2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

A Federated Transfer Learning-Empowered Blockchain-Enabled Secure Knowledge Sharing Scheme for Unmanned Any Vehicles in Smart Cities

Anik Islam and Hadis Karimipour
Schulich School of Engineering, Department of Electrical & Software Engineering,
University of Calgary, Calgary, Canada
Email: {anik.islam, hadis.karimipour}@ucalgary.ca

***Abstract*—Smart cities embrace unmanned autonomous vehicles (UxVs) for urban mobility and addressing challenges. UxVs include UAVs, UGVs, USVs, and UUVs, empowered by AI, particularly deep learning (DL), for autonomous missions. However, traditional DL has limitations in adapting to dynamic environments and raises data privacy concerns. Limited data availability and starting from scratch to adapt to a new environment during missions pose challenges. Additionally, cyber threats, particularly in terms of communication and data security, can jeopardize the missions performed by UxVs. This paper proposes a federated transfer learning scheme for UxVs, sharing prior knowledge and training with limited data while ensuring security through blockchain. Domain adaptation with maximum mean discrepancy enhances the DL model's performance in target domains. The proposed scheme's feasibility is demonstrated in an empirical environment, and it outperforms existing works.**

***Index Terms*—blockchain, domain adaptation, federated transfer learning, mist computing, unmanned any vehicles**

I. Introduction

Smart cities are rapidly evolving, driven by the integration of cutting-edge technologies that aim to enhance urban living and address various challenges associated with growing populations [1]. Among these technologies, unmanned any vehicles (UxVs) play a crucial role in modernizing urban mobility, offering innovative solutions for transportation, surveillance, delivery services, and more. UxV is a paradigm in which unmanned vehicles (i.e., unmanned aerial vehicle (UAV), unmanned ground vehicle (UGV), unmanned surface vehicle (USV), and unmanned underwater vehicle (UUV)) are employed to perform missions (e.g., surveillance) cooperatively [2].

Artificial Intelligence (AI) has emerged as a leading force in technological progress, revolutionizing various industries and shaping the future of human society [3]. Deep learning (DL), a groundbreaking branch of AI, is rooted in the concept of building intricate, hierarchical representations from vast data sets [4]. This empowers machines to learn and make decisions autonomously, much like the neural networks of the human brain. DL enables automation in UxVs, facilitating their missions. During these missions, UxVs encounter dynamic environments that were not considered during the creation of the DL model. In traditional DL, adjustments require retraining to adapt to new changes. Additionally, developing an efficient DL model demands extensive data, which can be challenging, especially when UxVs contain privacy-sensitive information, leading to data refusal. Moreover, training from scratch during a mission can reduce the lifespan of UxVs with limited resources. Besides, both data and communications are exposed to cyber threats.

Federated Learning (FL) is an innovative technology that enables models to be trained on numerous decentralized devices or servers, ensuring that data remains localized [5]. By allowing on-device training without sharing raw data, FL fosters collaborative model training among organizations, preserving privacy, and tapping into collective insights from distributed datasets [6]. In FL, only model parameters are shared, not raw data [7]. This approach can assist UxVs in constructing efficient models without privacy concerns. However, UxVs operating in this new environment may encounter limited data, even though they can access other UxVs' data through FL. Additionally, training from scratch can jeopardize the mission.

Transfer learning (TL) is a potent AI technique that harnesses knowledge acquired from one task to enhance performance in another related task [8]. Its rising popularity is due to its ability to expedite model training, improve generalization, and optimize the use of limited data. TL can be valuable for UxVs, helping them overcome the challenges posed by limited datasets and starting from scratch. By leveraging knowledge from other UxVs and fine-tuning with limited data, UxVs can swiftly adapt to their environment. However, the system faces security challenges, both in terms of data and communication, which could potentially compromise the success of the mission.

Blockchain, a disruptive innovation in cutting-edge technology, is a distributed ledger where data is shared and validated by participants known as miners, with identical copies held by all [9]. Changes to the data require validation from each participant, ensuring uniform updates and immutability for data integrity and security [10]. In the blockchain network, each participant possesses a private key for validating incoming messages and a corresponding public key for initiating secure communication, enhancing security and authenticity. Leveraging blockchain, UxVs can effectively address cyber threats in data and communication.

