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/TII.2020.2974555, IEEE Transactions on Industrial Informatics

A Trustworthy Privacy Preserving Framework for Machine Learning in Industrial IoT Systems

Pathum Chamikara Mahawaga Arachchige, Peter Bertok, Ibrahim Khalil, Dongxi Liu, Seyit Camtepe, and Mohammed Atiquzzaman

***Abstract*—Industrial internet of things (IIoT) is revolutionizing many leading industries such as energy, agriculture, mining, transportation, and healthcare. IIoT is a major driving force for Industry 4.0, which heavily utilizes machine learning (ML) to capitalize on the massive interconnection and large volumes of IIoT data. However, ML models that are trained on sensitive data tend to leak privacy to adversarial attacks, limiting its full potential in Industry 4.0. This paper introduces a framework named PriModChain that enforces privacy and trustworthiness on IIoT data by amalgamating differential privacy, federated ML, Ethereum blockchain, and smart contracts. The feasibility of PriModChain in terms of privacy, security, reliability, safety, and resilience was evaluated using simulations developed in Python with socket programming on a general-purpose computer. We used Ganache_v2.0.1 local test network for the local experiments and Kovan test network for the public blockchain testing. We verified the proposed security protocol using Scyther_v1.1.3 protocol verifier.**

***Index Terms*—IIoT, Industry 4.0, IIoT trustworthiness, blockchains, Ethereum, smart contract, federated learning, differential privacy, machine learning, IPFS**

I. Introduction

THE Industrial Internet of Things (IIoT) uses sensors and actuators with computing and communication capabilities to change the way data is collected, exchanged, analyzed, and transformed into decisions. Their increasing pervasive ability leads to innovative Industry 4.0 (also referred to as Industrial Internet) applications for improved productivity and efficiency in major industries such as energy, agriculture, mining, transportation, and healthcare. Machine learning (ML) plays a significant role in Industry 4.0, enabling predictive analytics, and uncovering essential insights to transform industries. With the advancement of computing and communication technologies, ML enables the analysis of massive quantities of data such as those produced by an IIoT-based system, and can use the extracted knowledge (e.g. trained models) to aid real-time decision making in complex situations. Compared to other areas of ML, deep learning (DL) shows remarkable accuracy towards image classification, natural language processing, and speech recognition. Fault detection and isolation in industrial processes [1], real-time quality monitoring in additive manufacturing [2], and automatic fruit classification [3] are three of the recent examples of DL in IIoT-based Industry 4.0 systems. Consequently, IoT has become one of the enabling forces for Industry 4.0, pushing companies towards a paradigm shift to attain advantages in the competitive dynamic market [4].

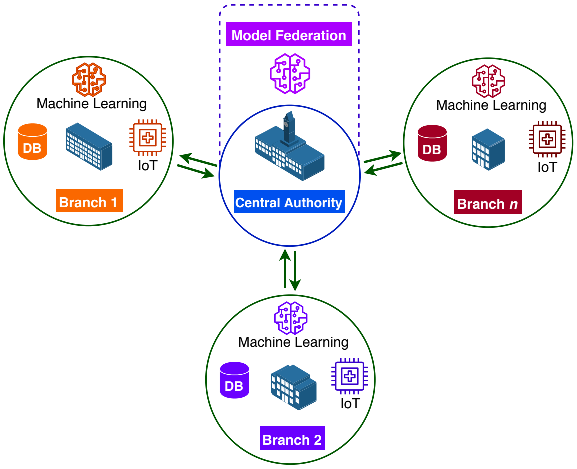


Fig. 1: Model knowledge sharing in IIoT-based Industry 4.0

A large scale IIoT-based Industry setup is composed of a collection of vastly geographically distributed entities, as depicted in Fig. 1. Consequently, the advanced Industry 4.0 features such as predictive maintenance and ML-based quality control and runtime reasoning need to be facilitated by distributed data acquisition [4]. As shown in Fig. 1, in an IIoT-based system such as smart healthcare, and open banking, data and ML models trained within the local boundaries need to be communicated with the intended users/branches to generate organization-wide knowledge. Vendors often want to restrict their internal insights on product development and improvements within their organizational boundaries to increase their business value against their contenders. Moreover, industries such as smart healthcare and open banking are massively convoluted with human-specific sensitive private data. This complexity makes the processes of distributed data acquisition quite challenging in an IIoT-based Industry 4.0 setting. ML models that are trained on sensitive data can reveal private or confidential information to advanced adversaries [5], [6]. An attack such as “man in the middle” conducted by an adversary

Manuscript received August 30, 2019; revised January 03, 2020; accepted February 13, 2020. (Corresponding author: M.A.P. Chamikara)

M.A.P. Chamikara is with the department of Computer Science and Software Engineering at the School of Science, RMIT University, Australia. He is also with CSIRO Data61, Melbourne, Australia.
E-mail: pathumchamikara.mahawagaarachchige@rmit.edu.au

P. Bertok and I. Khalil are with the department of Computer Science and Software Engineering at the School of Science, RMIT University, Melbourne, Australia.

D. Liu and S. Camtepe are with CSIRO Data61, Sydney, Australia.

M. Atiquzzaman is with the School of Computer Science at the University of Oklahoma.

can cause changes to the original ML knowledge transferred by the source. Malicious algorithms can be implemented, offering them as part of the underlying training processes to memorize the sensitive information. Adversaries can later extract and approximate the memorized information, thereby obtaining sensitive information to breach privacy [7]. Privacy inference attacks, such as membership inference and model inversion, show more vulnerability of machine learning models trained on sensitive data [6], [8]. Hence, privacy and trustworthiness are essential components of ML in IIoT systems.

Fig. 2 illustrates the five pillars/parameters of trustworthiness in IIoT systems [9]. Enforcing these parameters, guarantees a safe and trustworthy IIoT platform that avoids the threats (e.g. spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege) identified by the STRIDE threat model [9].

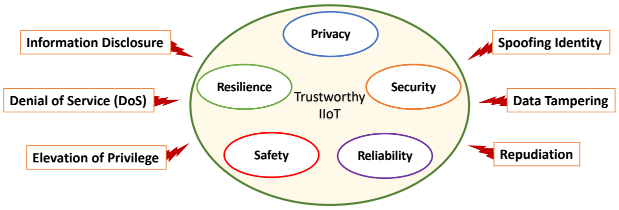


Fig. 2: Five pillars of a trustworthy IIoT system vs. STRIDE model of threats

Our contribution is a framework named as PriModChain (**Pri**vacy-preserving trustworthy machine learning **mod**el training and sharing framework based on block**chain**) that addresses the privacy and trust issues of machine learning in IIoT systems. PriModChain blends differential privacy, federated learning (FedML), smart contracts and Ethereum blockchain (EthBC). PriModChain uses the interplanetary file system (IPFS) for off-chain data management.

