacy and security problems remain largely unresolved in the safety area. Various rule-based methods have been considered to recognize aberrant behaviors in IoMT and have demonstrated high accuracy of misbehavior detection appropriate for lightweight IoT devices. However, most of these solutions have privacy concerns, especially when giving context during misbehavior analysis. Moreover, falsified or modified context generates a high percentage of false positives and, in some cases, causes a by-pass in misbehavior detection. Relying on the recent powerful consolidation of Blockchain and federated learning (FL), we propose an efficient privacy-preserving framework for secure misbehavior detection in lightweight IoT devices, particularly in the artificial pancreas systems (APS). The proposed approach employs privacy-preserving bidirectional long-short term memory (BiLSTM) and augments the security through the integration of Blockchain technology based on Ethereum smart contract environment. Furthermore, the effectiveness of the proposed model is benchmarked empirically in terms of sustainable privacy preservation, commensurate incentive scheme with an unretraceability feature, exhaustiveness, and the compact results of a variant neural network approach. As a result, the proposed model has a 99.93% recall rate, showing that it can detect virtually all possible malicious events in the targeted use case. Furthermore, given an initial ether value of 100, the solution’s average gas consumption and Ether spent are 84,456.5 and 0.03157625, respectively.

Index Terms—Blockchain, federated learning, Internet of Medical Things (IoMT), misbehavior detection, privacy preservation, smart contract

I. INTRODUCTION

Along with the exponential growth of Internet of Things (IoT) in the medical industry, new security problems are continually emerging. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2020R11A2A07389303).

Corresponding Author: Ilsun You
*These authors contributed equally to this work.
Sandi Rahmadika is with the Department of Electronic Engineering, State University of Padang (UNP), Kota Padang, Sumatera Barat 25171, Indonesia. (email: sandi@ft.unp.ac.id)
Philip Virgil Astillo is with the Department of Computer Engineering, University of San Carlos, Go, M. Cuenco Ave., Talamban, Cebu City 6000, Philippines. (email: pvbastillo@usc.edu.ph)
Gaurav Choudhary is with the Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU), Richard Petersens Plads, 322, 115, 2800 Kgs. Lyngby, Denmark. (email: gauravchoudhary777@gmail.com).
Daniel Gerbi Duguma is with the Department of Information Security Engineering, Soochunhyang University, Asan, 31538, South Korea (email: danielgerbi2005@gmail.com).
Vishal Sharma is with the School of Electronics, Electronic Engineering and Computer Science, Queen’s University Belfast (QUB), Belfast, UK. (email: v.sharma@qub.ac.uk).
Ilsun You is with the Department of Information Security, Cryptology, and Mathematics, Kookmin University, Seoul, 02707, South Korea. (email: ilsunu@gmail.com).

while existing security threats become more apparent. Additionally, privacy issues that may arise from the various data collected from IoT devices can negatively impact users. Particularly, these collected data may be used to determine the users’ location, ID, role, and other information, posing a valid concern about data privacy and security. Data access is a crucial aspect of realizing a secure system by controlling who and what associations have access to such information, especially because IoT devices store different sensitive information in their memory or their controlling gadgets. IoT security turns into an essential subject since it manages sensitive information that streams over the Internet or network, and IoT security governs protection, privacy, and trust during execution [1–3]. In any case, the greatest challenge is a trade-off between the performance of IoT devices and security solutions. Privacy concerns apply in a variety of contexts, including when exchanging personal information, privacy in communications, and privacy in sensitive data records. Irrespective of multiple privacy domains, the confidentiality of personal information [4], behavior, and communication are the most significant dimensions in the IoT context. Thus, privacy should be protected by enabling IoT device management rather than relying on consumers’ side. Unnecessary access control should be avoided, and all data should be verified as there are circumstances where a single vulnerability can be exploited owing to assumptions made at component interfaces in misbehavior detection approaches [5–7].

Misbehavior detection in IoT devices is a challenging task given their severely constrained computing, communication, power, and other resources. To make matters worse, data transfer between IoMT devices and monitoring agents is inconsistent and vulnerable to insider assaults. As a result, IoMT devices must be more secure and sensitive in terms of misbehavior detection, because altered data increases false-positive rates and decreases the effectiveness of misbehavior detection approaches [19], [20]. Despite the existence of numerous communication protocols that address the issues related to trust and security of context during message exchanges [21], there is still a deficiency concerning privacy preservation mechanisms [22], [23]. Blockchain-based privacy protection is an alternative solution that has recently gained popularity in the domain of IoT context sharing. [24], [25].

Blockchain is inherently tamper-proof and does not require the presence of a middleman during transactions, making it a feasible solution to various problems in the IoT ecosystem. Blockchain is used extensively in healthcare applications to provide the security, transparency, and immutability of data records via autonomous contracts [26]. Transparency properties and tamper-proof records can be achieved by straightforwardly adopting Blockchain platforms such as Ethereum (smart contract) and Hyperledger Fabric (chain code). However, for the application of private data that is confidential, transparency needs to be considered further. This is the most important aspect of incorporating Blockchain into our research. As a result, we use the Ethereum smart contract to control how data is kept and accessed safely, in conjunction with our security measures built inside the contracts. In addition, we also exert the merits of the smart contract feature as an incentive mechanism for the data provider. Our
Table I