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

Its decentralized and secure nature offers a promising solution for mitigating risks and ensuring the integrity of UxV missions.

Among the studies focused on autonomous unmanned vehicles, Masud et al. [11] introduced an innovative automated and secure garbage management system. They employed a DL model integrated with UxV to reduce human effort in traditional garbage management, utilizing various types of UxVs for collecting and disposing of garbage from land and sea surfaces. However, privacy issues during training and the challenges of dynamic environments and limited datasets were not addressed in their work. In contrast, Jamshid et al. [12] proposed a novel and efficient FL scheme, the hierarchical FL algorithm, designed for edge-aided UAV networks. Their approach utilized edge servers in base stations as intermediate aggregators, leveraging shared data. While they considered privacy challenges, the issues of dynamic environments and limited datasets were not taken into account, and no measures were implemented against cyber threats. Furthermore, Gianluca et al. [13] presented a reinforcement learning-based solution for UAV connectivity-aware path planning in diverse simulation environments. They also proposed a transfer learning technique to improve agent learning in a new mmWave domain using knowledge from a source domain. Nevertheless, their work lacked attention to privacy issues related to entities (UAVs) and did not incorporate security countermeasures against cyber attacks. A platform is needed that can aid UxVs in conducting missions securely and autonomously, while also taking into account dynamic environments and limited datasets without compromising privacy.

Considering the aforementioned challenges (privacy, dynamic environment, limited dataset, and security), this paper presents a collaborative scheme called FTL-UxV for UxVs. The proposed approach involves sharing prior environmental knowledge and training limited available data in a distributed manner. Furthermore, security modules are integrated on top of blockchain technology. To the best of our knowledge, the adaptation of FTL to dynamic environments while ensuring security through blockchain has not been studied before. The main contributions of this paper are as follows. (1) An FTL-based knowledge-sharing scheme is introduced, which incorporates a domain adaptation technique to effectively map the provided knowledge to the target task. Additionally, blockchain is integrated to facilitate seamless communication and data management. (2) A discussion is presented on preparing the dataset and DL models for training by applying FTL. (3) Experiments are conducted on UxVs using FTL and evaluated based on precision, recall, F1-score, and accuracy metrics. A blockchain environment is established on top of Ethereum, and performance is measured by observing the update time in the blockchain during the transmission of training data.

The paper's outline: Section II introduces preliminaries. Section III presents proposed schemes and system architecture. Section IV details scheme functionality. Section V covers experimental setup and performance analysis. Conclusion in Section VI summarizes findings, contributions, and future works.

II. Preliminaries

A. Federated Learning

FL, introduced by Google in 2016, offers a decentralized alternative to traditional machine learning. It enables multiple entities (termed as participants) to collaboratively train a global model while keeping their data local. During the training process, each participant shares model parameters (i.e., weights and bias) instead of the raw data. A central entity (termed as an accumulator) coordinates model updates by accumulating these parameters collected from individual participants. In FL, each participant P_i holds a local dataset DS_i, where $DS_i = \langle (x_1^i, y_1^i), (x_2^i, y_2^i), ..., (x_L^i, y_L^i) \rangle$, with x being the feature space containing d features, y as the label, and each P_i holding L^i samples. In FL, models are trained and share model parameters ϕ_i with the accumulator α. The objective is to minimize the loss as follows [14].

$$\phi_i = \arg\min_{\phi_i \in \Re^d} F_i(\phi_i), \tag{1a}$$

$$where\ F_i(\phi_i) \stackrel{\text{def}}{=} \frac{1}{L_i} \sum_{\forall j \in L_i} f(\phi_i\ (x_j^i, y_j^i)). \tag{1b}$$

Where $f(.)$ represents the local loss function. When a participant P_i receives the model from the accumulator α, it immediately begins training and continues to optimize the local loss function, $\phi_i^{e+1} = \phi_i^e - \eta \nabla F_i(\phi_i^e)$. When all P_i return ϕ_i, the accumulator α initiates the accumulation process by employing FedAvg [15] for a total of N participants as follows.