The proposed framework (PriModChain) uses FedML to generate a global representation of the distributed machine learning knowledge in a distributed IIoT environment. FedML provides the capability of training an ML model against both static data and data streams. As the original models do not leave the model owners, FedML provides a certain level of privacy in its default setting. In order to strengthen the privacy of input data, we apply differential privacy on the locally generated models. The federation of the differentially private models is coordinated using a smart contract on the EthBC to introduce security, safety, and resilience. The smart contract provides transparency to PriModChain in generating the global ML model once an agreement between the central authority (CENTAUTH) and the distributed entities (DISTEN) is established. EthBC makes sure this agreement is supported by the highest level of data encryption to enforce security. The transparent and autonomous nature of the agreements in the EthBC enforces unbiased and error fee data manipulations, improving the trust in terms of safety and reliability. Traceability and immutability are two other vital properties of EthBC, where the essential details of the transactions are permanently stored for future reference. This property allows verifiable computation in PriModChain, enforcing safety and resilience over IIoT data. Due to the high capacity of large ML models, we use IPFS as the off-chain storage module of PriModChain. IPFS introduces immutability, low latency, and fast decentralized archiving with secure P2P content delivery. We apply encryption over the differentially private ML model parameters to enhance the security of data stored in IPFS. We tested the trustworthiness of PriModChain in terms of security, privacy, reliability, safety, and resilience. The experiments show that PriModChain is a feasible framework for privacy-preserving trustworthy machine learning and model sharing in IIoT systems.

The rest of the paper is organized as follows. The underlying concepts used in PriModChain are presented in Section II. Section III explains the steps employed in the proposed approach. The results of PriModChain are discussed in Section IV. Section V provides a summary of existing related work. The paper is concluded in Section VI.

II. Background

This section provides brief discussions on the underlying concepts used in PriModChain. We discuss the basic principles related to differential privacy, federated machine learning, interplanetary file system (IPFS), blockchain technology, Ethereum, and smart contracts.

A. *Differential Privacy*

Differential privacy (DP) is a privacy model that provides a strong level of privacy by minimizing the likelihood of individual record identification [10]. In principle, DP defines the limits to how much information about a data item can be made available to a third party for analysis. Traditionally, ε (epsilon) and δ (delta) are used to define these limits. Laplace and Gaussian mechanisms are the two most commonly used perturbation approaches in DP [11].

1) Definition of differential privacy: Let's take the dataset, $\mathcal{D}$ and two of its adjacent datasets, x and y, where y differs from x only by one data item. Assume, datasets x and y as being collections of records from a universe $\mathcal{X}$, where $\mathbb{N}$ denotes the set of all non-negative integers including zero. Then the randomized algorithm $\mathcal{M}$ satisfies (ε, δ)-differential privacy if it holds Eq. (1).

Definition 1: A randomized algorithm $\mathcal{M}$ with domain $\mathcal{N}^{|\mathcal{X}|}$ and range R is (ε, δ)-differentially private for $\delta \geq 0$ if for every adjacent x, $y \in \mathcal{N}^{|\mathcal{X}|}$ and for any subset $\mathcal{S} \subseteq \mathcal{R}$

$$Pr[(\mathcal{M}(x) \in \mathcal{S})] \leq \exp(\varepsilon) Pr[(\mathcal{M}(y) \in \mathcal{S})] + \delta. \quad (1)$$

2) Privacy budget (ε) and probability of error (δ): ε denotes the privacy budget, which provides an insight into the privacy rendered by a differentially private algorithm. The lower the value of ε, the lower the loss of privacy. δ is the probability of error/failure that accounts for "bad events" of an output revealing the identity of a particular individual. Hence, δ should be very small for a particular database.

3) Sensitivity (Δf): Sensitivity is defined as the maximum influence that a single data item can exert on the result of a numeric query of a function. Consider a function f, the sensitivity (Δf) of f can be given as in Eq. (2) where x and y are two neighboring databases and $\|.\|_1$ represents the $L1$ norm of a vector [12].

$$\Delta f = max\{\|f(x) - f(y)\|_1\} \tag{2}$$

4) Post processing invariance and composition: Postprocessing invariance/robustness property of DP ensures that the results of additional computations on an ε-DP outcome will still be ε-DP [13]. Composition property of DP is the degradation of privacy when multiple differentially private algorithms are performed on the same or overlapping datasets [13]. For example, when ε_1-DP, ε_2-DP,..., ε_n-DP are applied on the same or overlapping datasets, the union of the results is equal to $(\varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_n)$-DP [13].

B. Federated Learning

Federated learning is an approach to build machine learning models based on datasets that are distributed over multiple locations [14]. Assume that there are $\mathcal{N}$ data owners $\{\mathcal{O}_1, \dots \mathcal{O}_N\}$ each training their local models using the respective datasets $\{\mathcal{D}_1, \dots \mathcal{D}_N\}$. All the data owners $(\mathcal{O}_i)$ train models locally without exposing their local datasets $(\mathcal{D}_i)$ to other participating entities. The locally trained model parameters are then collected in a central server to federate into a global model, which is named as the federated model $(\mathcal{ML}_{fed})$. According to the definition of federated learning, the accuracy $(\mathcal{A}_{fed})$ of $\mathcal{ML}_{fed}$ should be very close to the accuracy $(\mathcal{A}_{ctr})$ of the model trained centrally with all the data [15]. This relationship can be represented using Eq. 3, where δ is a non-negative real number.

$$|\mathcal{A}_{fed} - \mathcal{A}_{ctr}| < \delta \tag{3}$$

C. The InterPlanetary File System (IPFS)

IPFS is a peer-to-peer distributed file system that provides a high throughput content-addressed block storage model with content-addressed hyperlinks (which is a unique hash value) [16]. Any modification to the file will destroy its original hash value, making the data saved in IPFS immutable. IPFS forms a generalized Merkle Directed Acyclic Graph (DAG) and combines a distributed hash table, an incentivized block exchange, and a self-certifying namespace to support building versioned file systems and blockchains [16].

D. Blockchains, Ethereum, and Smart Contracts

A blockchain is a distributed ledger of data records maintained by network nodes that are not owned by a central authority [17]. The blocks of data in the blockchain are chained to each other using cryptographic principles. The transaction data of a blockchain are immutable and public, which makes everyone accountable for their actions on the blockchain [17]. An application that is built on blockchain will automatically become transparent and resilient to attacks. Ethereum is an open-source blockchain platform for decentralized applications that control digital value. The programs that run on the Ethereum Virtual Machine (EVM) are referred to as "smart contracts". Solidity and Vyper are two of the popular languages that are used to write smart contracts on Ethereum[1].

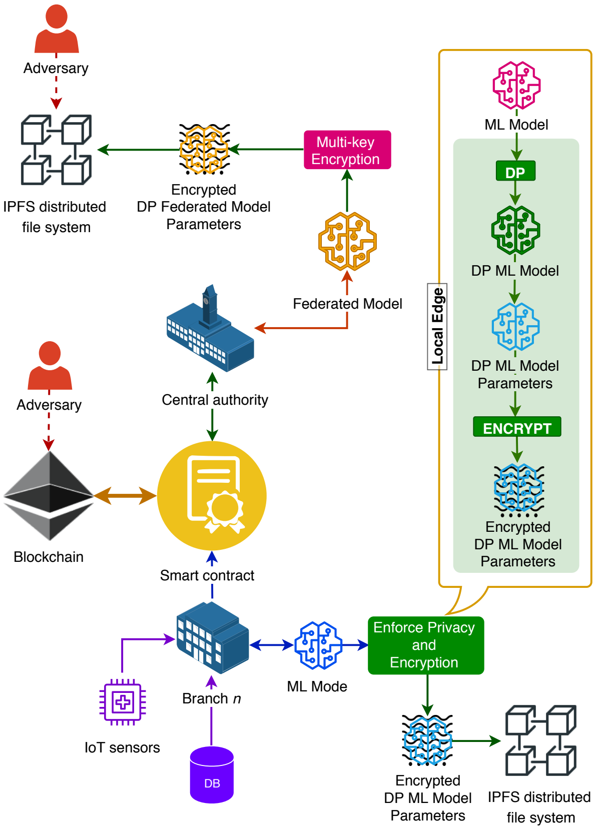

Adversary
IPFS distributed file system
Encrypted DP Federated Model Parameters
Multi-key Encryption
ML Model
DP
DP ML Model
Federated Model
Local Edge
DP ML Model Parameters
Central authority
Adversary
ENCRYPT
Encrypted DP ML Model Parameters
Blockchain
Smart contract
Enforce Privacy and Encryption
Branch *n*
ML Mode
IoT sensors
DB
Encrypted DP ML Model Parameters
IPFS distributed file system

