IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 May 2024, accepted 8 July 2024, date of publication 24 July 2024, date of current version 30 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3432610

RESEARCH ARTICLE

UAVs and Blockchain Synergy: Enabling Secure Reputation-Based Federated Learning in Smart Cities

SYED M. AQLEEM ABBAS[1], MUAZZAM A. KHAN KHATTAK[2,3], WADII BOULILA[4], ANIS KOUBA[4], M. SHAHBAZ KHAN[5], (Graduate Student Member, IEEE), AND JAWAD AHMAD[5]

[1]Department of Computer Science, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad 44000, Pakistan
[2]Department of Computer Science, Quaid-i-Azam University at Islamabad, Islamabad 45320, Pakistan
[3]ICESCO Chair for Big Data Analytics and Edge Computing, Quaid-i-Azam University at Islamabad, Islamabad 45320, Pakistan
[4]Robotics and Internet of Things Laboratory, Prince Sultan University, Riyadh 12435, Saudi Arabia
[5]School of Computing, Engineering, and the Built Environment, Edinburgh Napier University, EH10 5DT Edinburgh, U.K.

Corresponding authors: Muazzam A. Khan Khattak (muazzam.khattak@qau.edu.pk) and Syed M. Aqleem Abbas (muhammad.aqleem@szabist-isb.edu.pk)

ABSTRACT Unmanned aerial vehicles (UAVs) can be used as drones' edge Intelligence to assist with data collection, training models, and communication over wireless networks. UAV use for smart cities is rapidly growing in various industries, including tracking and surveillance, military defense, managing healthcare delivery, wireless communications, and more. In traditional machine learning techniques, an enormous amount of sensor data from UAVs must be shared to central storage to perform model training, which poses serious privacy risks and risks of misuse of information. The federated learning technique (FL), which can be applied to UAVs, is a promising means of collaboratively training a global model while retaining local access to sensitive raw data. Despite this, FL is a significant communication burden for battery-constrained UAVs due to local model training and global synchronization frequency. In this article, we address the major challenges associated with UAV-based FL for smart cities, including single-point failure, privacy leakage, scalability, and global model verification. To tackle these challenges, we present a differentially private federated learning framework based on Accumulative Reputation-based Selection (ARS) for the edge-aided UAV network that utilizes blockchains to prevent single-point failures where we switched from central control to decentralized control, Interplanetary File System (IPFS) for off-chain model storage and their respective hash-keys on-chain to ensure model integrity. Due to IPFS, the size of the blockchain will be reduced, and local differential privacy will be applied to prevent privacy leakages. In the proposed framework, an aggregator will be selected based on its ARS score and model verification by the validators. After most validators approve it, it will be available for use. Several parameters are taken into consideration during evaluation, including accuracy, precision, recall, F1-score, and time consumption. It also evaluates the number of edge computers vs test accuracy, the number of edge computers vs time consumption for global model convergence, and the number of rounds vs test accuracy. This is done by considering two benchmark datasets: MNIST and CIFAR-10. The results show that the proposed work preserves privacy while achieving high accuracy. Moreover, it is scalable to accommodate many participants.

INDEX TERMS Blockchain, drones, edge computing, federated learning, unmanned aerial vehicles.

I. INTRODUCTION

The Industrial Internet of Things (IIoT), which is drastically changing a variety of industries such as transportation [1], Machine Learning (ML) [2], and the medical sector [3], [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Jose Saldana.

Utilizing artificial intelligence (AI), IIoT technology analyses the enormous amount of data generated by connected devices. IIoT is the primary driving factor behind Industry 4.0. However, as 5G/6G technologies develop, the Internet of Drones (IoD) is gaining popularity in smart cities. Drones also known as Unmanned Aerial Vehicles (UAVs) are operated autonomously. The majority of their management occurs remotely or with integrated autonomous instructions. UAVs first appeared in the early 1920s and were first created for military use [5].

