On-Demand Security Framework for 5GB Vehicular Networks

Abdelwahab Boualouache*, Bouziane Brik**, Sidi-Mohammed Senouci**, and Thomas Engel*
(*) Faculty of Science, Technology and Medicine, University of Luxembourg, Luxembourg
(**) DRIVE Lab, University of Bourgogne, Nevers, France
Email: (*) {abdelwahab.boualouache, thomas.engel}@uni.lu}
(**) {bouziane.brik, sidi-mohammed.senouci}@u-bourgogne.fr

***Abstract*—Building accurate Machine Learning (ML) attack detection models for 5G and Beyond (5GB) vehicular networks requires collaboration between Vehicle-to-Everything (V2X) nodes. However, while operating collaboratively, ensuring the ML model's security and data privacy is challenging. To this end, this article proposes a secure and privacy-preservation on-demand framework for building attack-detection ML models for 5GB vehicular networks. The proposed framework emerged from combining 5GB technologies, namely, Federated Learning (FL), blockchain, and smart contracts to ensure fair and trusted interactions between FL servers (edge nodes) with FL workers (vehicles). Moreover, it also provides an efficient consensus algorithm with an intelligent incentive mechanism to select the best FL workers that deliver highly accurate local ML models. Our experiments demonstrate that the framework achieves higher accuracy on a well-known vehicular dataset with a lower blockchain consensus time than related solutions. Specifically, our framework enhances the accuracy by 14% and decreases the consensus time, at least by 50%, compared to related works. Finally, this article discusses the framework's key challenges and potential solutions.**

***Index Terms*—5G and Beyond Vehicular Networks; Security and Privacy; Federated Learning; Blockchain**

I. Introduction

The fifth-generation mobile communications networks (5G) are revolutionizing our daily life by enabling new applications with new requirements, including extensive coverage, high bandwidth, and ultra-reliable, low-latency communications. These advances have allowed significant developments in many life domains, such as transportation, agriculture, and health. Vehicular networks are one of the leading transportation applications witnessing tremendous advances. Thanks to 5G, Connected and Automated Vehicles (CAVs) provide more road safety and an enjoyable driving experience. However, as technological levels increase, vulnerabilities also increase. 5G-enabled CAVs are facing a massive vector of attacks that can lead to hazardous situations for drivers and passengers [1]. More specifically, internal attacks such as message droppings, denial of service, and position falsification pose a real danger since attackers are authenticated members, making them resistant to cryptographic solutions [2]. Fortunately, recent advances in Machine Learning (ML) have led to interesting solutions to cope with these attacks. Several ML-based Misbehavior Detection Systems (ML-based MDSs) have been proposed to detect internal attackers efficiently. However, most are centrally trained, limiting their detection capabilities, especially for unseen malicious behaviors (e.g., zero-day attacks). Indeed, suppose a new attack behavior appears. In that case, the systems will not be able to detect the attack until sufficient security datasets are collected and the central entity retrains the model and redeploys it. On the other hand, collaborative ML-based MDSs offer a continuous accuracy evolution and flexibility to detect unseen attacks. However, several limitations exist in early proposed collaborative ML-based MDSs [3–5]. In particular, they: (i) violate CAVs' privacy preservation since the training datasets, with private information, are shared among learning nodes, and (ii) generate a large overhead during ML model updates, which can increase communication latency. To deal with these issues, a few ML-based MDSs [6–10] have recently exploited Federated Learning (FL). FL is a distributed ML approach that allows learning nodes (FL workers) to jointly train a global model without sharing their datasets with the FL server. Thus, FL can avoid overhead and mitigates privacy risks [11]. However, each of the existing FL-based MDSs for 5GB vehicular networks has limitations. More specifically, (i) [6–8] is vulnerable to single-point-of-failure and attack risks since one single learning node (FL server) aggregate the global model; (ii) [9] is vulnerable to Byzantine attacks and ML-model falsification attacks [12] since local models are not verified and securely saved, and (iii) [10] leverages a heavy blockchain-based security infrastructure with a limited consensus algorithm.