Fig. 3: Modular arrangement of the main components of the proposed framework

III. Our Approach: PriModChain

This section discusses how the proposed framework blends the concepts of differential privacy, federated learning (FedML), Ethereum blockchains (EthBC), smart contracts, and the interplanetary file system (IPFS) to enforce privacy-preserving trustworthy distributed machine learning on IIoT-based Industry 4.0 systems. As available in any conventional Industry 4.0 based IIoT setting, PriModChain involves the two actors: (1) the distributed entity/ branch (DISTEN), and (2) the central authority/coordinating server (CENTAUTH). Fig. 3 shows how the smart contract, DISTEN, CENTAUTH, IPFS, and EthBC are organized in the PriModChain framework. We assume that each DISTEN is a full-scale factory with its own IIoT configuration. The DISTENs conduct differentially private ML model training and testing locally using the local data (both static data and stream data produced by IIoT). PriModChain uses FedML to generate a global representation of the ML models available at DISTENs by communicating the model parameters between the CENTAUTH and DISTENs. EthBC plays the role of keeping track of the consensus of the

[1] https://www.ethereum.org

contributions made by each actor during the model federation process. A smart contract maintains the coordination between DISTEN, CENTAUTH, IPFS, and EthBC.

Fig. 4 presents a layered architecture of PriModChain, where each layer concentrates on how different technologies are amalgamated to enforce different parameters for trustworthiness. The figure also depicts the on-chain and off-chain data storage selections preferred in each layer, where on-chain refers to storing data in EthBC, and off-chain refers to storing data in IPFS. PriModChain uses IPFS as the off-chain data storage mechanism since the ML model parameter datasets are too large to be stored on EthBC.

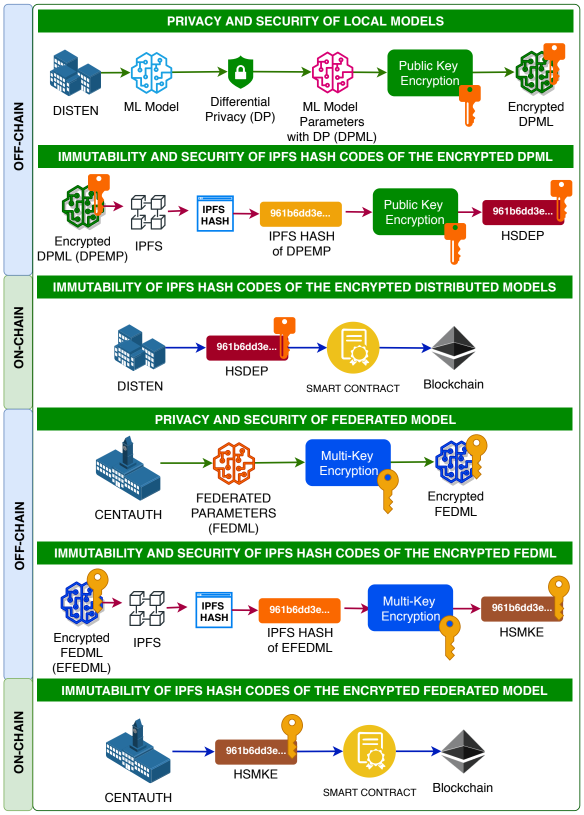


Fig. 4: A layered architecture of PriModChain and the trustworthiness properties enforced by each layer.

A. Composition of a DISTEN

In this section, we discuss the role of a DISTEN in generating a global representation of the distributed ML models. Algorithm 1 shows the sequence of execution of the responsibilities of a DISTEN. In the proposed experimental setup, each DISTEN trains a differentially private (refer section III-A1 for differentially private ML model generation) deep neural network (DNN) using the local data for a certain number of epochs (e.g. 30). After generating the differentially private ML model for local analysis, the DISTEN extracts the model parameters (e.g. trained weights) to be shared with the CENTAUTH. DISTEN applies public-key encryption on the model parameters using the public key ($\mathcal{P}_k^c$) generated by the CENTAUTH. The encrypted model parameters are then stored in IPFS, and a unique IPFS hash is obtained. The IPFS hash of the encrypted model parameters is then encrypted using the same public key $\mathcal{P}_k^c$ and added to the EthBC using the smart contract. The DISTEN's wallet key is then forwarded to the CENTAUTH to notify the completion of the local model training (at the corresponding DISTEN) for the current round of federation. After receiving the CENTAUTH notification about the completion of the current federation cycle, DISTEN retrieves the encrypted global model parameters. DISTEN then updates the local model using the global parameters decrypted using the multi-key (multi-key_$\mathcal{S}_k$) generated according to Algorithm 3. Then, DISTEN will start the next federation round by locally training the model with the updated model parameters, and repeating all the steps for a predefined number of federation rounds.

Algorithm 1: Role of a DISTEN in PriModChain for generating a federated global ML model

```
Input:  $\{x_1, \ldots, x_j\}$  ←  examples
        $\varepsilon$  ←  privacy budget
        $numeps$  ←  number of epochs
        $numrnds$  ←  number of federation rounds
        $\mathcal{P}_k^c$  ←  public key of CENTAUTH
Output:
        $DPML$  ←  differentially private local model
        $HSDEP$  ←  encrypted IPFS hash of $DPEMP$
        $\mathcal{P}_i^d$  ←  DISTEN's public key
generate a public-private key pair, ($\mathcal{P}_i^d$ = public key) ;
$fedcycnum$ = 0;
for $numrnds$ rounds do
    $fedcycnum = fedcycnum + 1$;
    train a differentially private ML model ($DPML$) for
      $numeps$ epochs using $\{x_1, \ldots, x_j\}$ (refer section
      III-A1);
    return $DPML$ for local analysis ;
    extract the model parameters from $DPML$;
    encrypt the model parameters using the $\mathcal{P}_k^c$ ;
    store the encrypted model parameters ($DPEMP$) in
      IPFS;
    encrypt the IPFS hash of $DPEMP$ using the $\mathcal{P}_k^c$ (refer
      section III-A2);
    add the encrypted hash ($HSDEP$) to Ethereum via the
      smart contract (fn_4() of Fig. 5);
    forward wallet address to notify CENTAUTH about the
      current round update;
    if CENTAUTH notified federated round update with
      multi-key_$\mathcal{S}_k$ and CENTAUTH $fedcycnum$ then
        if local $fedcycnum \leq$ CENTAUTH $fedcycnum$
          then
            use the multi-key_$\mathcal{S}_k$ to decrypt the IPFS hash of
              the global model parameter update;
            retrieve the global model parameters using the
              decrypted IPFS hash;
            use the multi-key_$\mathcal{S}_k$ to decrypt the model
              parameter updates from the CENTAUTH;
            load global parameters to the local model before
              next round;
```