UAVs are made up of three primary parts: channels for communication, hardware, and software. The operating system, middleware, and firmware that control UAV movement and enable dynamic decision modeling make up the software component. The hardware includes sensors, radar parts, light recognition and ranging (LiDAR) systems, and flight controller units (FCUs). The interaction between ground stations and UAVs, sometimes known as UAV swarms, is made possible through communication channels [6]. Unmanned air operations, unmanned ground vehicles, unmanned underwater vehicles, unmanned surface vehicles, and unmanned spacecraft are the five categories into which UAVs are divided depending on their intended uses [6]. The worldwide UAV market is anticipated to develop at a 16.4% compound annual growth rate (CAGR) to reach USD 58.4 billion by 2026, per a Markets study [7]. This shows how important and profitable UAVs are becoming.

Thanks to technological developments, UAVs are now employed in various industries, such as healthcare, agriculture, and the Internet of Things (IoT). More specifically, by using thermal imaging technologies, zoom cameras, and lighting capabilities, drones can quickly traverse vast distances in search of stranded hikers and provide them with guidance to return to safety. According to the Mountain Rescue Association of North America, approximately 80% of its members utilize drones for search and rescue operations [8], [9]. A company called Flyzipline, as well as Amazon Prime Air, are using drones to deliver goods in an instant [10], [11]. A new service for Uber customers is planned to utilize UAVs to provide air taxis as part of the Uber air taxi service. UAVs have been identified as the next phase in personal travel [12]. As the second largest construction company in the United States, Betel Corporation has used drones to enhance the construction process within the virtual project delivery approach [13]. Using UAVs has decreased the need for personal involvement and boosted logistical assistance in difficult environments. Artificial Intelligence (AI)-enabled UAVs will increasingly be used in various industries in the near future. These industries include Agricultural Precision Using Imagery from UAVs, transport planning, managing traffic, society governance, intelligent healthcare, and other pertinent disciplines [14]. UAVs for smart cities are rapidly growing in various industries, including tracking and surveillance, crowd sensing, object recognition, military defense, managing healthcare delivery, and wireless communications. In particular, EasyJet, a British multinational low-cost airline company, uses drones to inspect its aircraft. They state that ''Checks that normally require more than a day could be performed in a couple of hours with a higher level of accuracy'' [15], [16]. United Parcel Service (UPS), an American multinational shipping Company operated the first-ever U.S. drone COVID-19 vaccine delivery service to remote areas [17]. It was the British Broadcasting Corporation (BBC) that first used drones to provide visual context for a news report about the proposed high-speed rail network, ''HS2'' which would link London with several major English cities [16].

In order to derive intelligent predictions and make intelligent decisions for these industries, traditional AI algorithms require the transfer of enormous amounts of sensing data collected by UAVs to a central server for model training [18], [19]. Using data to train ML models turns out to be a very effective method of unlocking the potential of that data but data obtained by UAVs, such as images and videos, often contain confidential information. It is possible to experience serious privacy breaches and data misuse concerns [20], [21], [22], [23].

UAV industries must deal with two major issues: privacy and security, which have recently grown to be major problems [24]. Conventional machine learning models primarily take into account centralized, data centers, however, data owners don't wish to share their exclusive information. To address these issues, federated learning (FL), which is based on the distributed training of ML models, can be adopted [25]. As an approach to acquiring intelligence from such sensory data, FL is a privacy-preserving approach, whereby multiple UAVs can contribute to creating shared AI models without revealing any private sensory information [26], [27]. Moreover, UAVs participating in FL processes will train their local models using their sensory data, such as images captured by drones. Afterward, UAVs send local model weights to the server for aggregation at the global level. Until the desired global model accuracy has been achieved, this process is repeated [28]. As more participants join the collaborative training process, the global model in FL also becomes more successful. By including more domains in the training process, the distributed training performance may be considerably improved. This cooperation makes it possible to create a dataset that is more comprehensive and varied, which results in the discovery of a better model. During the past few years, FL has had great success in a variety of domains, including next-word recognition using the Google keyboard [29], device failure detection using FL in industrial applications [30], and immersion in virtual reality [31].