To this end, this article comes to overcome the above-mentioned limitations. This mainly comprises avoiding single-point-of-failure and attacks on the FL learning process while providing blockchain-based lightweight security. We propose thus a novel on-demand, decentralized, and secure security framework for 5GB vehicular networks. The proposed framework leverages a scalable and lightweight edge-based blockchain infrastructure, FL, and smart contracts to enable secure and privacy preservation collaboration in building FL global attack detection models while avoiding Byzantine attacks. The framework also has a game-theoretic-based incentive mechanism for selecting the best FL workers in each FL round. The article also includes a comparative analysis performance evaluation validating the framework's building blocks. Finally, it discusses the framework's challenges and potential solutions. We can summarize the main contributions of this article as follows:

- We propose an on-demand security framework for 5GB vehicular networks in which building ML models for attack detection are made available to the stakeholders as needed. This framework is also modular with four building blocks: (i) edge-based blockchain system, (i) smart contract Design, (iii) accuracy-based fault tolerance consensus Protocol, and (iv) Game theoretic-based incentive mechanism.
- We examine our framework's key features compared to state-of-the-art works and evaluate its performance through three experiments regarding model accuracy, consensus time, and incentives.

The remainder of this paper is organized as follows. Section II discusses related works and their limitations. Section III describes the building blocks of the framework. Section IV describes the framework workflow and how the system works. Section V performs a comparative analysis and evaluates the performance of the framework's cornerstones. Section VI discusses the challenges of implementing this framework and potential solutions. Section VII concludes the paper.

II. State-of-the Art: Collaborative ML-based Misbehavior Detection Systems for 5GB Vehicular Networks

Several collaborative ML-based MDSs have been proposed for detecting internal attacks in 5GB vehicular networks. The authors of [3] presented a detection system that enables (re)training of the global model based on data newly collected in a cluster of servers instead of one server. The authors of [4] proposed a collaborative detection system based on Software Defined Networking (SDN), enabling multiple distributed SDN controllers to train a global model. However, both [3] and [4] have privacy preservation issues since datasets are shared between learning nodes. The authors of [5] proposed a distributed ML approach that enables CAVs to train a global model without sharing their datasets. However, peer-to-peer distributed learning generates a large overhead, degrading communications performance. The authors of [6] proposed an FL-based privacy preservation collaborative attack detection system. The authors of [7] proposed FL-SDN-based MDS in which SDN controllers first collect vehicle data. Then they train local models and send them for aggregation in the cloud. The authors of [8] proposed an FL-based MDS to detect vehicular inter-slice attacks consisting of two FL processes. FL models are first aggregated in intermediate FL nodes. Then, they are sent to the central FL server for building the global model. However, [6–8] are vulnerable to single-point-of-failure and attack risks since there is only one centralized server to calculate the global model. The authors of [9] proposed FL-based privacy preservation collaborative attack detection that leverages a set of FL servers to calculate global models. However, this solution is vulnerable to Byzantine and model falsification attacks. The authors of [10] combined blockchain and FL for collaborative attack detection. The blockchain system consists of Road Side Units (RSUs), which store global models obtained after running the Proof-of-Accuracy (PoA) algorithms. However, this solution designs unrealistic and heavy blockchain infrastructure since RSUs have limited storage and processing capacities that can prevent them from running blockchain operations. Moreover, selecting FL workers (CAVs) is very challenging, given the limited coverage of RSUs, and the high mobility of CAVs.

Unlike the above-mentioned FL-based solutions, our framework has a decentralized FL architecture avoiding thus single-point-of-failure and attack risks. It also has verification mechanisms for overcoming Byzantine and model falsification attacks. Furthermore, the framework offers a lightweight edge-enabled blockchain system enabling smart contracts and managing incentives for trusted and successful collaboration between FL servers and workers.