<table>
<thead>
<tr>
<th>Authors</th>
<th>Category</th>
<th>Scheme</th>
<th>Mechanism</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Astild et al. [8]</td>
<td>Behavior-based IDS</td>
<td>Trust Management</td>
<td>Smoothened-trust-based scheme</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Sedjelmaci et al. [9]</td>
<td>UAV-IDS</td>
<td>Hierarchical Detection Scheme</td>
<td>Combines rules-based detection</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Sedjelmaci et al. [10]</td>
<td>UAV-IDS</td>
<td>Ejection framework against lethal attacks</td>
<td>Use Bayesian game Model for detection</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Choudhary et al. [11]</td>
<td>IoT-IDS</td>
<td>Lightweight misbehavior detection in Medical IoT</td>
<td>Formally verified Behavior Rule based IDS</td>
<td>No</td>
<td>Yes</td>
<td>Yes*</td>
<td>Yes*</td>
<td>No</td>
</tr>
<tr>
<td>Khan et al. [12]</td>
<td>Behavior-based IDS</td>
<td>Secure ICSS based on the behavior of their computational resources</td>
<td>Detect DFI attacks</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Sharma et al. [13]</td>
<td>Behavior-based IDS</td>
<td>Fuzzy based HC-APN</td>
<td>Detect zero day attacks</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Jokar et al. [14]</td>
<td>Specification-based IDS</td>
<td>IDS for Home Area Networks</td>
<td>Used feature space for IDS targeting the IEEE 802.15.4 standard covering the PHY and MAC layers of the ZigBee technology</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Salem et al. [15]</td>
<td>IoT-IDS</td>
<td>Anomaly Detection in Wireless Body Area Networks for Reliable Healthcare Monitoring</td>
<td>The approach is based on Haar wavelet, Non-Seasonal Holt-Winters (NSHW)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Dwivedi et al. [16]</td>
<td>Data sharing privacy in healthcare IoT</td>
<td>Privacy-Preserving Healthcare</td>
<td>Modified block chain models</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Saeed et al. [17]</td>
<td>IoT-IDS</td>
<td>Intelligent Intrusion Detection in Low-Power IoTs</td>
<td>Intelligent secure architecture using random neural networks (RNNs)</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Zhu and Yu [18]</td>
<td>Sensor data privacy</td>
<td>Privacy-Preserving in Deep Learning</td>
<td>Privacy of the data used for learning a model or as input to an existing model</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes*</td>
<td>No</td>
</tr>
<tr>
<td>Loukil et al. [19]</td>
<td>Data Privacy in IoT</td>
<td>Privacy-preserving IoT device management framework</td>
<td>Judging the misbehavior of the smart device and determines the corresponding penalty</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

II. Related Work

The healthcare industry has benefited from various developments such as IoTs, sensors, and organized contraptions, in addition to portable and electronic applications. However, the security, privacy, and assurance of gadgets continue to be an open concern. Patients’ privacy and safety may be threatened owing to a lack of a systematic and accurate line of defense against a wide range of security threats from sensitive data breaches via malware/viruses to more serious life-threatening injuries. The shortcomings in existing medical Cyber-Physical Systems (MCPS) security make it unbearable for the medical services sector and healthcare providers to verify and ensuring gadgets.

There are several important research works related to misbehavior detection in medical IoT that address diverse issues such as security, privacy, dependability, and attack detection techniques. Meng et al. [27], for instance, focused on trust-based interruption identification using social profiling and used Euclidean distance between two behavioral profiles. Celdrán et al. [28] emphasized the current MCPS security problems and presented Virtual Medical Device (VMD). In the paper, the mobile edge and fog computing models are employed to maintain an automated and reasonable framework used by Network Function Virtualization (NFV) and Software-Defined Network (SDN) procedures to enable a consistent association of MCPS security.

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on July 28,2022 at 10:29:13 UTC from IEEE Xplore. Restrictions apply.
Blockchain-based privacy preservation in IoT devices has emerged as a prominent solution, as shown in Figure 1. The figure illustrates the essence of using Blockchain technology in various use cases. With this regard, researchers like Loukil et al. [19] proposed a privacy-preserving IoT device management framework by highlighting data privacy. The proposed solution judges the misbehavior of intelligent devices and determines the corresponding penalty. Zhu and Yu [18] designed a privacy-preserving scheme leveraging Deep Learning on sensor data. In this scheme, the privacy of the data is used for learning a model or as input to an existing model. Dwivedi et al. [16] proposed a system of adjusted Blockchain models to fit for IoT devices. The authors constructed several supplementary cryptographic primitives' protocols to tackle the drawbacks in IoT applications running on a top Blockchain-based network. A similar objective was introduced by Kuo et al. [33] that combines multiple techniques such as level-wise model learning, Blockchain, and a new consensus algorithm for the model ensemble to preserve privacy modeling on the distributed ledger. Furthermore, various other researches (such as [34]-[36]) make use of FL to enhance privacy in different application areas. The existing state-of-the-art Blockchain-based privacy preservation and IoT misbehavior detection are shown in Table II.

### III. Core System Components and Models

#### A. Architectural Framework and The essence of Decentralized Approach

In this research, Blockchain-based privacy preservation techniques are merged with lightweight IoT devices to secure malicious behavior detection over the wireless network. Our system is designed for the insulin pump case with the respective controller to continuously monitor patient glucose levels within a specific time. It is well known as a compact medical system called continuous glucose monitors (CGM). The communication is conducted through a distributed information technology and edge computing manner. In other words, the user's data is managed at the periphery of the network (closer to the originating source). We leverage a variant of recurrent neural network (RNN), namely bidirectional long-short term memory (BiLSTM). On the other hand, Blockchain technology can provide an immutable data record with several cryptography protocols embedded into smart contracts. The Ethereum platform also supports the decentralized revenue mechanism for the data owners. The overall overview of our suggested method is depicted in Figure 1II-A. In the diagram, each APS controller collects critical information from the CGM and Insulin Pump, such as blood glucose level and remaining insulin quantity. Normally, the controller would be a smartphone that communicates with the Blockchain layer and the aggregator through a cellular network or WiFi network. The controller is made up of several components, such as the lightweight deep-learning framework (LDLF), MDS, data storage, controller algorithms, and so on. Instead of sending it immediately to the verifiable aggregator model provider, the private information held in the controller is utilized for training the private information revealed in the controller. The Blockchain-enabled aggregator after the Blockchain validates the aggregated weight parameters, they are averaged and sent back to each of the participating APSs. Further details of the entire process are described in the subsequent subsections. Meanwhile, it is important to note that the proposed framework can be adapted to all subdomain of IoMT, but each subdomain may have a specific structure of the deep learning model that is more suitable for the environment.

One of the Blockchain merits that need to be considered in this research is the transparency properties that are inherent in the decentralized approach. Every entity that is incorporated in a Blockchain network can access information or transaction records. This transparency trait is beneficial in some cases. Still, it is not desirable for some scenarios, for instance, any systems that manage sensitive data [37]. Some examples of sensitive data are private information revealing racial or ethnic origin, religious or philosoph-
ical beliefs, sexual orientation, health-related data, etc. Therefore, Blockchain-based applications require several additional dynamic protocols. By doing so, the system can obscure the information stored in the database. Transparency properties are inherent to Blockchain-based applications. However, the users can manage whether certain features are open or hidden from the public. In our cases, the log of transactions is available to every member. However, the actual identity or data remains secret since we construct several protocols embedded into the system. We detail the decentralized privacy-preserving protocols in Section III-C.