It is important to keep in mind that the centralized aggregator may be open to possible assaults, which could compromise the FL system's security [32]. The second major issue with the FL process is, model updates can disclose details about a participant's training data. According to the study discussed in reference [33], making it is possible

for attackers to possibly retrieve data from the weights that were shared and it is called a privacy leakage issue. As a consequence of this, both the academic and business communities have expressed serious interest in and worry about privacy protection [34]. In addition to single-point failures, centralized control, and privacy leakages, there are a number of other challenges associated with FL based on UAVs, including model verification, scalability, reputation-based mechanisms, and computational costs [6], [35].

As a solution to these challenges, we present a differentially private federated learning framework for edge-aided UAV networks based on accumulative reputation-based selection (ARS) and blockchains to prevent single-point failures in the transition from central to decentralized control. The following are the main accomplishments of our work:

1) Reputation-based Decentralized Federated Machine Learning for global model aggregation. It Ensures trustable and reliable aggregator selection based on their accumulative reputation score.
2) Differentially private local model sharing. It aims to ensure privacy leakage challenges in the traditional FL approach.
3) Blockchain and IPFS implementation. Blockchain tackles the challenges of single-point failure and centralization, which are present in the traditional FL approaches. It will also ensure model integrity and trust among unknown participants.
4) Smart Contract-Based Model Verification and Secure Storage. It ensures the consistent global model performance and the reliability of the selected aggregator.

The proposed framework also promotes the integration of multiple disciplines. It includes drones technology, blockchain technology, urban area planning, and data science. Collaboration between these disciplines is essential for the successful implementation of the proposed approach. It also highlights the new research gaps and the need for research efforts [6], [36]. With the help of UAV experts, efficient communication protocols can be developed, and they can also in flight control systems for UAVs [35]. Blockchain developers will ensure the decentralization of the system and the integrity of models weights while sharing with other participants [6]. They can also help in building secure and decentralized storage systems such as IPFS. Urban planners may assist regarding infrastructure, regulations, and deployment approaches [35]. At the edge computers, data scientists can perform computational tasks such as local model training and global model aggregation [36]. Additionally, they can contribute to the verification of the performance of the global model. Through the combination of these diverse fields, this framework offers a comprehensive solution for the development of secure, efficient, and scalable smart city applications.

The remainder of this paper can be categorized as follows: The second section of this paper discusses related research. The problem statement is described in Section III of the report. A detailed discussion of the proposed framework is provided in Section IV. Section V presents a discussion of the simulation and its results, and Section VI concludes the paper.

II. RELATED WORK

In this section, several studies related to our research are discussed.

A. FEDERATED LEARNING (FL)

Research on Federated Learning (FL) has recently attracted the attention of researchers and is taking Machine Learning (ML) Applications to an entirely new level in terms of data sharing and computational capability [37], [38]. In addition to preserving the privacy of users, the motivation for using FL is to take advantage of the distributed and shared nature of machine learning. Taking a look back over the previous four years, the research study [38] provides a systematic assessment of privacy and security issues associated with blockchain-based FL methodologies in smart environments [39].

B. PRIVACY-PRESERVED FEDERATED LEARNING (FL)

A major advantage of FL is the provision of privacy. However, FL also poses the risk of privacy leakage, as the user's actual data might be derived from the model weights. Numerous studies have been conducted recently to preserve privacy in FL. Liu et al. [40] and Yin et al. [41] proposed FL frameworks where training takes place locally on individual nodes, with only model updates being transmitted for centralized aggregation. Based on differential privacy (DP) techniques, Wei et al. [42] and Zhao et al. [43] developed frameworks that enhance data privacy. A DP will, however, cause training to be slowed down and accuracy to be reduced [44]. A proxy layer was added in the reference [45] and then DP was implemented. Because of the DF, the proxy layer increased overhead, and concluded that DP decreased the accuracy of the model. In terms of computational cost, using DP is more efficient than other methods, such as encryption, however, there is a trade-off between accuracy and privacy.