III. On-Demand Security Framework for 5GB Vehicular Networks: Building Blocks

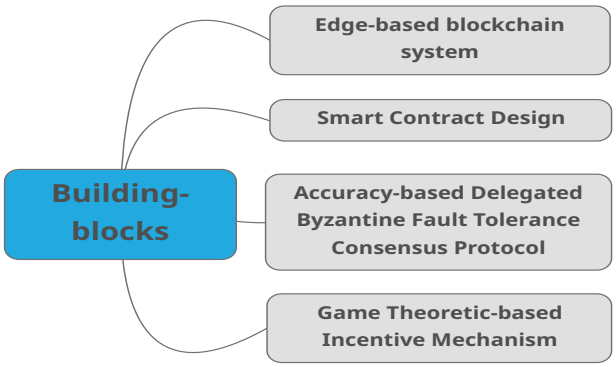

Fig. 1: The framework's building blocks

As shown in Figure 1, the cornerstones of our framework are: (i) an edge-based blockchain system that ensures secure collaboration in building global models while avoiding Byzantine attacks; (ii) efficient smart contracts between stakeholders and FL workers that ensure trusted and fair rewards for FL workers; (iii) an incentive mechanism to stimulate and select the best FL workers that provide highly accurate local models; and (iv) a lightweight consensus protocol for providing fast and reliable mining of blocks. In the following, we describe these building blocks in detail.

A. Edge-based blockchain system

Our framework consists of two layers, as shown in Figure 2. The first layer is the infrastructure that includes CAVs and gNodeBs equipped with 5G-V2X technologies. The nodes of this layer can be honest or malicious and be selected to serve as FL workers. Indeed, the threat model includes malicious nodes that perform internal attacks such as position falsification and message dropping. The same nodes can also perform Byzantine and model falsification attacks if selected to serve as FL workers. Moreover, the second layer is the 5GB edge, mainly consisting of distributed FL-enabled edge nodes with sufficient data storage, processing, and computing capabilities connecting through wired and secure fiber connections for sharing local ML models. Each FL-enabled edge

node has a consortium blockchain hosting transactions and smart contracts to enable reliable sharing of local models and secure aggregating global models. Moreover, each FL-enabled edge node manages a limited geographic zone (e.g., a smart city), has global knowledge of the mobility of CAVs, and communicates with gNodeBs and CAVs within this zone through secure 5G communication links. If we consider implementing this framework in Luxembourg as a study case. An FL-enabled edge node can be dedicated to each country's twelve provinces. An FL-enabled edge can communicate with vehicles via 5G base stations in the province. Specifically, FL-enabled edge nodes can be deployed on the cloud RAN in a Baseband Unit (BBU).

B. Smart Contract Design

The framework receives creation requests for smart contracts from stakeholders. A stakeholder is an entity interested in building ML models for detecting specific types of attacks. Stakeholders can comprise, for example, government entities, security organizations, or road organizations. The smart contract has several parameters, as shown in Figure 2. The model parameters are publicly shared among the FL workers, while other parameters are only shared with the FL-enabled edge nodes. Model parameters include the model structure, regularization parameters, weights, loss function, and learning rate. Other parameters comprise, for example, (i) the dataset used for validating local models and (ii) the maximum number of rounds to run for building a global model. The smart contract also defines five functions, which are: (i) *Initialize()*: it is called in step 2 and used to initialize the smart contract's parameters such as the initial global model, the validation dataset, and the reward, (ii) *Select()*: it is called in step 3 and used to select FL workers in each FL round, (iii) *Invoke()*: it is invoked at the end of each FL round to aggregate the global model in step 6 for the next round; (iv) *Reward()*: it is an internal function called in step 7 and used to calculate rewards for FL workers, and (v) *Close()*: it is used to close the contract after completing the number of FL rounds and returns the resulting global model to the stakeholder.

C. Accuracy-based Delegated Byzantine Fault Tolerance Consensus Protocol

To ensure a coherent and recognized ledger for all blockchain members, we propose a lightweight and scalable consensus protocol called the Accuracy-based Delegated Byzantine Fault Tolerance (A-DBFT) consensus protocol based on [13]. Our consensus protocol comprises two steps:

(i) **Selecting the leader and consensus members**: the blockchain members (edge nodes) can be simple; they can thus only verify, broadcast local models into the blockchain network, and accept the validated blocks, or they can be miners acting as simple and participating in consensus processes. After the end of every consensus process, all edge nodes recalculate the average accuracy of the aggregated final global model. The framework selects the top M edge nodes with the highest average accuracy values as consensus members. M should be greater than $3f + 1$, where f is the maximum number of faulty nodes to tolerate.