$$\phi = \frac{1}{N} \sum_{\forall i \in N} \phi_i \tag{2}$$

B. Transfer Learning

TL addresses the challenge of data scarcity by allowing models to leverage knowledge gained from previous tasks or domains, thus reducing the data requirements for new tasks significantly. It is based on the intuition that knowledge acquired from solving one problem can be valuable when dealing with a related but different problem. This concept draws inspiration from human learning, where individuals often apply knowledge learned in one context to solve new problems in different situations. The main idea behind transfer learning is to extract and transfer the valuable representations (i.e., features) learned during the training of a source domain to enhance the learning process in a target domain. In TL, the main objective is to leverage the source domain $\mathcal{DS}^s$, $\mathcal{DS}^s = \{\cup_{l=1}^{M'} (x_l^s, y_l^s), P^s(.)\}$ in the target domain $\mathcal{DS}^t$, $\mathcal{DS}^t = \{\cup_{k=1}^{M} (x_k^t, y_k^t), P^t(.)\}$ to enhance the performance of the target task $T = (Y, f(.))$, where $x_i^s \in X^s$, $y_i^s \in Y^s$, $x_j^t \in X^t$, $y_j^t \in Y^t$, $X^s \neq X^t$ and $M << M'$. In this context, X represents the feature space, Y denotes the label space, $P(.)$ refers to the data distribution, $f(.)$ represents the prediction

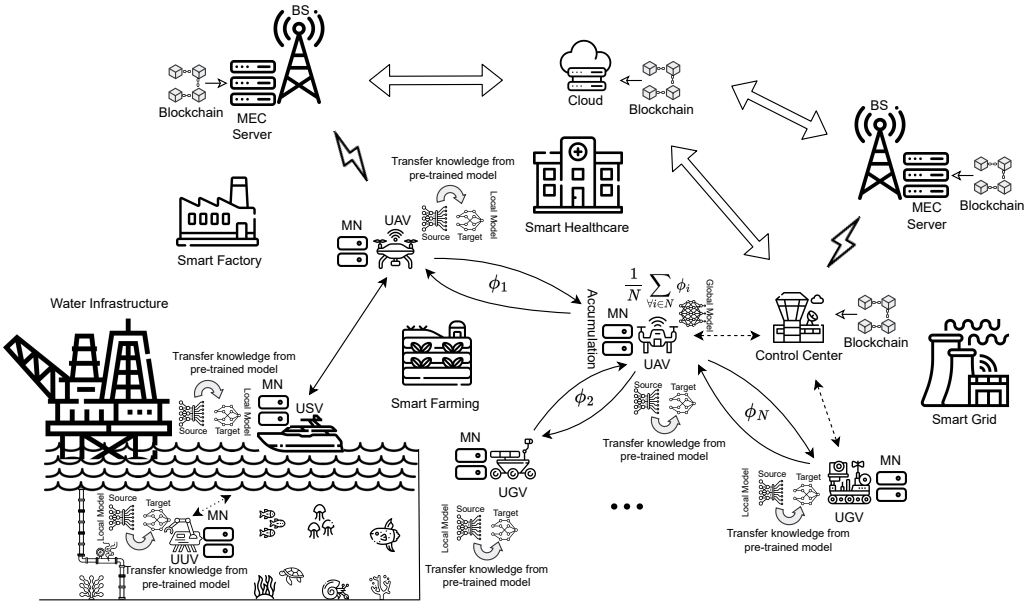

Fig. 1: FTL-enabled knowledge sharing process for UxVs on top of the blockchain in a smart city.

function, M' denotes the total number of samples in $\mathcal{DS}^s$, and M represents the total number of samples in $\mathcal{DS}^t$. The primary objective of TL is to improve the prediction function $\mathfrak{f}(.)$ in $\mathcal{DS}^t$ by leveraging the knowledge acquired from $\mathcal{DS}^s$.