1) Generating a differentially private ML model by a DISTEN: In order to apply differential privacy to an ML model generated by a DISTEN, PriModChain uses the differentially private stochastic gradient descent (DPSGD) mechanism proposed by Abadi et al. [5]. This method is based on the Gaussian noise mechanism, which is shown in Eq. 4 to enforce DP. ($f : \mathcal{D} \rightarrow \mathbb{R}$) is a real-valued function and $\mathcal{N}\left(0, S_f^2 \cdot \sigma^2\right)$ is the normal (Gaussian) distribution noise with mean 0 and standard deviation $S_f \sigma$, so that the noise is calibrated to f's sensitivity $S_f = |f(d) - f(d')|$, where d and d' are adjacent inputs. In DPSGD, at each step of SGD, they compute the gradient for a random subset of examples, clip the l_2 norm of each gradient, compute the average, and add noise in order to protect privacy.

$$\mathcal{M}(d) \triangleq f(d) + \mathcal{N}\left(0, S_f^2 \cdot \sigma^2\right) \tag{4}$$

Algorithm 2: Role of the CENTAUTH in PriModChain for generating a federated global ML model

```
Input:
    $\{HSDEP_1, \ldots, HSDEP_p\}$ ← IPFS hashes of $DPEMP$
    $\mathcal{P}_1^d, \ldots, \mathcal{P}_m^d$ ← public keys of DISTENs
    $numrnds$ ← number of federation rounds
    $\mathcal{T}_{fed}$ ← federation time interval
Output:
    $FEDMOD$ ← federated global model
    $HSMKE$ ← encrypted IPFS hash of encrypted global ML model parameters
    $\mathcal{P}_k^c$ ← CENTAUTH's public key
1  generate a public-private key pair, ($\mathcal{P}_k^c$ = public key) ;
2  $fedcycnum$ = 0;
3  for $numrnds$ rounds do
4      $fedcycnum = fedcycnum + 1$;
5      if minimum number of DISTEN updates are available then
6          retrieve the IPFS hashes of the model parameters $\{HSDEP_1, \ldots, HSDEP_p\}$ (pushed by DISTENs that are whitelisted and within $\mathcal{T}_{fed}$) from Ethereum via the smart contract (fn_7() of Fig. 5);
7          decrypt $\{HSDEP_1, \ldots, HSDEP_p\}$ using the private key;
8          retrieve the model parameters using decrypted IPFS hashes;
9          federate the parameters (trained weights) to obtain the global model (FEDML) parameter updates (refer section III-B1) ;
10         apply multi-key encryption (refer section III-B2) on the global model parameters using $\mathcal{P}_1^d, \ldots, \mathcal{P}_m^d$;
11         store the encrypted global model (EFEDML) in IPFS;
12         apply multi-key encryption (refer section III-B2) on the IPFS hash of the encrypted global model ;
13         add the multi-key encrypted IPFS hash ($HSMKE$) to Ethereum via the smart contract (fn_8() of Fig. 5);
14         forward the multi-key_$\mathcal{S}_k$ (using Algorithm 3) and $fedcycnum$ to notify DISTEN about the global model parameter update;
```

2) Public key encryption by a DISTEN: PriModChain uses RSA encryption scheme for the public key encryption in steps 8 and 10 in Algorithm 1. However, the choice of the public-key encryption algorithm is not restricted to RSA. We assume that all the participating entities preserve their private keys and do not leak them to any other party. In a possible event of a private key leak, the corresponding entity will be refreshed with a new key pair and will restart its responsibilities in PriModChain.

B. Composition of the CENTAUTH

In this section, we discuss the role of the CENTAUTH in generating a federated (global) ML model. As shown in Algorithm 2, the CENTAUTH first retrieves the encrypted IPFS hashes of the encrypted local (DISTEN) model parameters (from the whitelisted DISTENs) that are released within the federation interval ($\mathcal{T}_{fed}$). The model parameters are retrieved using their IPFS hashes decrypted using the private key of CENTAUTH. The decrypted model parameters are then federated by averaging the model parameters (i.e. weight matrices), as proposed by McMahan et al. [18] (refer Section III-B1 for more details about the model parameter federation). As discussed in Section III-B2, the CENTAUTH applies multi-key encryption on the global model parameters using the public keys of all DISTENs (who are in the whitelist as discussed in Section III-C). The encrypted global parameters are then stored in IPFS. The CENTAUTH applies the same multi-key encryption protocol on the IPFS hash of the encrypted global parameters according to Section III-B2. The DISTENs are then acknowledged about the global parameter update by the CENTAUTH, forwarding multi-key_$\mathcal{S}_k$ and CENTAUTH $fedcycnum$.

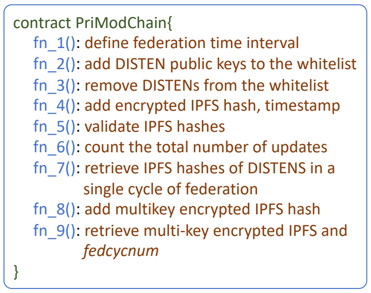
Algorithm 3: Multi-key encryption in PriModChain

```
Input:
    $\mathcal{P}_1^d, \ldots, \mathcal{P}_m^d$ ← public keys of DISTENs
    $\mathcal{T}_{fed}$ ← federation time interval
Output:
    multi-key_$S_k$ ← multi-key of the current federation cycle
1  generate a symmetric key, $S_k$ randomly for the current $\mathcal{T}_{fed}$;
2  encrypt the global model parameters generated in step 9 in Algorithm 2 using $\mathcal{S}_k$;
3  encrypt the IPFS hash generated in step 11 in Algorithm 2 using $\mathcal{S}_k$;
4  retrieve the public keys $\mathcal{P}_1^d, \ldots, \mathcal{P}_m^d$ of DISTENs in the whitelist;
5  for $\mathcal{P}_i^t \in \mathcal{P}_1^d, \ldots, \mathcal{P}_m^d$ do
6      encrypt $S_k$ using $\mathcal{P}_i^t$;
7      pass the encrypted $S_k$ (multi-key_$S_k$) to the corresponding DISTEN;
```

1) Federated model parameter update: A federated model parameter update, w_{t+1} is given by Eq. 5, where w_t is the current model parameter state, and p is the number of DISTENs contributed within the federation time interval [18]. $\triangle w^i = w^i - w^t$ is the difference between the optimized local model and the central model. At the end of the federation cycle, $\triangle w^i$ will be considered as the i^{th} federated model parameter update. As explained in Section II-A4, post-processing invariance property of differential privacy guarantees that the federated model is also differentially private as all the local models (of DISTENs) are differentially private.