Apart from the transparency concern inherent in the Blockchain system, several other matters are listed in Table II that describe the taxonomy of concerns in the Ethereum smart contract through Ethereum Virtual Machine (EVM) [38]. The sequence of instructions developed into Ethereum smart contracts may have security concerns caused by internal errors such as program defects. A tool called OYENTE [39] exposes four varieties of potential security bugs in the smart contract in 2016 (8833 out of 19366 Ethereum smart contracts are vulnerable). Those security bugs are as follows: transaction-ordering dependence, timestamp sequence, mishandled exceptions, reentrancy in input issues. Nevertheless, the smart contract improves in many aspects, including security, making it more sophisticated compared to various versions.

B. Federated Learning (FL)-based Misbehavior Detection

The massive proliferation of the Internet of Things (IoT) has opened more opportunities for an adversary to compromise, especially by exploiting newly discovered vulnerabilities, of a target system. This situation is critical, particularly in the healthcare industry where the integration of IoT for remote medical services is increasing [40].

Among the variety of promising solutions, the network MDS showed high efficiency and effectiveness in detecting malicious actions of devices as the consequence of both internal and external attacks [8], [11], [41]. This scheme requires careful analysis of a large number of operational data, or the solution provider must possess extensive domain knowledge to differentiate benign network events from malicious ones. Accordingly, this is done by comparing the runtime data with a model, which represents the operational behavior of the target system. In this case, proper construction of the model is vital in achieving high detection accuracy. In turn, considering the diversity of data in the healthcare industry supported by the internet-connected wearable sensors and devices, the machine learning technique offers the best means to build the most appropriate model for network behavior classification.

Machine learning deals with building models from a large amount of collected data with minimal to no human intervention. In the healthcare industry, most of this data contains sensitive personal health information of the patients. FL, an unconventional learning paradigm, is desirable and appropriately augments privacy-preservation of the patients since training data stays within the digital space of the owner and global model is built out of sub-models, which are trained locally by participating devices. This paper proposes a deep neural network-based MDS trained under an FL paradigm as applied to the APS.

In this case, APS controllers participate in the model-building process at a predefined iteration or communication round with the server, which serves as the aggregator. At every round, the participating devices receive the global model (GM) from the server and train it using their locally stored data. Subsequently, all devices submit their trained sub-models back to the server for aggregation using equation [1] in which the updated parameters of the global model, carried out to the next round, is the weighted average. The weight is defined by the cardinality of respective local dataset stored in device $i$ over the cardinality of the union of local datasets $(\bigcup D_i)$. Afterwards, the next round starts wherein the server distributes the aggregated model back to the devices.

$$GM ← \sum_{i=1}^{P} \frac{|D_i|}{|\bigcup D_i|} \times W_i$$

where $W_i : \text{(weights, bias) of device } i$

$D_i : \text{local dataset stored at device } i$

$P: \text{number of participating devices}$

To illustrate the concept of FL and show its advantages in misbehaviour detection, we used state-based modelling where states are connected to each other via weights. Each local device can generate state machines that will get updated via weights only, say, $w$. Based on the concept of FL, only weights get transferred, and the number of states cannot be predicted by an adversary trying to dodge the detection or prevent misbehaviour from getting caught. With $k$ number of states, an adversary would have to check $\frac{k(k+1)}{2}$ transitions in a given span of time, which by the property of determinism cannot be completed effectively. Thus, maintaining the system’s privacy and contents anonymous.

If $S_k$ is the set of states for $i$th device such that $S_k = \{S_{1,i}, S_{2,i}, \ldots, S_{k,i}\}$ having n number of states and $W_i$ be the set of weights for the $i$th device, such that $(S_{m,i}, S_{k,i})$ are the transitions connecting the two states, $m$ and $n$, with weight $W_{mk,i}$, then the task is to effectively offload the weights and share them across devices through aggregation before the adversary can identify the sequence of transitions to avoid misbehaviour detection. An interesting observation is that if the state machines have near to complete connected graph, the attack prediction for an adversary will increase, which can be prevented by considering a secondary constraint, i.e., keeping the number of states too high than the number of transitions to delay the adversaries from avoiding the detection. Thus, if $\tau$ is the time for updating the weights, and the prediction rate of states is $\theta$, then the number of states that can be predicted within $\tau$ should be $\leq \frac{\theta}{\text{transitions}}$ for the $i$th device. This situation for state-based FL allows understanding the performance of the system by exploiting the model through an appropriate game theory where the situation between the adversaries and the honest device can be evaluated subject to the properties of FL.

To model this, let $L(\mathcal{N}) \geq |\mathcal{M}|$ be the function representing convergence cost as the divisibility of the weights across different state machines based on the properties of FL for $k$ states of honest device and $m$ states of adversarial device. Here, $L(\mathcal{N}) \geq |\mathcal{M}|$ must be followed such that updates of the states and weights should be done before the adversary can predict the weights and transitions across the states leading to the high possibility of attacks. To understand such a situation, we rely on the Stackelberg game [42] formation between the devices and the adversaries, specifically when the adversaries can collate to attack the system. This game formation expresses two advantages – the first is that it helps to understand when the weights must be updated, and the second is that it allows understanding actions points for the adversaries ensuring the utility of FL for securing the operations of misbehaviour detection. Following it, $L(\mathcal{N}) \geq |\mathcal{M}|$ is defined as the difference between the honest players and the collated adversaries operating against the honest players, such that $H$ is the operational function for honest devices and $A$ is the action function for the adversary, defined as:
Here, we describe the number of times the weights are accurately identified by the adversary. Specifically, the adversary updates the weights by integrating the lightweight deep learning framework (LDLF) and MDS. The outputs are stored securely, and the rewards are distributed equitably.