III. System Overview

This research introduces FTL-UxV, a secure knowledge-sharing scheme for UxVs in smart cities, which leverages FTL and blockchain technology. The FTL-UxV system involves multiple entities and consists of several key components. UxVs equipped with IoT sensors are deployed in smart cities to carry out diverse tasks. They can enhance their collective performance by sharing knowledge and insights with one another during task execution. In FTL-UxV, a hierarchy of servers is established to accommodate entities with varying levels of computing power. A mist node[1] (MN) is embedded within UxVs to support low computing entities, assisting them in performing onboard computations and managing connectivity with the nearest server. At the edge of the network, a MEC server[2] is utilized to provide real-time support to UxVs. Additionally, a cloud server is incorporated to serve as one of the data storage platforms. All MEC servers and the cloud act as miners in the network. FTL-UxV incorporates blockchain technology to ensure secure data management and enable a secure computing environment.

IV. Proposed Approach for Secure Knowledge Sharing

In FTL-UxV, all entities are required to register, and their data is securely stored in the blockchain. Each entity possesses a private key Pr and a corresponding public key Pb, which serves as its identity. During the registration process, entities share their Pb along with basic information, such as device details, with the nearest server. Upon receiving this information, the server stores it in the blockchain. When a UxV U is deployed for a task, it aims to gather knowledge from other UxVs already operating in the field to quickly adapt to the environment. The UxV that collects and accumulates this knowledge is referred to as the accumulation UxV (U_c) in the system. Before deploying U_c, a model m is generated based on publicly available datasets or data collected previously. To enhance the performance of m, U_c initiates a FL-based training process. Firstly, U_c sends invitations to nearby entities (i.e., other UxVs) containing information about the target domain and the sender's public key. Entities with relevant data respond using public keys as participants U_p.

After receiving acceptance from all participants, U_c compiles a list of participants. This list denoted as $\overline{P}$, $\overline{P} = \bigcup(U_p^i) \mid i \in N$, where N represents the total number of participants. Next, U_c chooses an active participant from $\overline{P}$, $SP = max(r \times N, 1)$, where r represents dropout ratio. Subsequently, the final list of participants is then formed by including the contributors corresponding to the indices present in SP, $\overline{SP} = \cup_{i \in SP}(U_p^i)$. These selected participants from $\overline{SP}$ are the ones allowed to participate in the training process.

Upon finalizing participants, U_c creates a secret key δ to enable a fast and secure training process, $\delta = K(\langle \overline{SP}, \tau, H(\tau) \rangle) \mid K : \{0,1\}^* \mapsto \{0,1\}^z, H : \{0,1\}^* \mapsto \{0,1\}^{z'}$. Once key is created, U_c shares it to all along with the global model m_G encrypted by U_p's public key, $e_{msg}^i = E_{Pb_i}(\langle \delta, m_G \rangle), \forall i \in \overline{SP}$. A nonlinear mapping is calculated based on the DS^s, $nm = \frac{1}{|DS^s|} \sum_{\forall x \in DS^s} \mu(x)$

[1]Mist Computing is a computing paradigm that processes data locally for reduced latency and real-time analysis [16].

[2]Multi-access Edge Computing (MEC) revolutionizes mobile networks, deploying computing resources near access points or gateway to meet low-latency, high-bandwidth demands of real-time applications [17].

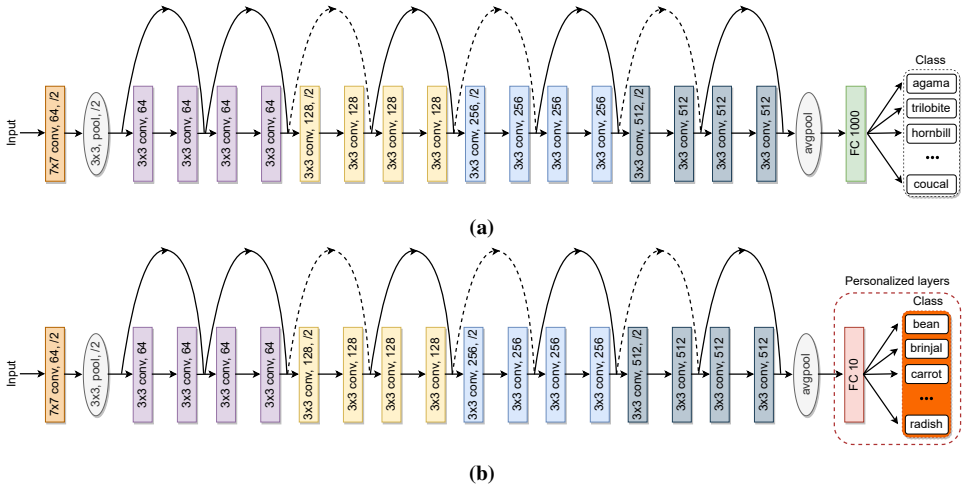

Fig. 2: Model preparation for FTL: (a) source model, (b) personalized target model.