$$w_{t+1} = w_t + \frac{1}{p}\left(\sum_{i=0}^{p} \triangle w^i\right) \tag{5}$$

2) Multi-key encryption: In order to apply multi-key encryption in steps 10 and 12 of Algorithm 2, PriModChain uses Algorithm 3. As shown in the algorithm, multi-key encryption uses a randomly generated symmetric key during the encryption. All the DISTENs in the whitelist, are notified with the encrypted symmetric key (multi-key_S_k) via the blockchain through the smart contract.



```
contract PriModChain{
    fn_1(): define federation time interval
    fn_2(): add DISTEN public keys to the whitelist
    fn_3(): remove DISTENs from the whitelist
    fn_4(): add encrypted IPFS hash, timestamp
    fn_5(): validate IPFS hashes
    fn_6(): count the total number of updates
    fn_7(): retrieve IPFS hashes of DISTENS in a
            single cycle of federation
    fn_8(): add multikey encrypted IPFS hash
    fn_9(): retrieve multi-key encrypted IPFS and
            fedcycnum
}
```

Fig. 5: The list of functions in the smart contract

C. Smart Contract

The smart contract plays a vital role in PriModChain in coordinating and administrating the ML knowledge sharing process. The CENTAUTH has higher privileges to the functions in the smart contract compared to a DISTEN. Fig. 5 shows an overview of the PriModChain smart contract. fn_1() is used to define a federation time interval. fn_1() can be accessed only by CENTAUTH to make changes to the federation interval. CENTAUTH can declare a suitable federation interval based on its capacity and workload of federating models. The federation interval should be defined by considering the constraints in Eq. 6, where $\Delta\mathcal{T}_{fed}$ is the federation interval, $\mathcal{T}_{Ethts}$ is the Ethereum transaction time, $\mathcal{T}_{dist}$ is the local model training time, $\mathcal{T}_{cent}$ is the processing time for a single federation in CENTAUTH, and $\mathcal{T}_{other}$ is the other latencies such as encryption-decryption delays and network delays such from IPFS.

$$\Delta\mathcal{T}_{fed} = \mathcal{T}_{dist} + \mathcal{T}_{Ethts} + \mathcal{T}_{cent} + \mathcal{T}_{other} \tag{6}$$

fun_2() is another CENTAUTH privileged function, which is used to add DISTENs' public keys to the whitelist. A DISTEN has to be in the whitelist to contribute to the model federation process. The CENTAUTH uses func_3() to remove DISTENs from the whitelist. fn_4() is used by DISTENs to add the encrypted IPFS hashes (of the locally trained model parameters) to Ethereum. fn_5() is used by the CENTAUTH to retrieve the DISTENs in the whitelist to validate the authenticity of the encrypted IPFS hashes. CENTAUTH uses fun_6() to count the number of DISTEN updates available in a particular federation cycle number. fn_7() is used by the CENTAUTH to retrieve the encrypted IPFS hashes of the model parameters, which correspond to a particular federation cycle (within the federation interval). fun_8() is used by CENTAUTH to add the multi-key encrypted IPFS hash of the global model to the blockchain. DISTENs use fn_9() to retrieve the muli-key encrypted IPFS hash of the global model.

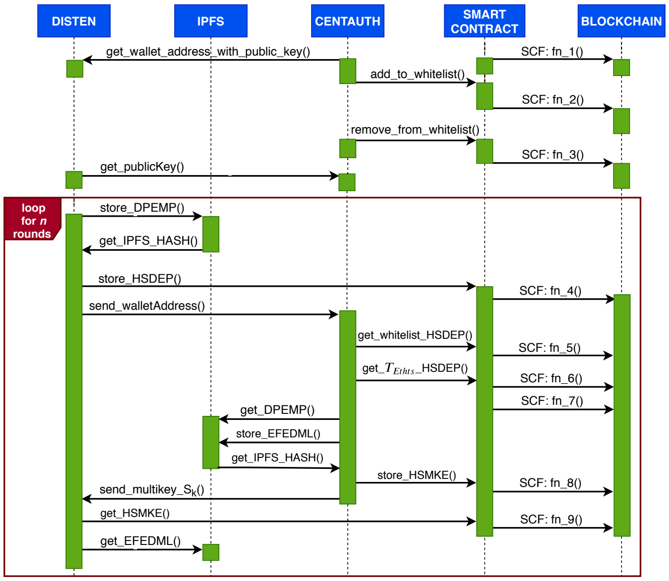


Fig. 6: Sequence of function calls in PriModChain

Fig. 6 is the sequence diagram, which shows the flow of function calls between the five main entities (DISTEN, CENTAUTH, IPFS, Smart Contract, and Blockchain) of PriModChain. The prefix SCF is used to abbreviate the "smart contract functions". The names of the function calls are self-explanatory and follow the explanations given under Section III. n in the iteration module represents the number of federation rounds declared during the initialization of PriModChain.

IV. Results and Discussion

In this section, we discuss the experiments, experimental configurations, and the results of PriModChain. We simulated PriModChain and conducted experiments upon it on a MacBook Pro (macOS Mojave, 13-inch, 2017) computer with Intel Core i5 CPU (2.3 GHz), 8 GB RAM and 1536MB GPU (Intel Iris Plus Graphics). The MNIST dataset [5] was selected for the experiments as it is benchmarked as a reliable dataset that produces good accuracy for deep learning. We can use this property of MNIST to investigate the dynamics of different modules and parameters of PriModChain, such as model convergence, and ε selection (for differential privacy) explicitly. A more complex dataset would introduce challenges towards the assessment of the principal PriModChain parameters such as privacy, accuracy, and ML model convergence. The MNIST dataset is composed of 70,000 grayscale handwritten digits (which corresponds to 10 classes/numbers), where an image has a resolution of 28x28. We chose a convolutional neural network (CNN) as the choice of the ML algorithm in testing PriModChain. The CNN accepts 28 × 28 input images. It has two convolutional layers with ReLU activation functions, one max pooling layer with 2×2 max pools, a fully connected layer with 128 neurons with ReLU activation function, and a fully connected layer with 10 neurons which produces the output, that corresponds to the 10 classes of the MNIST dataset.

A. *Experimental setup*

Fig. 7 shows the arrangement of the component in the experimental setup of PriModChian. We used Python (version 3.6.5) as the primary programming language to develop the programs in CENTAUTH and DISTENs. We used python socket and _thread interfaces to simulate the communications between DISTENs and CENTAUTH in the federated learning setup. Solidity v0.5.0 was used to implement the PriModChian smart contract. The smart contract was deployed to the EthBC networks using Truffle v5.0.24. For the local experiments on the blockchain, we used the Ganache v2.0.1 local test network. Kovan test network was used as the public blockchain for the PriModChain's experiments. PriModChain was connected to Kovan through Infura, which is a hosted Ethereum node cluster that lets running applications without needing a personal Ethereum node. Python cryptography v2.3.1 package was used for the RSA encryption-decryption (using cryptography.hazmat) scenarios in PriModChain. The main python programs communicate with the smart contract through the Web3.py library, which interacts with the smart contracts through their ABIs (application binary interface). The main programs of CENTAUTH and DISTENs communicate with IPFS (go-ipfs v0.4.21) connected through ipfsapi python library for model parameter exchange and storage.