$$L(|N| \geq |M|) = \sum_{i=1}^{N} H_i - \sum_{j=1}^{M} A_j,$$  \hspace{1cm} (2)

where $|N|$ denotes the number of honest devices and $|M|$ denotes the adversarial devices, such that $H$ is the operational function for honest devices and $A$ is the action function for the adversary, defined as:

$$H_i = P_{H_i}[w] \cdot \frac{\Delta W_{H_i} \varphi|S_i|}{(k_i(k_i+1))},$$  \hspace{1cm} (3)

and

$$A_j = P_{A_j}[w] \cdot \Delta W_{A_j},$$  \hspace{1cm} (4)

where $P_{H_i}[w]$ and $P_{A_j}[w]$ define the probability of weights being updated accurately and traced by the adversary accurately, respectively using Poisson distribution, such that $P_{H_i}[w] = \frac{\lambda^{|N|} e^{-\lambda}}{|N|!}$, where $\lambda$ is the number of times the weights are updated for transitions between $\tau + 1$ and $\tau$. $\lambda$ is the rate of changes occurring in states to the total number of states for ith device. $\varphi|S_i|$ is the total transitions for the actual $|S_i|$ states of ith device. Similarly, for the adversary, $P_{A_j}[w] = \frac{\lambda^{|M|} e^{-\lambda}}{|M|!}$, where $\lambda$ is the number of times the weights are accurately identified by the adversary. Here, $\Delta W_{H_i} = ||W_{i,\tau+1} - W_{i,\tau}||$ and $||W_{j,\tau+1} - W_{j,\tau}||$ denote the weight difference for the honest devices and the adversaries between two intervals denoted by $\tau + 1$ and $\tau$. Based on the Stackelberg game, the requirement is converted into an optimization problem, where the objective is:

$$P1: \quad L(x), \quad \max_{\forall i \neq j}$$

$$C1: \Delta W_{H_i} \neq 0 \quad \text{and} \quad \Delta W_{A_j} \geq 0, \quad i \neq j, \forall i, \forall j$$

$$C2: \Lambda \geq 0, \quad U_{(\tau+1)\rightarrow \tau} > 0, \quad \forall \theta \geq 0$$

$$C3: \varphi|S_i| << \frac{k_i(k_i+1)}{2}$$

There exists a trade-off between the FL operations, anonymity and the connectivity of the graph, which can improve the traceability of misbehaviour – if the number of transitions for the given number of states, $\varphi|S_i|$, is too close to $k_i(k_i+1)$ number of transitions (total), then the adversaries will require to match weights for lesser number of combinations, thus, making it easier to predict. Now, the system can operate using FL for the devices in its control; hence the problem deduces to:

$$P2: \quad \max \left(\frac{\Delta W_{H_i} \varphi|S_i|}{k_i(k_i+1)}\right)$$

Following the constraints in C1 to C3. The weights follow a standard normal distribution if the model is operated for a longer duration as the converging rate will become constant, resulting in a zero mean and a unit standard deviation ($\mu = 0$ and $\sigma = 1$), which can be used to represent as $P(\Delta W_H) = \frac{-\Delta W_H^2}{2}$. Now, it can be used to understand the bounds for the behaviour of the problem, P2. If such is the case, then P2 will be operating with two controlling variables $P_{H_i}$ and $P(\Delta W_{H_i})$, both of which will depend on the number of times the weights are adjusted in the setup.
If $U_{t+1-\tau}$ becomes too large, $P(\tau + 1 - \tau)$ will decrease considerably, however, it will also affect the convergence rate of the model and can cause latency, which is not an ideal situation for a setup operating with IoMT. Hence, the model can further divide itself to reduce the burden on performance while keeping intact the data privacy.

If $U_{t+1-\tau} = \Lambda$, then the function will be bounded by values $\geq 1$, which is the non-complex minimum calculable when both the rates are equal. Alternatively, the function will attain its minimum when $U_{t+1-\tau} = \Lambda$. This helps to understand the impact of using FL and how it can be used for identifying misbehavior by finding values of weights leading to a minimum of the function and giving a minimum chance to an adversary for avoiding this detection. This has been illustrated using Fig. 3. It can be observed that the convergence rate will be affected if too frequent updates are performed to the weights, which will give more chances to an adversary to avoid the misbehaviour detection as it can predict the states and avoid getting caught by the detection mechanism.

Following the understanding of the utility of FL, it is supported by two layers of Bidirectional Long-Short Term Memory (BiLSTM) to help with the misbehaviour detection and ensure the privacy of the user relying on the properties of FL. The details are as follows:

1) **Bidirectional Long-Short Term Memory (BiLSTM):** LSTM is a variant of recurrent neural network (RNN). Because of its success, LSTM was specifically claimed as a solution for the technical issues of the classical RNN structure. In addition, LSTM practically remembers information for a long term, apart from learning the patterns of a given data. In consequence, its design feature makes LSTM appropriate for building models for those data that are collected in time order. Moreover, unlike classical RNN, LSTM can operate with flexible time steps of time-series data and resolves the vanishing gradient problem [43].

Like classical RNN top-level design, LSTM still follows the chain-like structure of repeating modules, called LSTM cells. The information flowing in the chain are controlled by four network passes, namely forget gate ($f_t$), input gate ($i_t$), cell state candidate ($\hat{C}_t$), and output gate ($o_t$). Each passes results from an element-wise sigmoid function ($\sigma$) and are combined through point-wise multiplication operation, as shown in Fig. 4. The inference ($h_t$) of the model is obtained by the following composite function:

$$
\begin{align*}
  f_t &= \sigma(W_f \times [h_{t-1}, x_t] + b_f) \\
  i_t &= \sigma(W_i \times [h_{t-1}, x_t] + b_i) \\
  \hat{C}_t &= \tanh(W_C \times [h_{t-1}, x_t] + b_C) \\
  C_t &= f_t \times C_{t-1} + i_t \times \hat{C}_t \\
  o_t &= \sigma(W_o \times [h_{t-1}, x_t] + b_o) \\
  h_t &= o_t \times \tanh(C_t)
\end{align*}
$$

where $W_s$ and $b_s$ are weight matrix and bias of the gate layers.

The processing flow of a typical LSTM-based prediction model is limited to forward direction, making use of only the previous events [44]. To overcome this limitation, LSTM is extended to a bidirectional flow structure with the goal of enhancing the model performance in sequence-related problems. The output of the model is obtained wherein the future context is included in the analysis. Figure 5 portrays the logistic flow of BiLSTM. Accordingly, BiLSTM is composed of two hidden layers, respectively processing input sequence data in forward and reverse timestep directions.