to assist participants performing domain adaptation. Upon receiving the e^i_{msg}, each U^i_p decrypts it using their Pr_i, $d^i_{msg} = D_{Pr_i}(\langle \delta, nm, m_G \rangle)$.

In FTL-UxV, a classification task is considered in the training. After obtaining the global model m_G, $y^s \approx m_G(\phi \mid DS^s)$, each U^i_p freezes the first v layers of m_G. The objective is to maintain the parameters ϕ_i while optimizing the loss function F based on the target data DS^t. The goal is to fine-tune the model to adapt it to the specific task at hand. The process of optimizing the parameters for the target domain can be described as follows, $\phi_i = \underset{\phi_i}{\text{argmin}}\ F(d(x^i_j, \phi^i_j), y^i_j), \forall j \in |DS^t_i|$. Here, $d(.)$ is the objective function of the target domain. Training a classifier solely using the source data can often lead to overfitting to the source domain, resulting in reduced performance when applied to the target domain during testing. To address this issue and improve generalization, a domain adaptation technique is adopted to learn a representation that minimizes the dissimilarity between the source and target domains. The primary objective is to create a more robust model that can effectively generalize to unseen data from the target domain. FTL-UxV employs Maximum Mean Discrepancy (MMD) ϑ to minimize dissimilarity which can be calculated as follows [18].

$$\vartheta(nm, DS^t_i) = \| nm - \frac{1}{|DS^t_i|} \sum_{\forall x' \in DS^t_i} \mu(x') \| \tag{3}$$

The objective is to develop a representation that facilitates the training of robust classifiers capable of transferring knowledge across diverse domains, extending beyond the sole reduction of domain distance. To achieve this, Eq. 3 s incorporated into the loss function, $F_i = F(d(x^i_j, \phi^i_j), y^i_j) + \lambda \vartheta^2(nm, DS^t_i), \forall j \in |DS^t_i|$. Here, λ is utilized to control the degree of domain distances. After completing training, each U^i_p returns ϕ_i encrypted using δ. When U_c receives all updates, it performs accumulation following Eq. 2. At the end of each training episode, U_c creates a block containing the training updates, $b = \langle \{Pb_c, \phi_c\}, \cup_{i \in |\overline{SP}|} \{Pb_i, \phi_i\}, ... \rangle$. FTL-UxV adopts the proof of authority (PoA) as a consensus algorithm. In PoA, each miner must wait for their turn to propose a block. Once a miner gets its turn, it broadcasts its block across the network. Upon acceptance from the majority of miners, the block is added to the blockchain. After this, U_c broadcasts b in the blockchain network once it gets its turn, and it gets appended upon acceptance from other miners. The training process continues until convergence is achieved.

V. Experimental Setup and Performance Evaluation

A. Dataset preparation

In the experiments, the source model was trained using the ImageNet dataset [19], which is widely recognized and used for vision-based classification tasks. This dataset comprises more than 15 million high-quality images organized into 22,000 categories. The dataset consists of approximately 1.2 million training samples, 150,000 testing samples, and 1000 distinct classes. The target domain utilized a vegetable image dataset (https://www.kaggle.com/datasets/misrakahmed/vegetable-image-dataset), which consisted of 21,000 vegetable images belonging to 15 different classes. However, for training, only around 10 to 30 images per class were considered, and for testing, approximately 15 to 16 images per class were used. Additionally, only those classes and their corresponding data were selected (i.e., 10 classes) that were not present in the source dataset. The data was non-independent and identically distributed among the participants.