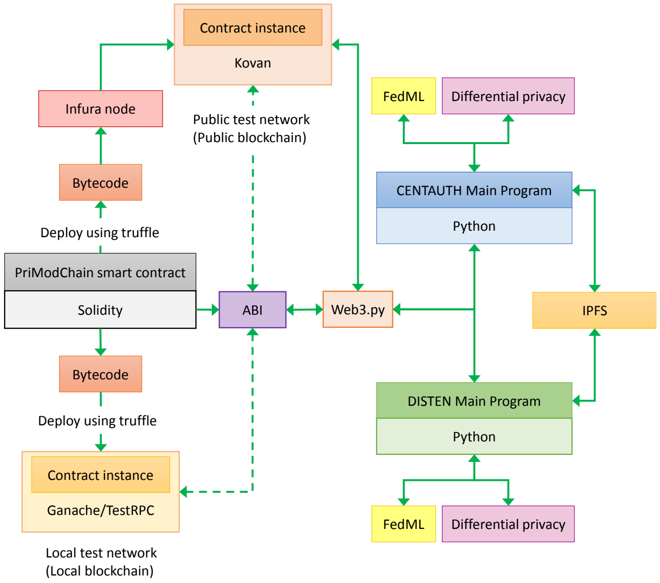


Fig. 7: The arrangement of the PriModChain components in the experimental setup

B. *Simulation results of PriModChain on trustworthiness*

In this section, we explain how PriModChain enforces the properties (refer Fig. 2) of a trustworthy IIoT system by presenting the experimental results under each pillar of trustworthiness (security, privacy, safety, resilience, and reliability).

1) *Security in PriModChian:*

In this section, we verify the multi-key encryption security protocols used in PriModChain to confirm that it does not suffer from any unanticipated security vulnerabilities. The multi-key encryption protocol needs verification due to its unique approach to securing the federated ML model data.

a) Security verification of the multi-key encryption protocol (Algorithm 3): We used Scyther v1.1.3 [19] for the security verification of the multi-key encryption protocol. Scyther is a security protocol verifier which is based on a pattern refinement algorithm. It provides concise representations of (infinite) sets of traces. The encryption data (e.g. keys) communication of the protocol presented in Algorithm 3 can be given by the following steps where Epub$_{cent}$ represents encrypting using the public key ($\mathcal{P}_k^c$) of the CENTAUTH, Epub$dist_1$ represents encrypting using the public key ($\mathcal{P}_i^d$) of the corresponding DISTEN who contributed to the current federation cycle, and E_S_k represents encrypting using the symmetric key. DIST$_{Model_i}$ represents the model that was generated by a particular DISTEN. FED$_{Model}$ represents the federated model. S_k represents the randomly generated session key (symmetric key) of a particular federation cycle.

Phase 1
DISTEN1 sends Epub$_{cent}$(DIST$_{Model_1}$) to CENTAUTH
...
DISTENn sends Epub$_{cent}$(DIST$_{Model_n}$) to CENTAUTH

Phase 2
CENTAUTH broadcasts [E_S_k(FED$_{Model}$)]
CENTAUTH sends [Epub$dist_1$(S_k), ... ,
Epub$dist_n$(S_k)] to DISTEN1...DISTENn

Table I shows the results returned on seven security claims. The first three claims (Secret_mod_i, Secret_symkey, Secret_fedmod) check whether the local models, the symmetric key ($\mathcal{S}_k$), and the federated model are kept secret by the DISTENs and the CENTAUTH. Alive or aliveness (of all roles) checks whether the responder has previously been running the protocol, when an initiator agent completes a run of the protocol, as defined in [20]. Weakagree checks for weak agreement (of all roles) as defined in [20]. Niagree checks for non-injective agreement on messages as defined in [21]. Nisynch checks for non-injective synchronization as defined in [21]. symkey is the session key generated during a federation cycle, mod_i represents a model generated by a DISTEN, and fedmod is the federated model. As shown in Table I, the multi-key protocol used in Algorithm 3 does not leak any information to third parties.

TABLE I: Protocol verification results

User	Claim	Verification status comment
DISTEN / CENTAUTH	Secret_mod_i	no attack within bounds
	Secret_symkey	no attack within bounds
	Secret_fedmod	no attack within bounds
	Alive	proof of correctness
	Weakagree	proof of correctness
	Niagree	does not occur
	Nisynch	does not occur

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/TII.2020.2974555, IEEE Transactions on Industrial Informatics

2) Privacy in PriModChain: PriModChain enforces privacy on data using differential privacy. However, federated learning also provides a certain level of privacy as the local data are not directly shared with the distributed entities. In this section, we investigate the level of privacy enforced by a DISTEN and the CENTAUTH separately.

a) Privacy of a local ML model: A DISTEN releases only the privacy-preserving version of the model parameters of a locally generated differentially private ML model. Differential privacy guarantees that the model parameters do not leak privacy. Additionally, the parameters alone do not allow adversaries to derive the architectural properties of the underlying ML model. As the model parameters are encrypted using public-key encryption, the model needs to be decrypted before any privacy attack, which makes the attacks even more difficult.

b) Privacy of the global model: Since the local models (generated at DISTENs) are differentially private; the federated global model is also differentially private due to the post-processing invariance property (refer Section II-A4). Additionally, the global model parameters are encrypted using a unique session key (S_k), which is created randomly for each federation cycle for the corresponding federation time interval ($\mathcal{T}_{fed}$). S_k is protected using the multi-key protocol explained in Section III-B2. Even leaking S_k does not allow an adversary to retrieve private information from the global/federated ML model due to differential privacy of the model parameters, and unavailability of the details on the ML model architecture. As discussed in Section II-A4, the privacy budgets add up when the global model is generated based on local models trained on the same or overlapping datasets. However, in PriModChain, we consider a horizontal federation setup where there is no overlapping on the datasets, and each DISTEN presents a unique dataset.

3) Safety and Resilience in PriModChain: As discussed in Section II-A, differential privacy enforces a strong privacy guarantee on the data, whereas data encryption strengthens the safety of data in PriModChain. Consequently, any adversarial attack on the PriModChain ML models will not reveal private information; the data privacy will remain safe on any catastrophic situation of security exploitation in PriModChain. Additionally, EthBC guarantees the resilience of the framework as it keeps a transparent log of all the events. Any undesirable event can be tracked and recovered effectively by identifying the exact point of failure. Moreover, the randomness of the encryption key generation process makes it even harder for adversaries to crack PriModChain for misuse.

4) Reliability in PriModChain: We investigated the reliability of PriModChian in terms of accuracy, transaction cost (Ethereum), latency, and real-time data processing capabilities.

a) Accuracy against the change in privacy during the federation: Fig. 8 shows the change of accuracy of the ML model after each round of model federation. We considered a varying number of DISTENs (2 to 5), each applying ($\varepsilon = 4$, $\delta = 10^{-5}$)-differential privacy to the local models. At the start and after each round of federation, the ML models were locally trained for 30 epochs. As shown in Fig. 8, the accuracy of the global model improves with the number of federation rounds. However, the exact number of federations necessary can vary based on the number of DISTENs, the input data, and the underlying architecture of the ML model. The input dataset was equally divided between the DISTENs. Consequently, the higher the number of DISTENs, the lower the number of tuples in each DISTEN. When the number of DISTENs is high, the starting accuracy is lower, and the time taken for the model convergence becomes high. This also results in an accuracy drop in the second epoch as the second federation round can introduce an extensive parameter modification after the first epoch.