C. DECENTRALIZED FL FOR UAVS

A wide variety of applications have been developed for unmanned aerial vehicles (UAVs), also known as drones, including military, construction, mapping of photographs and videos, healthcare, parcel delivery, exploration of hidden areas, search, and rescue, parcel delivery, monitoring of oil rigs, power lines, precision farming, wireless communications, and aerial surveillance. According to the latest research [46], autonomous UAV networks based on artificial intelligence are thoroughly analyzed. A focus is placed on drones' significance, objectives, and functional issues. This review analyzes UAVs, swarms, types, classification, charging, and standardization. In light of recent research and development, UAV applications, difficulties, and security

concerns are explored. In a thorough analysis of more than 100 UAV articles, the classification of autonomous features, network resource planning and management, multiple access protocols, and power management and energy efficiency were the main topics addressed. Based on the literature review and analysis of UAV networking, it has been discovered that AI-based UAVs are a technologically and commercially viable paradigm for developing and deploying such self-governing networks in the future.

Several recent surveys and review papers have discussed the use of UAVs to assist smart cities in providing services and the opportunities and challenges associated with the use of blockchain-enabled FL integration to assist smart cities. This article [47] presents fundamental ideas, which also examine blockchain-based FL's potential in Mobile Edge Computing (MEC) networks. Although blockchain is a promising solution to tackle the single-point failure issue in FL, however Blockchain-based FL poses several key design challenges, including computation costs, resource distribution systems, incentive systems, privacy concerns such as privacy leakage, model verification, scalability, and consensus mechanism such as reputation-based selection of aggregator. Following that, the author [47] examines well-known mobile edge computing (MEC) fields such as edge data sharing, edge content caching, and edge crowd sensing using blockchain-based FL. Several significant research challenges are also highlighted in the report, as well as possible directions for future research. Saraswat et al. [6] have conducted a survey that fills in the gaps and offers a taxonomy of solutions for BC-based FL in UAVs for B5G networks. A comparison is made between the potential benefits of a reference architecture and those of conventional BC-based UAV networks. Open issues and challenges are discussed along with potential future directions. It is evident from these surveys that UAV-based FL is an active research area with lots of potential for smart cities.

To implement an edge-aided UAV network, Tursunboev et al. [58] developed a hierarchical FL algorithm, referred to as the hierarchical FL algorithm. As an intermediary aggregator, this algorithm utilizes edge servers located in base stations. As a means of improving learning performance in wireless UAV networks under bandwidth constraint conditions and dynamic channel conditions, Liu et al. [49] proposed a novel distributed learning architecture known as hybrid split and federated learning (HSFL). The algorithm is based on the asynchronous model training process of FL and split learning's (SL) mechanism of splitting models. Tursunboev et al. [58] developed the hierarchical FL algorithm for edge-aided UAV networks with non-i.i.d distributions of data. Rather than performing FL-related computations on the UAVs themselves, the author used edge servers near the UAVs to reduce the cost of UAV computation. Sharma et al. [51] have constructed a Synchronous Federated Learning system and compared it to Asynchronous Federated Learning (AFL) to save FL processing time for the UAVs. Despite taking longer to execute than AFL, there won't be any kind of packet or loss of data. The proposed solutions resulted in improved global model iterations and greater performance in comparison to AFL. Although most of these works involve UAVs as active components of the network system in order to carry out the tasks, they do not take into account the scalability of the system and the privacy of the model weights into account.