B. Model preparation

ResNet18 is a variant of the ResNet architecture, comprising a total of 18 layers (17 convolutional layers and one fully connected layer), as shown in Fig. 2(a). The model starts with an input layer of size 224×224. The first convolutional layer uses a 7×7 kernel with a stride of 2, followed by a 3×3 max-pooling layer with a stride of 2. The first

TABLE I: Global model performance for FTL-based knowledge sharing.

Prior Research →	Masud et al. [11]			Jamshid et al. [12]			**Proposed FTL-UxV**		
Features ↓	Precision	Recall	F1-score	Precision	Recall	F1-score	**Precision**	**Recall**	**F1-score**
Bean	0.4286	0.1875	0.2609	0.5000	0.0625	0.1111	**1.0000**	**0.9375**	**0.9677**
Bitter_Gourd	0.3256	0.9333	0.4828	0.2821	0.7333	0.4074	**0.8824**	**1.0000**	**0.9375**
Bottle_Gourd	0.2545	0.1333	0.1750	0.6667	0.4000	0.5000	**1.0000**	**0.9333**	**0.9655**
Brinjal	0.2727	0.1033	0.1498	0.5714	0.2667	0.3636	**0.9375**	**1.0000**	**0.9677**
Capsicum	0.8000	0.5333	0.6400	0.6923	0.6000	0.6429	**1.0000**	**0.9333**	**0.9655**
Carrot	0.7500	0.8000	0.7742	1.0000	0.8667	0.9286	**1.0000**	**1.0000**	**1.0000**
Papaya	0.6875	0.7333	0.7097	0.7143	0.6667	0.6897	**1.0000**	**1.0000**	**1.0000**
Pumpkin	0.5000	0.7333	0.5946	0.4615	0.8000	0.5854	**1.0000**	**0.9333**	**0.9655**
Radish	0.5000	0.6000	0.5454	0.3125	0.3333	0.3226	**1.0000**	**1.0000**	**1.0000**
Tomato	0.3333	0.1333	0.1905	0.4167	0.3333	0.3704	**0.9375**	**1.0000**	**0.9677**
Accuracy			0.4834			0.5033			**0.9735**
Macro avg	0.4852	0.4858	0.4522	0.5617	0.5062	0.4922	**0.9757**	**0.9738**	**0.9737**
Weighted avg	0.4854	0.4856	0.4522	0.5613	0.5033	0.4896	**0.9759**	**0.9735**	**0.9737**

residual block includes two convolutional layers with 3×3 kernels and 64 kernels each. The output of this block is combined with the output from the initial convolutional layers through a 3×3 convolution with 128 kernels, forming the second residual block. This pattern continues with the third residual block, which incorporates the output of the second block through skip connections and uses convolutional layers with 3×3 kernels and 256 kernels each. Finally, the fourth residual block is formed by combining the output from the third block through skip connections and using convolutional layers with 3×3 kernels and 512 kernels each. The output of the final residual block undergoes average pooling before being fed into fully connected (FC) layers. The FC layer applies the softmax function to produce the final output. Pre-trained ResNet18 is capable of classifying 1000 objects, such as agama, trilobite, and others. In the target domain, a pre-trained ResNet18 model was utilized. As mentioned earlier, the dataset used in the target domain was the vegetable image dataset. The original FC layer of ResNet18, which consisted of 1000 classes, was replaced with a new FC layer containing 10 classes, as shown in Figure 2(b). This modified pre-trained ResNet18 was then prepared for training in the target domain, leveraging knowledge transferred from the source domain. During training, participants froze the top layer of the model, which acted as a feature extractor with knowledge from the source domain.

C. Data Preprocessing and Model Training

In the training set, the images were randomly cropped and resized to 224×224, and a horizontal flip was applied with a probability of 0.5. Subsequently, the images underwent normalization using a specific normalization technique. The testing set was preprocessed similarly to the training set. During training, a batch size of 3 was utilized, and the initial learning rate was set to 0.001. The learning rate was reduced by a factor of 0.11 using the StepLR technique. As for the optimizer, the stochastic gradient descent with momentum (SGDM) [20] algorithm was selected. The entire training process was implemented using PyTorch.