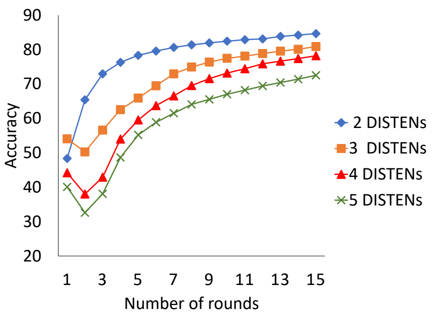


Fig. 8: Accuracy vs. the number of rounds of federation

TABLE II: Ethereum transaction costs of the most frequent operations of PriModChain (1000000000000000000 Wei = 1 $Ether$ = 132.17 USD)

Function	Gas used (wei)
step 11, Algorithm 1	36112
step 6, Algorithm 2 (for 10 addresses)	42334
step 13, Algorithm 2	36112
add a user to the whitelist	23796
remove a user from the whitelist	14276

b) Transaction cost analysis: It costs around 2563999 Wei to deploy the PriModChain smart contract to the Ethereum network. However, a particular organization has to execute this step only once in the PriModChain life cycle. Table II includes the Ethereum transaction costs of the most frequent operations (function) of PriModChain. As per the current exchange rates, the transaction costs are minimal in terms of USD.

TABLE III: The parameter estimates of the federation interval

Latencies	Time
$\mathcal{T}_{dist}$	∼120-150 seconds (for 5000 tuples of MNIST, with a batch size of 64). However, this depends on the choice of the ML model, size of the dataset, and its architecture (10 seconds - a few hours)
$\mathcal{T}_{Ethts}$	∼15-30 seconds
$\mathcal{T}_{cent}$	∼3-10 seconds
$\mathcal{T}_{other}$	∼10-30 seconds
TOTAL	∼148-220 seconds

c) Federation Interval: As shown in Table III, the main factor that governs the federation interval is the local model generation time (at a DISTEN). This is governed by the parameters such as number of data tuples, number of epochs, batch size, and learning rate. By adjusting these parameters, the local model generation time can be adjusted to meet the demands of an industrial environment. During the experiments, we considered a federation interval proportional to the local model generation time (e.g. 300 seconds for 5000 tuples and 1500 seconds for 60,000 tuples).

d) Real time data stream processing capacity of PriModChain: In the proposed setting of PriModChian, the distributed entities work with both static and stream IIoT data, as shown in Fig. 3. After buffering a certain number of tuples, DISTENs conduct local model training before each round of federation, to generate a local model for a given number of epochs. Next, the trained parameters are passed to the CENTAUTH using the trustworthy approach formulated in PriModChain. As discussed in Section IV-B4c, one round of federation takes as low as 148 seconds to as high as a few hours. This latency is used by the DISTENs to buffer new records through the connected data streams. As a result, PriModChain can accept infinite data streams, and due to the large window of the data buffer, PriModChain can work on data streams with high speeds, given the memory of a DISTEN is large enough to hold data with high capacity. Fig. 9 shows the time consumption when we increment the number of tuples in one DISTEN where a total of 2 DISTENs are used. Each DISTEN trained the model locally for 30 epochs under a batch size of 64. As the figure shows, the time consumption shows a linear pattern, which suggests that PriModChain is a feasible solution towards large scale machine learning.

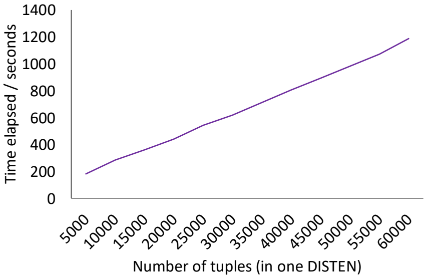


Fig. 9: Effect of the number of tuples (per DISTEN) on the time consumption

V. Related Work

IIoT and related innovations such as Industry 4.0 are motivated to use the vast distribution and heterogeneity of the entire industrial value chain to encompass business advantages in the competitive market. Although this can introduce many advantages, the extensive integration of heterogeneous technologies and concepts introduce trustworthiness issues in internal actions and communications [22]. The importance of trustworthiness in a sub-system or a system should be looked at from different dimensions, which involve quantification/measurement, standardizations/certifications, and deployment of state-of-the-art cybersecurity frameworks and standards. A trustworthiness level matrix is an example of a theoretical measurement that tries to measure the degree of trustworthiness required from a component, a composed sub-system, or a system [22]. Cybersecurity frameworks for IIoT systems include the National Institute of Standards and Technology (NIST) framework for infrastructure cybersecurity [23], and the European Union Agency for Network and Information Security baseline security recommendations for IoT [22]. The standards which are applicable for IIoT and Industry 4.0 include ISA/IEC 62443 and OWASP [22].

It was identified that security, privacy, reliability, safety, and resilience are the five pillars of a trustworthy IIoT system [9]. To enhance these pillars for machine learning in IIoT, we investigated the systematic amalgamation between smart contracts, Ethereum blockchain [17], differential privacy [10], federated learning [14], and interplanetary file system (IPFS) [16]. The application of blockchain in various areas has become popular due to its underlying properties such as immutability, traceability, and security [24], [25]. Nikolay et al. proposed a blockchain-based information sharing platform for IIoT trust [26]. Jiafu et al. developed a blockchain-based solution for enhancing security and privacy in smart factory [25]. This method uses smart contracts for processing and storing information. Zhetao et al. used a consortium blockchain for secure energy trading in IIoT [27]. However, these methods failed to look at privacy as one of the essential components of a trustworthy IIoT system for machine learning. Differential privacy (DP) is the most preferred privacy model as it enforces a strong privacy guarantee on the underlying data [28], [29]. Laplace mechanism, Gaussian mechanism [30], geometric mechanism, randomized response [31], and staircase mechanisms [10] are a few of the fundamental mechanisms used to achieve differential privacy. Chamikara et al. proposed a method that utilizes differential privacy for IoT streams [32]. Rongxing et al. proposed a lightweight privacy-preserving data aggregation scheme for computing-enhanced IoT [33]. Muneeb et al. discussed the implementation of privacy-preservation strategies in blockchain-based IoT systems using differential privacy [34]. However, the existing approaches fail to provide a complete solution for trustworthy IIoT machine learning.

VI. Conclusion

We proposed a new framework named PriModChain that can be used for trustworthy machine learning and sharing in an IIoT setting. PriModChain amalgamates the concepts of smart contracts, blockchain, federated learning, differential privacy, and interplanetary file system (IPFS) to enforce privacy and trustworthiness on ML in IIoT. Federated learning is used as the global ML model federation and sharing approach, while differential privacy enforces privacy on the ML models. The integration of smart contracts and the Ethereum blockchain introduce traceability, transparency, and immutability to the framework. IPFS introduces immutability, low latency, and fast decentralized archiving with secure P2P content delivery. The proposed framework was tested for its feasibility in terms of privacy, security, reliability, safety, and resilience. PriModChian generates excellent results towards the five pillars of trustworthiness and proves to be a feasible solution for trustworthy privacy-preserving machine learning in IIoT systems.

One of the potential future directions of the proposed work is to investigate different approaches to reduce latency to improve efficiency.