2) **Misbehavior Detection System (MDS) Architecture Overview:** The proposed MDS contains two-tier of the FL-based model, as presented in Fig. 6. The first tier implements a BiLSTM-based estimator, which periodically forecasts the blood glucose level (BGL) of a patient based on the previous $n$ timesteps. The estimated value is utilized in computing insulin dosage and insulin amount in the pump vial. Furthermore, the estimate and its derivatives are operationally combined with other attributes, generating input features for behavior prediction. The second level also implements the BiLSTM structure in classifying the system as well-behaved or malicious. In the latter case, the logical agent notifies the authorized person to quickly address the anomaly. Note that this paper leverages the input features introduced in [45]. For detail, expressions [7] and [10] defines the input features of the CGM estimator and classifier, respectively. We refer the reader to [45] for clarity.
EstimatedCGM_t ← [G_{t-n}, ..., G_{t-2}, G_{t-1}]

where G ∈ \{glucose reading, carbs intake, insulin intake\}

Behavior ← [B_{t-n}, ..., B_{t-2}, B_{t-1}]

where B ∈ \{glucose estimation error, message 1 arrival error, insulin dosage estimation error, message 2 arrival error, in-vial estimation error, message 3 arrival error\}

To this end, the application of this deep learning solution generally demands model-carrier with high computation capability and memory due to increasing complexity. Hence, the conversion of trained models to lightweight version is desirable to the case study, i.e., memory requirement of the model is reduced and the latency of arriving to a decision is significantly low, but still achieving considerably high accuracy. Accordingly, this work applies the simplest model compression method called post-training quantization technique, wherein the 32-bit floating-point precision of the based model parameters are transformed into an 8-bit precision. In turn, the inference latency and memory consumption is ideally reduced by 16 and 4 times, respectively [46, 47].

Fig. 6. Two tier design of collaboratively-trained BiLSTM-based MDS.

C. Decentralized Privacy Preserving

Our incentive scheme is designed by diversifying the CryptoNote protocol [48], where this protocol is a pioneer in delivering user’s privacy in a decentralized incentive mechanism. This technique is implemented in one of the Blockchain cryptocurrencies known as Monero (XMR). It is an open-source, private, decentralized cryptocurrency that keeps transactions confidential and secure.

1) Secure Distributed Ledger Management: We construct a secure distributed ledger management protocol empowered by Ethereum smart contract by referring to the CryptoNote layer protocol to solve specific issues identified in Bitcoin transactions. The designed protocols leverage a group of ring signatures. It is a multi-signer digital signature scheme where a group of ring signature schemes possesses N signers forming a ring. Whenever the signers in the group receive a transaction to be endorsed, they can produce a ring signature with a corresponding private key. Once the ring is successfully created, every group member can use the signature on behalf of the group to keep their identity secret. A ring signature can be defined as a type of digital signature algorithm that any member of the group can calculate. Whilst, the ring confidential transactions perform a list of prior transactions to obscure the original value of current transactions. The user US_k enables to choose the number of signatures RG_x to be used in a transaction T_x. The selected signature is part of the ring that has been generated in advance, where RG_x ∈ RG_to ≥ 1.

The primary key values for each lightweight IoT device within MEC are obtained from the trapdoor permutation function as a set of one-way function f_{trap}_z: M_{kr} → N_{kr}(kr ∈ KR), where for all KR, M_{kr}. N_{kr} is a subset of binary strings value \{0, 1\}^z, fulfilling a certain number of requirements, such as there exists a sampling of probabilistic polynomial time Create(1^n) = (kr, x_{kr}) with \{kr ∈ KR ∩ \{0, 1\}^n; where x_{kr} ∈ \{0, 1\}^n meets |x_{kr}| < pol(n) with pol defined as some polynomial values. Thus, every x_{kr} is known as trapdoor corresponding to kr value. Suppose any kr ∈ KR with pol algorithm for every x ∈ M_{kr}, let z = α(kr, f_{trap}(X), x_{kr}). Accordingly, the function possess f_{trap}_z(x) = f_{trap}(x). Each entity holds a pair of parent keys (Pub_{kr}, Sec_{kr}) generated beforehand. The encryption Pub = trap_{n}(q_{n}) generates the public key; trap_{n} is a trapdoor permutation function, with trap_{n}(q_{n}) specifies f_{q}(q_{n}) = q^x mod n over \{0, 1\}^n. Eventually, the signature keys of entities can be defined as follows:

\[\begin{align*}
\text{RNG}_{\text{sgn}} & \rightarrow GVR_{n} \oplus PET_{n} \oplus INC_{n}, \ldots, \oplus NEW_{n}; \\
\{\text{trap}_{GVR}(q_{GVR}) \oplus \text{trap}_{PET}(q_{PET}) \oplus \ldots \oplus \text{trap}_{NEW}(q_{NEW})\}; \\
\text{trap}_{PET}(q_{PET}) \oplus \text{trap}_{NEW}(q_{NEW})\}.
\end{align*}\] (11)

The observers have no knowledge about the sender’s information due to the transaction being signed on behalf of the group. The sender is free to choose the number of signatures as stated in the formula (11). Members can use the signature within the ring for any lightweight IoT transactions without requiring approval from each group member. Each addition of a new entity can be executed regularly as long as the public key is known (Update_RNG_{sgn}). Likewise, excluding members from the ring can be managed unilaterally by the manager (Exclude_RNG_{sgn}).

The constructed protocols are based on the elliptic curve cryptography established with regard to multiplicative cyclic groups. The secret key Sec_{EN_T n} defines Q_x ∈ [1, l - 1]; with l represents the prime order of a base point in the elliptic curve cryptography. Meanwhile, the public key Pub_{EN_T n} is understood as a point of Pub_{a} = Sec_{a} ⋅ G (with G is a generator for Pub_{a}). There exists a pair of tracking keys track_keys(Q_x, Pub_{b}) obtained from Secret and Publickeys(Pub_{b} = Sec_{b} ⋅ G with condition Sec_{a} ≠ Sec_{b}) [49]. Finally, the description of protocols, which is also a part of ring confidential transactions, can be interpreted as follows:
The type of data stored can be adjusted through the description of continuous glucose monitors (GMC), and any other behavioural data, such as the main diagnosis function (\(\Delta\text{PET}\)). CID is short, a unique content identifier of IPFS. This key is being used to store signed data. The sender associates his/her public keys with a combining function \(\psi\). The first public key \(PET_{\alpha}\) is created based on PET's private key \(PET_{\text{Sec}}\) from a certain base point/generator \(\mathbb{G}_\alpha\). The key is being used together with the recipient’s random data \(rand\) where \(R = rand \cdot G\). Concurrently, the other PET’s public key \(PET_{\text{Pub}3}\) is generated from another PET’s secret key \(PET_{\text{Sec}3}\) corresponding to its generator: \(PET_{\text{Pub}3} \rightarrow PET_{\text{Sec}3} \cdot G\), with condition \(\mathbb{G}_\alpha \neq G\). Hence, PET_{\text{Sec}3} \neq PET_{\text{Sec}3} as depicted in formula (14), inspired by [50], [51]. The second public key \(PET_{\text{Pub}3}\) is assigned as a tracking key within Blockchain network. The sender \(TX_{\text{PET}1}\) can recognize the reward which belongs to him by checking the tracking key attached into \(TX_{\text{PET}1}\) as defined in (13).