D. Environment setup

An experiment environment was established to demonstrate the proof of concept (PoC). The computing environment consisted of an Intel Xeon CPU with 2 vCPUs and 13GB of RAM, along with an NVIDIA Tesla K80 GPU with 12GB of VRAM. Ubuntu 22.04.2 LTS served as the operating system. The simulation involved 10 participants, representing a combination of different UxVs, with one UxV acting as an accumulator. Shared key encryption was implemented using the Advanced Encryption Standard (AES-128), and Public Key encryption was performed using elliptic-curve cryptography. To ensure secure data storage, a consortium blockchain consisting of 20 mining nodes was established. The blockchain network was built on top of Ethereum, and the PoA consensus algorithm was employed. Python was used as the middleware for the experiment.

E. Performance Analysis

To evaluate performance, a comparison with existing works, namely, Masud et al. [11] and Jamshid et al. [12], is presented in Table I. The table presents precision, recall, and F1-score for each class label, along with accuracy, macro-averaged, and weighted-averaged F1-scores. Masud et al. [11] achieved moderate results with varying precision and low recall, indicating poor detection in some classes. Masud et al. [11] obtained an accuracy of 48%, but they did not address privacy and limited dataset issues, resulting in poor performance with data scarcity. On the other hand, Jamshid et al. [12] demonstrated better recall but sacrificed precision, leading to higher false positives. Their accuracy was slightly better at 50.33%. Although they considered FL, the dataset obtained was insufficient for high performance. In contrast, the proposed FTL-UxV method showed superior performance across all metrics. It achieved perfect precision, recall, and F1-score for certain classes, indicating accurate detection. The overall accuracy of 97.35% highlighted the method's reliability in precisely classifying instances from the target dataset. The proposed FTL-UxV approach considered FTL, effectively addressing privacy and limited dataset challenges.

Additionally, it employed domain adaptation techniques to adapt the target domain efficiently, which contributed to its high performance.

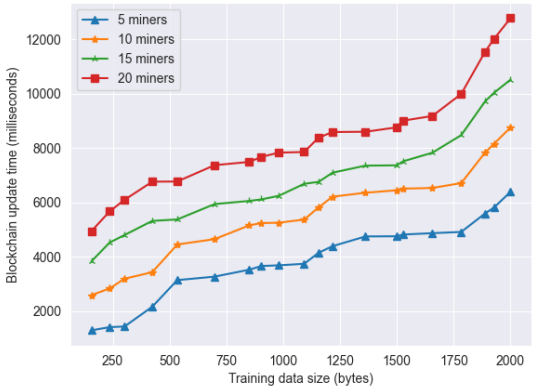

Fig. 3: Result of the blockchain update time over training data size.

Figure 3 illustrates the update time in the blockchain when appending training data along with the model to the network. The update time shows an increase with larger training sizes due to the necessity of transmitting more significant updates across the network, resulting in longer processing times. Additionally, with a greater number of miners participating, the time required to process a block and obtain acceptance from the miners also increases. This trend indicates that the update process becomes more time-consuming as the scale of the training data and the network's participation grows.

VI. Concluding Remarks & Future Works

A secure knowledge-sharing scheme was proposed for UxVs, utilizing FTL to train new models by borrowing knowledge from previously trained models. MMD-based domain adaptation techniques were employed during training to adapt the knowledge as much as possible. Subsequently, both training information and entities' information were securely stored on the blockchain, which also facilitated secure communication during training. A proof-of-concept environment was built, and several experiments were conducted, consistently demonstrating the superiority of the proposed scheme over existing research. For future work, the paper plans to extend the experiments to a real hardware environment, such as UAVs. Additionally, lightweight DL models, multi-dimensional incentive mechanisms, and the exploration of other cyber attacks, such as model poisoning, are on the agenda for future research.

References

[1] M. A. Hoque, M. Hossain, S. Noor, S. M. R. Islam, and R. Hasan, "Iotaas: Drone-based internet of things as a service framework for smart cities," *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 12 425–12 439, 2022.

[2] A. Islam, M. Masuduzzaman, A. Akter, and S. Young Shin, "Mr-block: A blockchain-assisted secure content sharing scheme for multi-user mixed-reality applications in internet of military things," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, pp. 407–411.