References

[1] R. Iqbal, T. Maniak, F. Doctor, and C. Karyotis, "Fault detection and isolation in industrial processes using deep learning approaches," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 3077–3084, 2019.

[2] S. A. Shevchik, G. G. Masinelli, C. Kenel, C. Leinenbach, and K. Wasmer, "Deep learning for in situ and real-time quality monitoring in additive manufacturing using acoustic emission," *IEEE Transactions on Industrial Informatics*, 2019.

[3] M. S. Hossain, M. Al-Hammadi, and G. Muhammad, "Automatic fruit classification using deep learning for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1027–1034, 2018.

[4] R. S. Peres, A. D. Rocha, P. Leitao, and J. Barata, "Idarts–towards intelligent data analysis and real-time supervision for industry 4.0," *Computers in Industry*, vol. 101, pp. 138–146, 2018.

[5] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 308–318.

[6] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017, pp. 3–18.

[7] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 587–601.

[8] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 1322–1333.

[9] F. Fraile, T. Tagawa, R. Poler, and A. Ortiz, "Trustworthy industrial iot gateways for interoperability platforms and ecosystems," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4506–4514, 2018.

[10] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," in *Advances in neural information processing systems*, 2014, pp. 2879–2887.

[11] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "Local differential privacy for deep learning," *IEEE Internet of Things Journal*, 2019.

[12] Y. Wang, X. Wu, and D. Hu, "Using randomized response for differential privacy preserving data collection." in *EDBT/ICDT Workshops*, vol. 1558, 2016.

[13] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Theory of Cryptography Conference*. Springer, 2016, pp. 635–658.

[14] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," 2016.

[15] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, p. 12, 2019.

[16] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.

[17] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, and V. Santamaría, "Blockchain and smart contracts for insurance: Is the technology mature enough?" *Future Internet*, vol. 10, no. 2, p. 20, 2018.

[18] H. B. McMahan, E. Moore, D. Ramage, S. Hampson *et al.*, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.

[19] C. J. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols," in *International Conference on Computer Aided Verification*. Springer, 2008, pp. 414–418.

[20] G. Lowe, "A hierarchy of authentication specifications," in *Proceedings 10th Computer Security Foundations Workshop*. IEEE, 1997, pp. 31–43.

[21] C. Cremers and S. Mauw, "Security properties," in *Operational Semantics and Verification of Security Protocols*. Springer, 2012, pp. 37–65.

[22] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (iiot): An analysis framework," *Computers in Industry*, vol. 101, pp. 1–12, 2018.

[23] A. Sedgewick, "Framework for improving critical infrastructure cybersecurity, version 1.0," Tech. Rep., 2014.

[24] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd International Conference on Open and Big Data (OBD)*. IEEE, 2016, pp. 25–30.

[25] J. Wan, J. Li, M. Imran, D. Li *et al.*, "A blockchain-based solution for enhancing security and privacy in smart factory," *IEEE Transactions on Industrial Informatics*, 2019.

[26] N. Teslya and I. Ryabchikov, "Blockchain-based platform architecture for industrial iot," in *2017 21st Conference of Open Innovations Association (FRUCT)*. IEEE, 2017, pp. 321–329.

[27] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial internet of things," *IEEE transactions on industrial informatics*, vol. 14, no. 8, pp. 3690–3700, 2017.

[28] C. Dwork, "The differential privacy frontier," in *Theory of Cryptography Conference*. Springer, 2009, pp. 496–502.

[29] N. Mohammed, R. Chen, B. Fung, and P. S. Yu, "Differentially private data release for data mining," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 493–501.

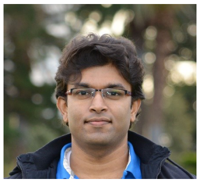
[30] T. Chanyaswad, A. Dytso, H. V. Poor, and P. Mittal, "Mvg mechanism: Differential privacy under matrix-valued query," *arXiv preprint arXiv:1801.00823*, 2018.

[31] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren, "Heavy hitter estimation over set-valued data with local differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 192–203.

[32] M. Chamikara, P. Bertok, D. Liu, S. Camtepe, and I. Khalil, "An efficient and scalable privacy preserving algorithm for big data and data streams," *Computers & Security*, p. 101570, 2019.

[33] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced iot," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.

[34] M. U. Hassan, M. H. Rehmani, and J. Chen, "Privacy preservation in blockchain based iot systems: Integration issues, prospects, challenges, and future research directions," *Future Generation Computer Systems*, vol. 97, pp. 512–529, 2019.



M.A.P. Chamikara is a Ph.D. researcher in Computer Science and Software Engineering at the School of Science, RMIT University, Australia. He is also a researcher at CSIRO Data61, Melbourne, Australia. He received his M.Phil. in Computer Science from the University of Peradeniya, Sri Lanka in 2015. His research interests include information privacy and security, data mining, artificial neural networks, and fuzzy logic.

P. Bertok is an associate professor in the School of Science at RMIT University, Melbourne, Australia, where he is a member of the Cyberspace and Security Group (CSG). He received his Ph.D. in computer engineering from the University of Tokyo, Japan. His research interests include access control, privacy protection and communication security.

I. Khalil is an associate professor in the School of Science at RMIT University, Melbourne, Australia. Ibrahim obtained his Ph.D. in 2003 from the University of Berne in Switzerland. Before joining RMIT University Ibrahim also worked for EPFL and University of Berne in Switzerland and Osaka University in Japan. He has several years of experience in Silicon Valley based companies working on Large Network Provisioning and Management software. His research interests are in scalable efficient computing in distributed systems, network and data security, secure data analysis including big data security, steganography of wireless body sensor networks and highspeed sensor streams and smart grids.

D. Liu is a senior research scientist at CSIRO Data61. He received his Ph.D. in Computer Science and Engineering from Shanhai Jiao Tong University, China. Dongxi Liu joined CSIRO in March 2008. Before that, he was a Researcher in the University of Tokyo from Feb 2004 to March 2008, and a Research Fellow in National University of Singapore from December 2002 to December 2003. His current research focuses on lightweight encryption for IoT security and encrypted data processing for cloud security.

S. Camtepe is a principal research scientist at CSIRO Data61. He received his Ph.D. in computer science from Rensselaer Polytechnic Institute, New York, USA, in 2007. From 2007 to 2013, he was with the Technische Universitaet Berlin, Germany, as a Senior Researcher and Research Group Leader in Security. From 2013 to 2017, he worked as a lecturer at the Queensland University of Technology, Australia. His research interests include mobile and wireless communication, pervasive security and privacy, and applied and malicious cryptography.

M. Atiquzzaman received the MS and PhD degrees in electrical engineering and electronics from the University of Manchester, United Kingdom. He currently holds the Edith Kinney Gaylord Presidential professorship in the School of Computer Science at the University of Oklahoma. He is the editor-in-chief of Journal of Networks and Computer Applications, founding editor-in-chief of Vehicular Communications and has served/serving on the editorial boards of various IEEE journals and co-chaired numerous IEEE international conferences including IEEE Globecom. His research interests are in communications switching, transport protocols, wireless and mobile networks, satellite networks, and optical communications. His research has been funded by the National Science Foundation, National Aeronautics and Space Administration, U.S. Air Force, Cisco, and Honeywell. He is a senior member of the IEEE.