(ii) **The consensus process**: it comprises the following steps: (a) **Broadcast**: after the end of every FL round, all the received local models are broadcast in the entire blockchain for audit and verification; (b) **Collect**: miners collect all the local models. Then, miners check local models and add validated ones to a list. Each miner waits to receive all the local models before executing the smart contract locally; (c) **Propose**: after all non-leaders members have finished building their local blocks, the leader miner broadcasts a proposal to all non-leaders; (d) **Confirm**: once a non-leader receives a candidate block, it first verifies its validity, then retrieves the state of the block for comparing it with its local state. If the check passes, each non-leader broadcasts a confirmation message. However, if the received block is invalid, the non-leader triggers a view change, which calculates the next view change. Therefore, the non-leader will broadcast a change view message; (e) **Publish**: each miner keeps counting the number of received confirmations and view changes. A miner reaches a consensus only if the number of received messages from other distinct consensus members exceeds $(M - f)$. Finally, the next leader is selected to prepare for the next consensus process.

D. Game Theoretic-based Incentive Mechanism

We propose formulating their selection process as a game theory model to stimulate FL workers to provide accurate local ML models. Various game-theoretical models can be used in this problem. For example, the problem can be formulated as a Stackelberg game. This game consists of an edge node acting as the leader and several FL workers operating as followers. Each FL worker is rational in deciding on its contribution level regarding the accuracy of its local model for serving stakeholders. Moreover, the FL worker's utility may also depend on other factors of reward and costs. On the one hand, the utility of edge nodes depends on their paying monetary costs and the trained model performance. On the other hand, FL workers should cover the costs spend on collecting data and training models. Stackelberg's problem could be solved in this conflict of interest situation to find the equilibrium.

IV. On-Demand Security Framework for 5GB Vehicular Networks: Workflow

Figure 2 shows the workflow of the framework, which consists of seven steps. **Step 1** is the system initialization; CAVs, gNodeBs, and FL-enabled edge nodes must register with the Certification Authority (CA). During the registration, each node obtains a legitimate identity consisting of a private key, a public key, and a public certificate. Then, the process starts in **step 2** when a stakeholder submits its smart contract into the blockchain system. Once the blockchain system receives the smart contract, a consensus process runs to store it on the blockchain, get an address, and be invocable by edge nodes. Once the smart contract is ready, edge nodes compete to find the best FL workers to perform the task. Thus, in **step 3**, edge nodes broadcast offers to stimulate CAVs and gNodeBs to participate. Potential FL workers then submit their requests