Once \(TX_{\text{PET}1}\) is completed, the data owner receives a reward in the form of Ether cryptocurrency. The amount of Ether obtained can be adjusted by the provider. In the first place, the provider confirms the transactions claimed by the sender. If all requirements are satisfied, then the provider unpacks the public keys attached in \(TX_{\text{PET}1}\), and straightforwardly executes a random base point \(\text{rand} \in [1, \ldots, 1]\) while also computes a provider’s one-time used \(\text{GVR}_{\text{OTU}}\) for the sender \(PET_{1}\) as shown in (15).

\[
GVR_{\text{OTU}} = \text{Hash} (\text{rand} \cdot PET_{\text{Sec}1} \cdot G) \cdot G + PET_{\text{Pub}3} \quad \text{(15)}
\]

\[
\text{OPK}_{\text{e1}} = H_1 (\text{Priv}_{\alpha} \cdot \text{R}) + \text{Priv}_{\beta} \quad \text{(16)}
\]

The reward sent by provider GVR is only available for patient 1 \(PET_{1}\) because only \(PET_{1}\) knows the secret information of the transaction. Before a certain amount of Ether is transferred through the Ethereum network, the GVR must confirm

**Table III**

**Summary of notations used (Blockchain transactions).**

<table>
<thead>
<tr>
<th>Notations</th>
<th>Definition</th>
</tr>
</thead>
</table>
| \(RNG_{\text{sgn}}\) | Number of ring signatures; Total \(RG_{\text{tot}}\) available in \(RNG_{\text{tot}}\) tab |}
| \(GVR_{\text{R}}\) | Healthcare providers (local hospitals); Patients |}
| \(\text{New}_{\alpha}, \text{Inc}_{\alpha}\) | New entity (\(n\) numbers); Incentive manager |}
| \(\text{Pub}_{\alpha}, \text{Sec}_{\alpha}\) | Public key \(n\) and secret key \(n\) |}
| \(\text{hash}, \text{trap}(f)\) | 256-bit hash; trapdoor permutation function |}
| \(f_{\text{inc}}\) | A set of one-way function |}}

**TABLE III**

**Summary of notations used (Blockchain transactions).**
that all states in $TX_{PET_1}$ are correct (labelled as "True"). Patient 1 examines every passing transaction using PET's private key $PET_{Secα_1}$. $PET_{Secβ_1}$, $PET_{1}$ is authorized to recover the corresponding $GV_{ROT_{k}}$ key since $PET_{1}$ is the owner of $PET_{Secα_1}, PET_{Secβ_1}$. Eventually, the $PET_{1}$'s one time private key is signified in [10]. This key is being used to spend the $Ether$ transferred by $GV_{ROT}$. In the original state of Monero (XMR) technology, the combination of these keys is being utilized as a part of a ring signature to obscure the sender's information. A fresh key-image is also attached to prevent the double-spending attack.

IV. PERFORMANCE RESULTS AND COMPARISON

A. Misbehavior Detection Experiment and Performance

In the experiment, we train and measure the performance of the proposed MDS. The accuracy of the estimation model is quantified by the root mean square error, while the classification model performance is measured based on the prediction accuracy, precision, recall, and F1-score. This section first briefly describes these performance indicators, machine environment, and the summary of results.

The root mean square error (RMSE) calculates the total deviation of the predicted glucose values ($G_{est_i}$) with the actual measurements ($G_{meas_i}$), as expressed in equation (17). A smaller total deviation from a number of samples indicates better accuracy. Meanwhile, classification performance indicators are measured by collecting the $T_n$, $T_p$, $F_p$, and $F_n$ of the given test samples. These key parameters denote true negative, true positive, false negative, and false-positive counts, respectively. The metric accuracy measures how well the models classify both the malignant and benign events, whereas the other three metrics measure how well the models are in classifying malignant events. Precision is the level of reliability with which the trained model correctly identifies the malignant events. As shown in expression [19], this indicator is calculated as the ratio between of correctly identified malignant events ($T_p$) and all events identified as malignant, where $F_p$ is the number of benign events misclassified as malignant. Meanwhile, recall indicates how good the models are at classifying malignant events. It is the ratio between the number of correctly identified malignant events and the total of actual malignant events. Regarding misbehavior detection performance, the models with relatively low recall rates are regarded as ineffective due to the high incorrect classification of malignant events. Thereby, an effective method requires a recall rate as high as possible since undetected attacks can place patients in unfavorable health conditions. Finally, the F1-score is a performance metric that incorporates precision and recall into a single value. A high F1-score indicates $F_p$ and $F_n$ are both low. In this article, the platform serving as the aggregator of the estimation and classification model are Raspberry-Pis. The system is implemented using Python language. The detailed machine specification and software tools utilized are listed in Table [IV]

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (G_{est_i} - G_{meas_i})^2}$$ (17)

$$\text{Accuracy} = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}$$ (18)

$$\text{Precision} = \frac{T_p}{T_p + F_p}$$ (19)

$$\text{Recall} = \frac{T_p}{T_p + F_n}$$ (20)

1) Dataset Preparation: The dataset utilized for the training and evaluation was generated from the glucose-insulin simulator of [20]. In this experiment, 2400 of 6-tuple time-series samples was collected from a virtual diabetes patient. The collected data contains six features, including glucose level, the message transmission time of glucose level to the controller, insulin dosage, message transmission time of injection command to the insulin pump, insulin-on-vial (IoV), and message transmission time of IoV back to the controller. In this case, five virtual patients, each having distinct glucose-insulin behavior, were simulated. Forty percent of the samples were utilized in training and evaluation of the estimation model, and the remainder was allocated for the evaluation of the classification model.