[3] P. McEnroe, S. Wang, and M. Liyanage, "A survey on the convergence of edge computing and ai for uavs: Opportunities and challenges," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15 435–15 459, 2022.

[4] S. Mittal, S. Srivastava, and J. P. Jayanth, "A survey of deep learning techniques for underwater image classification," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2022.

[5] A. Islam and S. Y. Shin, "A digital twin-based drone-assisted secure data aggregation scheme with federated learning in artificial intelligence of things," *IEEE Network*, vol. 37, no. 2, pp. 278–285, 2023.

[6] A. Islam and S. Y. Shin, "A blockchain-based privacy sensitive data acquisition scheme during pandemic through the facilitation of federated learning," in *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, 2022, pp. 83–87.

[7] A. Islam, M. K. Morol, and S. Y. Shin, "A federated learning-based blockchain-assisted anomaly detection scheme to prevent road accidents in internet of vehicles," in *Proceedings of the 2nd International Conference on Computing Advancements*, ser. ICCA '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 516–521. [Online]. Available: https://doi.org/10.1145/3542954.3543028

[8] T. V. Khoa, D. T. Hoang, N. L. Trung, C. T. Nguyen, T. T. T. Quynh, D. N. Nguyen, N. V. Ha, and E. Dutkiewicz, "Deep transfer learning: A novel collaborative learning model for cyberattack detection systems in iot networks," *IEEE Internet of Things Journal*, vol. 10, no. 10, pp. 8578–8589, 2023.

[9] J. W. Seo, A. Islam, M. Masuduzzaman, and S. Y. Shin, "Blockchain-based secure firmware update using an uav," *Electronics*, vol. 12, no. 10, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/10/2189

[10] A. Islam, T. Rahim, M. Masuduzzaman, and S. Y. Shin, "A blockchain-based artificial intelligence-empowered contagious pandemic situation supervision scheme using internet of drone things," *IEEE Wireless Communications*, vol. 28, no. 4, pp. 166–173, 2021.

[11] M. Masuduzzaman, T. Rahim, A. Islam, and S. Y. Shin, "Uxv-based deep-learning-integrated automated and secure garbage management scheme using blockchain," *IEEE Internet of Things Journal*, vol. 10, no. 8, pp. 6779–6793, 2023.

[12] J. Tursunboev, Y.-S. Kang, S.-B. Huh, D.-W. Lim, J.-M. Kang, and H. Jung, "Hierarchical federated learning for edge-aided unmanned aerial vehicle networks," *Applied Sciences*, vol. 12, no. 2, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/2/670

[13] G. Fontanesi, A. Zhu, M. Arvaneh, and H. Ahmadi, "A transfer learning approach for uav path design with connectivity outage constraint," *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 4998–5012, 2023.

[14] A. Islam, A. Al Amin, and S. Y. Shin, "Fbi: A federated learning-based blockchain-embedded data accumulation scheme using drones for internet of things," *IEEE Wireless Communications Letters*, vol. 11, no. 5, pp. 972–976, 2022.

[15] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *CoRR*, vol. abs/1602.05629, 2016. [Online]. Available: http://arxiv.org/abs/1602.05629

[16] S. Bebortta, S. S. Tripathy, S. Basheer, and C. L. Chowdhary, "Deepmist: Toward deep learning assisted mist computing framework for managing healthcare big data," *IEEE Access*, vol. 11, pp. 42 485–42 496, 2023.

[17] L. Yuan, Q. He, S. Tan, B. Li, J. Yu, F. Chen, and Y. Yang, "Coopedge+: Enabling decentralized, secure and cooperative multi-access edge computing based on blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 3, pp. 894–908, 2023.

[18] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *CoRR*, vol. abs/1412.3474, 2014. [Online]. Available: http://arxiv.org/abs/1412.3474

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vision*, vol. 115, no. 3, p. 211–252, dec 2015. [Online]. Available: https://doi.org/10.1007/s11263-015-0816-y

[20] Y. Liu, Y. Gao, and W. Yin, "An improved analysis of stochastic gradient descent with momentum," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS'20. Red Hook, NY, USA: Curran Associates Inc., 2020.