Zhao et al. [56] have recently proposed a blockchain-based crowd-sourcing FL system as a means for better understanding their customer base. The Industrial Internet of Things (IIOT) can be secure through the use of blockchain-based FL technology, which Aditya Pribadi and colleagues have introduced in a framework that incorporates FL technology within a trusted execution environment (TEE). Their approach improved privacy and security in federated learning by taking into account model accuracy impact. Researchers conducted a range of experiments using pre-trained Convolutional Neural Network (CNN) models and benchmark datasets to validate their proposed solution [48]. A blockchain-enabled horizontal FL for UAVs connected through the 5G network has been proposed by Feng et al. [52]. Instead of relying on a central server, the author authenticated multiple cross-domain UAVs and aggregated models using smart contracts. Abunadi et al. [54] developed a novel technique for performing secure communication and image classification in UAV networks. It consists of three phases, including clustering, secure communication using blockchain technology, and FL image classification using FL technology, as well as the validation of the framework's performance using pre-trained CNN models. Almost all of the studies above suggested blockchain as a solution to UAV-based FL, however, none of them consider a reputation-based mechanism for monitoring the accuracy and participation of the participants in the FL process for aggregator selection to perform global model aggregation.

The edge server, which is positioned between the UAVs, is essential for minimizing communication and computation costs [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59]. Battery-limited UAVs face heavy communication loads due to FL's frequent local training and global synchronization. In terms of model aggregation and fair incentives, the quality of UAV model updates can vary significantly. A Byzantine-robust aggregation rule, a model profit allocation rule, as well as reputation mechanisms, were developed by Wang et al. [53] by analyzing historical learning records and examining record freshness for weight assignment. This was to ensure credible recruitment and to prevent free-riders of UAVs. FL-aided crowd-sensing services were to be facilitated by fair incentives and robust aggregation. In spite of this, they did not tackle the single point of failure. A further challenge to applying FL algorithms to UAVs is their energy

TABLE 1. A qualitative comparison of our work with recent work in terms of its advantages.

Reference	Year	SPF[1]	PL[2]	OMS[3]	MV[4]	S[5]	RM[6]
Kalapaaking et al. [48]	2023	✓	✓	×	×	×	×
Liu et al. [49]	2023	×	×	×	×	×	×
Ranathunga et al. [50]	2023	✓	✓	✓	✓	×	×
Ayushe et al. [51]	2023	×	✓	×	×	×	×
Chaosheng Feng et al. [52]	2022	✓	✓	×	×	×	×
Yuntao Wang et al. [53]	2022	×	×	×	×	×	✓
Abunadi et al. [54]	2022	✓	×	×	×	×	✓
Hui Li et al. [55]	2022	×	×	×	×	×	×
Yang Zhao et al. [56]	2021	✓	✓	✓	×	×	✓
CHongming Zhang et al. [57]	2020	×	×	×	×	✓	×
Proposed Work	2023	✓	✓	✓	✓	✓	✓

[1] Single Point Failure
[2] Privacy Leakage
[3] Off-Chain Model Storage
[4] Model Verification
[5] Scalability
[6] Reputation Mechanism

consumption. Pham et al. [60] suggested an energy-efficient FL model for UAVs. An FL framework with a Ground Fusion Centre (GFC) acting as an integrator was investigated by Zhang and Hanzo [57] aimed at lowering the complexity of communication associated with UAV swarms in remote settings. Alternatively, the researchers of [61] used UAVs as a communicative bridge between users and edge nodes in order to reduce network latency. While these works consider communication costs and delays, they neglect to consider the privacy of model weights when sharing with others. As an incentive to encourage participants to participate in the FL process, Zhang and his team [18] proposed a reward system between UAVs and task publishers. Researchers [62] developed a number of services for a mobile edge computing network that utilized UAVs as dual-purpose tools for communication and computation, which included offloading computational tasks, distributing resources, and optimizing UAV positioning. In a study published in [63], Yang et al. use FL to conduct traffic monitoring in urban areas using a swarm of UAVs. Nearly all of these studies used FL as a base model for UAV networks, but none of them considered the single-point failure challenge of FL in their solutions.