It should be noted that the data being collected from the simulator are technically based on the regular operation, wherein the samples are assigned as benign. Furthermore, we augment the data to include malignant samples in the training and evaluation of the classification model. This is done by arbitrarily changing a feature value using equation (22). In addition, the dataset of each patient is augmented at different degrees to differentiate training and validation data size. Finally, the input features for the training and validation are arranged in blocks containing $n$ timesteps sequence. In the end, the generated training and validation datasets have sizes summarised in Table [V]

$$malignant \_ \text{feature} = \text{benign \_ feature} \pm \text{benign \_ feature} \times \text{rand}([10\% \ 50\%])$$ (22)

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Total Size</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device 1 Training</td>
<td>3532 blocks</td>
<td>2668 benign and 864 malicious</td>
</tr>
<tr>
<td>Device 2 Training</td>
<td>3140 blocks</td>
<td>2387 benign and 733 malicious</td>
</tr>
<tr>
<td>Device 3 Training</td>
<td>2748 blocks</td>
<td>2080 benign and 668 malicious</td>
</tr>
<tr>
<td>Device 4 Training</td>
<td>2356 blocks</td>
<td>1771 benign and 585 malicious</td>
</tr>
<tr>
<td>Device 5 Training</td>
<td>1963 blocks</td>
<td>1498 benign and 465 malicious</td>
</tr>
<tr>
<td>Validation Set</td>
<td>19631 blocks</td>
<td>14846 benign and 4785 malicious</td>
</tr>
</tbody>
</table>

2) Performance Results: Finally, the performance results of the proposed method are collated in different settings. Firstly, we investigate the performance of the forecasting model at varying numbers
of participating devices and in the timesteps sequence of 5, 7, and 9. Accordingly, we collate the RMSE from the collaborating devices and compute the average value at every setting. For comparison purposes, we also gathered RMSE from each while building model individually. Fig. 7 displays the average RMSE value at every communication round or epoch. It can be observed in the figure that when devices individually build the model, the average RMSE decreases slowly. On the other hand, as the number of model-building devices increases, the average RMSE between collaborating devices decreases at shorter communication rounds. It should be noted that the figure only presents the results when the timestep sequence is set to 5, and the same trend was observed in the other settings. To this end, the input training sample with five timesteps sequence produces the lowest RMSE from 5 collaborating devices as shown in Table VI.

Subsequently, we investigate the accuracy of the classification model at varying collaborating devices while adapting the timesteps sequence with the lowest average RMSE from the first experiment. As shown in Fig. 8 when more devices participate in the model building, the learning rate increases in shorter communication rounds. Finally, Table VII presents the comparison of the average classification performance of different deep learning approaches. On the one hand, the size of the BiLSTM model has decreased by 43% after applying post-quantization method, and the inference latency has reduced by almost 94%. However, both indicators are relatively higher than the other neural network structures due to the significantly more complex structure. On the other hand, the proposed BiLSTM-based MDS dominates the other approaches, achieving a high recall of 99.93%. This indicates that the proposed model can capture almost all potential malicious events in the target use case.

B. Blockchain Environment Setup

As mentioned earlier, we leverage the Ethereum Blockchain platform as a cornerstone of a decentralized application. In order to connect with the Ethereum network, we utilize Truffle - Suite framework for testing and deploying the instruction within the smart contract. The functions of smart contracts can be adjusted over time through the migrations feature provided by the framework. The smart contract is executed by a computation engine that acts as a decentralized computer network known as the Ethereum virtual machine (EVM). Every Blockchain node operates on the EVM to maintain consensus across the network [52]. The EVM can be understood as a mathematical function that takes any input to produce a deterministic output with a state transition function. Ethereum-enabled applications using Truffle Suite execute and inspect the state transition function with all essential dependencies installed. This framework is an *automining* mode and running on the remote call server http://127.0.0.1:7545 (network ID: 5777) with the currency symbol is ‘Ether’. The gas price and gas limit are set to 20,000,000,000 Wei, and 6,721,975 respectively. Prefix values of hexadecimal number and the unique identification of the entities are obtained from Truffle - Suite framework.

C. Compatible Privacy Preservation

The Blockchain-based privacy preservation framework is formed by deploying transactions (e.g., $TX_{PETn}$ patient’s data) into the Ethereum network. The patients state their relevant information in the smart contract, such as the primary diagnosis function, description, actual data, and details. The patient also attaches a pair of public keys to be used mutually with the recipient’s random data (in line with Diffie-Hellman key exchange), while the other key acts as a tracking key. The fundamental objectives of smart contract adoption in this research are the immutable transactions record and commensurate incentive mechanism. Tamper-proof property is achieved by design, while the cost of incentive given to the data owner is proportional to the data amount. We deploy our contracts to the Ethereum network by running the migration file. Over time, the contract functions can be changed by re-running the migrations file (responsible for

![Fig. 7. Average RMSE at every epoch/communication round in varying collaborating devices using 5 timesteps sequence](image)

**TABLE VI**

<table>
<thead>
<tr>
<th>Timestep Sequence</th>
<th>Average RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1.3482</td>
</tr>
<tr>
<td>7</td>
<td>1.9959</td>
</tr>
<tr>
<td>9</td>
<td>2.6099</td>
</tr>
</tbody>
</table>

![Fig. 8. Average classification accuracy at every communication round in varying collaborating devices using 5 timesteps sequence.](image)

**TABLE VII**

<table>
<thead>
<tr>
<th>Performance Comparison of Different Neural Network Structures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Algorithm</strong></td>
</tr>
<tr>
<td>Metrics</td>
</tr>
<tr>
<td>Accuracy</td>
</tr>
<tr>
<td>Precision</td>
</tr>
<tr>
<td>Recall</td>
</tr>
<tr>
<td>F1-Score</td>
</tr>
<tr>
<td>Latency (ms)</td>
</tr>
<tr>
<td>Mode size (bytes)</td>
</tr>
</tbody>
</table>

| **Legend:** | **Performance of the Base Model** |
staging the deployment tasks). The network id and block gas limit is set to 5777 and 6721975 (0x6691b7), respectively. When the contract is successfully deployed to the Ethereum network, then the contract will have an address. Eventually, the migration process details are displayed together, such as the number of blocks, current balance (Ether), block timestamp, account, gas used in deploying the contract, gas price, the value sent, and total cost. Roughly, the contract migration process consumes 225237 gas units, with a total cost of around 0.00450474 ETH.