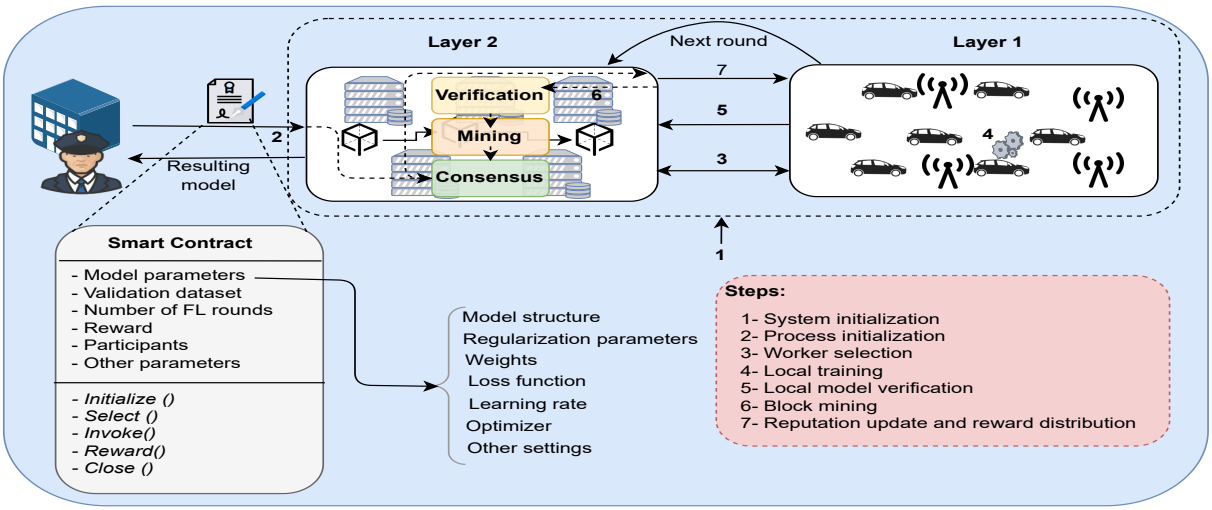

Fig. 2: The framework's Workflow

to edge nodes, broadcasting them over the blockchain system. Each edge node selects the best FL workers based on their reputations for the current FL round. The reputation of an FL worker is computed based on the accuracy of its previous local models and its trustworthiness. Once FL workers are selected, each edge node sends a confirmation to the selected FL workers within its controlled area.

At the beginning of each FL round, edge nodes send the latest global model to FL workers. Thus, each FL worker calculates the global model's weights update locally using its local dataset in **step 4**. At the end of the round, FL workers send their local models to edge nodes. Edge nodes broadcast received local models throughout the blockchain network for verification and consensus. Then, in **step 5**, FL-enabled edge nodes collaboratively verify local models to detect Byzantine attacks. Edge nodes evaluate models using the validation dataset. Then, they eliminate a suspicious model if the accuracy difference between the local and the global models exceeds a certain threshold. The validation threshold can be statically or dynamically set after each training round [12].

In step 6, once all local models are collected and checked by the mining nodes, the current leader aggregates all its validated models and calculates the current global model. It also calculates the distribution of rewards for FL workers and generates transfer transactions for rewards and reputation updates for FL workers. The block thus includes the global model, validated local models, rewarding transactions of FL workers, and the newly calculated reputation values. Finally, the consensus protocol runs to insert the block into the blockchain. After each FL round ends, the framework updates FL workers' reputation values in **step 7**. The framework decreases the reputation of FL workers with suspicious local models. In contrast, it increases the reputation of those with valid local models according to the accuracy of their models. The framework then uses the reputation values (R_W) to calculate the reward allocated for each FL worker. It first divides the stakeholder's reward in line with the number of FL rounds. Then, given C_{R_i} is the number of coins allocated for an FL round (R_i), the number of coins allocated to each FL worker (k) equals $C_{R_i} * \frac{R_{W_k}}{\sum_{j=1}^{l} R_{W_j}}$.

While steps 1 and 2 only run at the system and the process initialization, the remaining steps (3-7) repeat until the target number of FL rounds are reached. At the end of the process, the framework sends the resulting global FL model back to the stakeholder.

V. Comparative Analysis and Performance Evaluation

This section analyzes the framework's key features compared to the state-of-art works. We also evaluate the performance of our framework in terms of the accuracy of built models, blockchain consensus time, and incentive mechanism while demonstrating its out-performance of relevant FL-based solutions.

A. Comparative Analysis

Table I compares our framework with related ML-based collaborative MDS considering different criteria. Specifically, (i) columns ("Privacy preservation" and "Secure") indicate security and privacy preservation properties, (ii) columns ("Distributed", "Incentive FL workers", and "Overhead") examine the ML architecture and the overhead generated, and (iii) columns ("Consensus Protocol" and "Scalability") examine blockchain proprieties if applicable. Note that the "+" sign means that the solution provides the feature. More specifically, the "Distributed" column indicates that the solution uses several learning nodes to build the global model instead of a centralized one. The "Overhead" column indicates whether the overhead generated by the solution is small or large. The value mentioned in Table I is mainly based on the learning mode used by the solution (distributed Data-center/peer-peer/Federated). According to [11], distributed Data-center (as in [3]), centralized (as in [4]), and peer-to-peer (as in [5]) generate large overhead while Federated (as in [6, 9, 10] and our solution) generate a small overhead. The "Secure" column

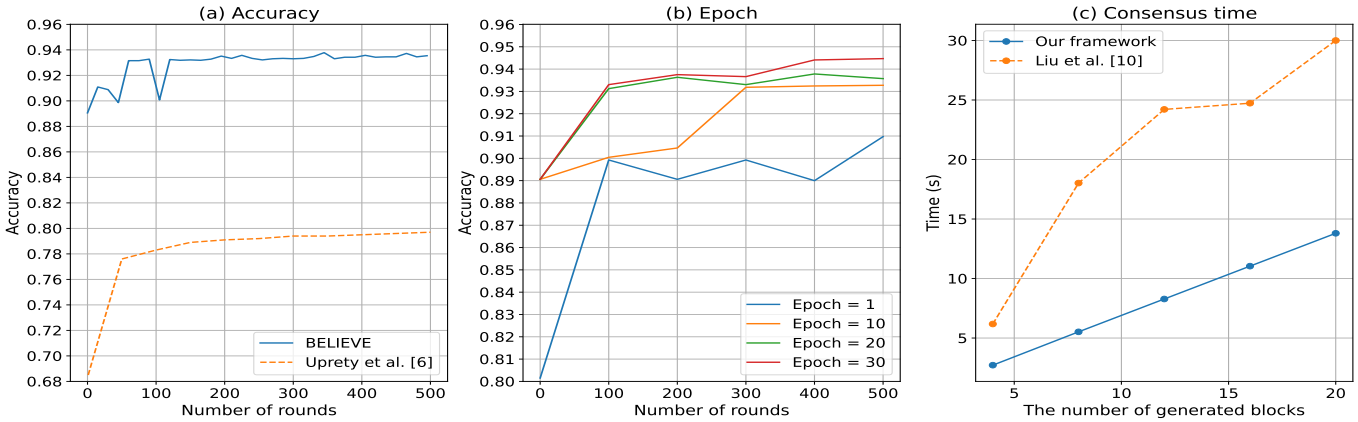

Fig. 3: Performance of our framework

shows that the solution proposed in [10] and our solution are the only approaches to secure learning. For the "Consensus protocol" column, we only mention the name of the protocol if applicable. The "Scalability" column depends on the used consensus protocol. This column is based on the comparison given in [14] to determine whether the work is scalable.

Table I shows that our framework is based on a secure distributed architecture to build attack detection models while preserving privacy. In addition, thanks to smart contracts, the framework ensures trusted and incentivized interactions with FL workers. Moreover, our framework uses a lightweight consensus algorithm (A-DBFT) that ensures secure, fast, and reliable mining of blocks and high scalability. Furthermore, in our framework, mobility and scalability are handled by a region-based management mechanism. Specifically, each FL-enabled edge node managed a limited geographic region. Moreover, our scheme does not manage many CAVs in each region since the number of FL workers is also limited.

TABLE I: Comparison of state-of-the-art works

Solution	Distributed	Privacy Preservation	Overhead	Secure	Incentive FL workers	Consensus Protocol	Scalablity
[3]	+	-	Large	-	-	-	-
[5]	+	+	Large	-	-	-	-
[4]	-	-	Large	-	-	-	-
[6]	-	+	Small	-	-	-	-
[7]	-	+	Small	-	-	-	-
[8]	-	+	Small	-	+	-	-
[9]	+	+	Small	-	-	-	-
[10]	+	+	Small	+	-	PoA	-
This work	+	+	Small	+	+	A-DBFT	+

B. Performance evaluation

We have conducted three experiments to demonstrate the effectiveness of our framework.

1) Experiment 01:

In the first experiment, We used Tensorflow and Keras Python libraries to design an abstract FL architecture. Specifically, deep learning models with two ten-unit hidden layers have been trained on a 12-CPU machine with 15 GB of RAM. The weights of local models have been computed using the Stochastic Gradient Descent (SGD). In addition, the loss function has used the categorical cross-entropy with a 0.005 learning rate. Moreover, the weights of the global model have been calculated after each round based on the Federated averaging algorithm. Our evaluation uses VeReMi, a dataset generated from network simulations that implements position falsification attack scenarios in 5GB vehicular networks. Our evaluation adopts the scenarios of the medium traffic density and 20% attacker ratio. The feature extraction leverages our previous work [15].

Figure 3.a compares the accuracy obtained by our framework with Uprety et al [6]. We compare our framework only with this work because the proposed solution has also been trained on the VeReMi dataset. The number of running epochs by FL workers equals 20. Our results demonstrate that accuracy values obtained by our framework outperform those obtained in [6]. Indeed, after running the same FL rounds considered in [6], our framework achieves more than 94% of accuracy, while [6] only achieves approximately 80%. Figure 3.b focuses on our framework; we evaluated the impact of the number of running epochs on the obtained accuracy values (the number of FL workers is set to 3). The results show that accuracy values increase with the number of running epochs.