In deploying the contracts, the genesis block is created automatically with zero units of gas used, and it is recorded in the Block 0 on the chain. The genesis block is perpetually hardcorded into the software of the applications that use the Ethereum smart contract. The first transaction \( T_X PET_1 \) of patient 1 \( PET_1 \) is recorded in the Block 5 with transaction index 0, gas used is 111367 units (6721975 gas limits), and the ID of transaction is \( \log_{10} 125444 \)eb. Intuitively, a timestamp for every block can be understood as a block generation after getting confirmation from the miners in the network. When the node receives a new block from another node in the network, the recipient confirms that the timestamp value is correct and does not outpace the Universal Time Coordinated (UTC) by more than 100 milliseconds. Otherwise, the block is rejected. The information of this transaction can be seen in Fig. 9.

![Fig. 9. The detail of information that the data owner has conducted. This transaction is mined in Block 5 with 111367 gas usage.](image)

### Table VIII

**Summary of the cumulative gas consumption and Ether spent by \( PET_1 \) and \( GV R \).**

<table>
<thead>
<tr>
<th>No.</th>
<th>Benchmark</th>
<th>( PET_1 )</th>
<th>( PET_2 )</th>
<th>( GV R_1 )</th>
<th>( GV R_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( Ether ) amount (init.)</td>
<td>100 ETH</td>
<td>100 ETH</td>
<td>100 ETH</td>
<td>100 ETH</td>
</tr>
<tr>
<td>2</td>
<td>Gas usage (min)</td>
<td>91468</td>
<td>100615</td>
<td>61141</td>
<td>61752</td>
</tr>
<tr>
<td>3</td>
<td>Gas usage (max)</td>
<td>125721</td>
<td>138293</td>
<td>65754</td>
<td>66411</td>
</tr>
<tr>
<td>4</td>
<td>Gas usage (avg)</td>
<td>100717</td>
<td>110789</td>
<td>62846</td>
<td>63474</td>
</tr>
<tr>
<td>5</td>
<td>( Ether ) spent (min)</td>
<td>5.56 x 10^{-3}</td>
<td>1.82 x 10^{-2}</td>
<td>6.42 x 10^{-3}</td>
<td>1.94 x 10^{-2}</td>
</tr>
<tr>
<td>6</td>
<td>( Ether ) spent (max)</td>
<td>9.87 x 10^{-3}</td>
<td>7.94 x 10^{-2}</td>
<td>9.57 x 10^{-3}</td>
<td>3.44 x 10^{-2}</td>
</tr>
</tbody>
</table>

The visual output of the transaction is shown in Fig. 10. The knowledge about **Ether** spent in Table VIII describe an accumulated calculation of gas usage, gas limits, and the gas price calculated automatically by the Truffle - Suite framework. The IPFS network benchmark used in our framework can be seen in Table IX. Every entity possesses a UNIX ID (ips-unixfs) and public key connected through a specific gateway, i.e. http://127.0.0.1:8080. The agent and UI versions are go-ipfs v0.11.0 and v2.13.0. The data stored can vary in terms of type, size, etc. The number of peers available dynamically change over time. As highlighted in Table IX, the CID and multihash functions are derived once the data are successfully stored. Eventually, our proposed scheme can be a plausible solution to address the privacy and transparency issues in the Blockchain. Nevertheless, the hard fork in the Blockchain is needed since the core of protocols is updated. Blockchain hard-fork is not a straightforward challenge to deal with. In other words, it can be a significant obstacle in embracing the proposed model in the real world.

### V. Conclusion

We have presented a Blockchain-based privacy preservation framework for secure misbehavior detection employed in lightweight IoT devices. Our proposed model becomes essential for securing sensitive healthcare data which runs through lightweight IoT devices systems. The privacy concerns are tackled by diversifying several cryptography protocols that cut off the linkage between private data and the corresponding owner. We also applied FL strategy that enhances patient privacy by keeping training data within the owner’s digital realm and building the global model out of sub-models that are trained locally by participating devices. Furthermore, the entities’ information is expressly disguised by an Ethereum smart contract. We have completed the main requirements stated in this research, such as privacy preservation and a commensurate incentive mechanism that legitimate entities can only recognize. However, our designed scheme is embedded into smart contract transactions, affecting gas usage per input of arbitrary values. Realizing our protocols on the core of the Ethereum network, a costly hard fork is required whereby...
Fig. 10. (a) Comparison of Ethereum gas usage between patients and healthcare providers in conducting transactions. Two patients with their respective data conducted $T_X,PET_1$ and $T_X,PET_2$, respectively; (b) Comparison for the last ten transactions of patients.

**TABLE IX**

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender_1 ID</td>
<td>12D3KoWN6q85mdmLYE2GWe9Z ...xxx</td>
</tr>
<tr>
<td>Public key</td>
<td>CAESIMGL4w622Zsk3l6CVQlCMzy ...xxx</td>
</tr>
<tr>
<td>AgentVersion &amp; UI ver.</td>
<td>go-ips v0.11.0 &amp; v2.13.0</td>
</tr>
<tr>
<td>API</td>
<td>/hp4/127.0.0.1/kcp/5001</td>
</tr>
<tr>
<td>Data size &amp; Peers avlb.</td>
<td>966 KB; 125 Peers (dynamically changing)</td>
</tr>
<tr>
<td>Content identifier (CID)</td>
<td>Qmd:FR:3G9v9QoWuyAvzusu ...xxx</td>
</tr>
<tr>
<td>Links</td>
<td>4 links (Path: Links/0, Links/1 - Links/3)</td>
</tr>
<tr>
<td>Multilash function</td>
<td>0x120E3617...xxx (0x12 = sha2-256; 0x20 = 256 bits)</td>
</tr>
<tr>
<td>Avg. Network traffic</td>
<td>76 KiB/s incoming: 32 KiB/s outgoing (3 mins pre - post transactions)</td>
</tr>
</tbody>
</table>

the massive number of clients must update the latest version of the Ethereum network. In addition, the performance of the misbehavior detection algorithms under the BiLSTM technique shows compact findings, with recall rates exceeding 99 percent, implying that the algorithms are successful in capturing almost all malicious events in the target healthcare system. Conclusively, the overall events positively recommend that our schemes satisfy the design objectives. The dataset utilized to evaluate the proposed approach were taken from the works in [20] and [45], in which the authors extended a simulation tool called UVA/Padova. It is known that this tool is already approved by Food and Drug Administration for educational and pre-clinical trials. Nevertheless, this current work acknowledges that validation of the proposed approach in real environment with true patient data is vital. Hence, this concern will be included in the future works.

**REFERENCES**