2) Experiment 02:

In the second experiment, we evaluate the consensus time. We run the A-DBFT consensus protocol developed using Python programming language in a machine equipped with a CPU (Intel i5 2.6 GHz) and 8 GB of RAM. Table II shows the number of transactions per block and consensus time versus the number of FL workers. Note that the number of consensus members considered equals seven. As we can see, the number of transactions increases as the number of FL workers increases. Consequently, the consensus time increases with the number of involved FL workers.

Figure 3.c compares the consensus time in our framework with Liu et al. [10] for various numbers of generated blocks. We compared our framework only with this work because it is the only solution among the state-of-the-art works presented

in Table I that proposes to secure learning. The number of FL workers considered in our evaluation equals ten. As we can see, the consensus time in our framework increases with the number of generated blocks, which is the same tendency for the values of consensus time observed in [10]. The values of the consensus time in our framework are lower than the consensus time in [10] for all cases, thanks to A-DBFT.

TABLE II: Transactions per block and consensus time vs. FL workers

	The number of FL workers		
	3	5	10
Number of transactions per block	10	16	31
Consensus time (ms)	122	204	690

3) Experiment 03:

In this experiment, we analyzed the game's theoretical-based incentive building block. This evaluation is based on our previously proposed model, taking costs as energy and processing overhead [8]. Specifically, we analyzed the best responses of the FL-enabled edge node, considering the reward, energy, and processing overhead. For this experiment, we set the processing overhead per unit size to 0.2, and the amount of energy spent to 0.15. In addition, we vary the reward values within a $[0.1 - 10]$ range. Figure 4 represents the utility of

Fig. 4: The utility of the FL-enabled edge node with the variation of the reward and the accuracy of the local model

the edge node with the variation of the accuracy and reward. Various reward values change the utility of the edge node. In addition, the utility raises as the accuracy does. Moreover, The red dashed lines depict the edge node's optimal reward for having the most utility given the model accuracy. Therefore, the edge node must raise its reward to incentivize CAVs to improve local model accuracy. For instance, the edge node should increase the optimal reward from 3.9 to 4.9, i.e., 25%, to increase the accuracy of the local model from 0.85 to 0.95. Note that the accuracy values are considered a reasonable and realistic range in [0.85, 0.95].

VI. Challenges and Potential Solutions

This section highlights the significant challenges that can appear for the successful deployment of our framework and their potential solutions.

A. FL workers selection

In our framework, the selection of FL workers is only based on their reputations which can increase the prejudice of the framework toward a specific set of FL workers. This has several disadvantages, such as increasing the bias in global FL models and leading to the unfairness of the framework by monopolizing some FL workers at the expense of others. One solution to avoid this issue could be randomly selected part of FL workers. Specifically, an explorer parameter could be added to determine the ratio of FL workers that should be selected randomly.

B. Privacy preservation of Global FL model

Our framework ensures privacy preservation in learning since datasets are not shared between learning nodes. However, FL workers can keep global FL models after returning their updates to edge nodes. This can increase the privacy risks since FL workers can sell it to other interested stakeholders or attackers who can adapt their attacks to avoid being detected by the same model. One solution could be to leverage blockchain for traceability and identification of FL workers violating FL model privacy preservation.

C. Dynamic selection of parameters

As shown in the evaluation part, the accuracy of the global model depends not only on model parameters but also on the number of FL rounds and the number of epochs running by FL workers. Stakeholders should set the latter in the initialization phase. However, when submitting their requests, stakeholders cannot decide on the best choice of these parameters to achieve higher accuracy in fewer FL rounds. One solution that could help with this is to equip our framework with a recommender system that gives some recommendations to stakeholders to set values for these parameters to obtain higher accuracy. This system can be built based on the past experiences of our framework and could continuously be enhanced.

VII. Conclusion

This article proposed a novel security framework for 5GB vehicular networks. Our framework enables on-demand privacy preservation collaborative attack detection models for CAVs by leveraging federated learning, blockchain, and smart contracts. Moreover, Our framework provides mechanisms for stimulating FL workers and preventing Byzantine attacks. The performance evaluations showed that our framework archives more than 94% accuracy in building models on a well-known vehicular dataset and less than 1s of blockchain consensus time, outperforming related solutions. We have also discussed the challenges involved in our framework and their potential solutions. In future work, we plan to consider different approaches for selecting FL workers based mainly on their reputations and available computing resources to build deep neural networks.

ACKNOWLEDGMENT

This work was supported by the 5G-INSIGHT bilateral project (ID: 14891397) / (ANR-20-CE25-0015-16), funded by the Luxembourg National Research Fund (FNR), and by the French National Research Agency (ANR).

REFERENCES

[1] C. Lai, R. Lu, D. Zheng, and X. Shen, "Security and Privacy Challenges in 5G-enabled Vehicular Networks," *IEEE Network*, vol. 34, no. 2, pp. 37–45, 2020.

[2] A. Boualouache and T. Engel, "A Survey on Machine Learning-based Misbehavior Detection Systems for 5G and Beyond Vehicular Networks," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2023.

[3] N. Negi, O. Jelassi, H. Chaouchi, and S. Clemençon, "Distributed Online Data Anomaly Detection for Connected Vehicles," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2020, pp. 494–500.

[4] J. Shu, L. Zhou, W. Zhang, X. Du, and M. Guizani, "Collaborative Intrusion Detection for VANETs: a Deep Learning-based Distributed SDN Approach," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[5] T. Zhang and Q. Zhu, "Distributed Privacy-preserving Collaborative Intrusion Detection Systems for VANETs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 148–161, 2018.

[6] A. Uprety, D. B. Rawat, and J. Li, "Privacy Preserving Misbehavior Detection in IoV using Federated Machine Learning," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2021, pp. 1–6.

[7] A. Hbaieb, S. Ayed, and L. Chaari, "Federated learning based IDS approach for the IoV," in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, 2022, pp. 1–6.

[8] A. Boualouache and T. Engel, "Federated Learning-based Inter-slice Attack Detection for 5G-V2X Sliced Networks," in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. IEEE, 2022, pp. 1–6.

[9] ——, "Federated learning-based scheme for detecting passive mobile attackers in 5G vehicular edge computing," *Annals of Telecommunications*, vol. 77, no. 3, pp. 201–220, 2022.

[10] H. Liu, S. Zhang, P. Zhang, X. Zhou, X. Shao, G. Pu, and Y. Zhang, "Blockchain and Federated Learning for Collaborative Intrusion Detection in Vehicular Edge Computing," *IEEE Transactions on Vehicular Technology*, 2021.

[11] Z. Du, C. Wu, T. Yoshinaga, K.-L. A. Yau, Y. Ji, and J. Li, "Federated Learning for Vehicular Internet of Things: Recent Advances and Open Issues," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 45–61, 2020.

[12] Z. Li, H. Yu, T. Zhou, L. Luo, M. Fan, Z. Xu, and G. Sun, "Byzantine Resistant Secure Blockchained Federated Learning at the Edge," *IEEE Network*, 2021.

[13] M. Castro, B. Liskov *et al.*, "Practical Byzantine fault tolerance," in *OSDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[14] S. Wang, X. Huang, R. Yu, Y. Zhang, and E. Hossain, "Permissioned Blockchain for Efficient and Secure Resource Sharing in Vehicular Edge Computing," *arXiv preprint arXiv:1906.06319*, 2019.

[15] F. Hawlader, A. Boualouache, S. Faye, and T. Engel, "Intelligent Misbehavior Detection System for Detecting False Position Attacks in Vehicular Networks," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–6.

VIII. BIOGRAPHY

Abdelwahab Boualouache is a research associate at the University of Luxembourg. His current research interests include security and privacy in 5GB and connected and automated vehicles.

Bouziane Brik is an associate professor at the University of Burgundy, France. His research interests include the Internet of Things (IoT) and vehicular networks.

Sidi Mohammed Senouci is a full professor at the University of Burgundy, France. He is also the Director of the laboratory DRIVE.

Thomas Engel is a Professor of Computer Networks at the University of Luxembourg. His SECAN-Lab team deals with performance, security, privacy, and identity handling in Next Generation Networks.