D. OUR CONTRIBUTIONS

For UAVs to overcome the aforementioned challenges, a better FL framework is still needed. As far as we are aware, few recent studies address single-point failures and few focus on privacy-preserving FL, but no framework for UAVs that focuses on single-point failure, privacy leakage, aggregator selection based on reputation, global model verification, and scalability all at the same time. Decentralized edge-assisted FL frameworks are required for smart cities to ensure the preservation of privacy while achieving high accuracy. Table No. 1 presents a qualitative comparison of our work with recent work in terms of its advantages. As a blockchain-based federated learning framework, our framework effectively addresses the discussed challenges faced by federated learning (FL) for UAVs, namely single point failure, privacy leakage issues during model sharing in the federated learning process, off-chain model storage to ensure the model integrity and to reduce the blockchain size, evaluating the scalability of the participants in the FL process, performance verification of the global model by the validators, reputation mechanism for the participants.

III. PROBLEM STATEMENT

Unmanned aerial vehicles (UAVs), also known as drones, have become a key enabler for a variety of applications in the age of smart cities, from delivery and surveillance to urban planning and disaster management. Despite their potential, utilizing the massive amounts of visual data that drone cameras collect effectively is a significant challenge. Traditional data processing techniques frequently involve sending this data to a centralized server, raising serious privacy issues and communication overhead. These issues are particularly pressing in the context of UAV-based FL due to the sensitive nature of the data collected and the potential for misuse if not properly secured. Our research aims to propose a cutting-edge, privacy-preserving, reputation-based FL framework that is powered by blockchain and IPFS for the use of UAV swarms in smart cities. Based on the

visual information captured by drone cameras, the framework envisions the use of local model datasets. These datasets are then sent to the edge computers participating in our FL framework. Our study was guided by the following research question:

RQ: How can UAV-based federated learning be optimized for decentralized aggregation, off-chain storage, model verifiability, privacy, and scalability in smart cities?

The scalability of the proposed framework is crucial given the distributed nature of FL and the potential for large-scale deployment of UAV swarms in smart cities. Addressing scalability challenges is essential to ensure the framework's effectiveness and efficiency in real-world applications.This study aims to contribute to the research on UAVs, FL, blockchain, and smart cities, thus paving the way for privacy-preserving machine learning in UAVs. The proposed framework introduces unique innovations such as the use of blockchain and IPFS, a reputation-based participant selection mechanism, and a focus on scalability. These aspects set our work apart from existing research and provide novel solutions to the challenges currently faced in UAV-based FL.

IV. PROPOSED FRAMEWORK

Our proposed framework consists of four primary components, namely, System Initiation and Registration, Differentially private local model Computation, communication model, and global model generation. The details of each component are as follows:

A. SYSTEM MODEL

Three layers comprise our proposed framework: the UAV layer, the Edge Cloud layer, and the Blockchain layer. The UAV layer is responsible for data collection. It consists of drones that fly over smart cities and take pictures relevant to their particular tasks. The purpose of this layer is to collect real-world data that will be used for the machine-learning process. Using the high-resolution image/video capture capability of onboard sensors and cameras, UAVs facilitate data collection in various applications such as surveillance, traffic monitoring, disaster management, and forest fire detection, among others [64]. UAVs are an integral part of our smart city framework in collecting localized data effectively, while the Edge Cloud layer is responsible for ensuring the performance of data processing on a real-time basis.

All edge servers, aggregators, and the validators reside in the edge cloud layer. UAV images are processed by the edge servers to produce differentially private local models, ensuring privacy in the learning process. Typically, a subset of edge servers' validators confirms the accuracy of the global model. An aggregator chosen by the blockchain layer based on reputation will aggregate differentially private local models into a global model.

The blockchain layer ensures data integrity due to its immutability advantage. Blockchain ensures data integrity by providing an unchanging ledger to verify models and manage reputation. Further, hash-based record storage ensures that uploaded models cannot be tampered with and are stored and accessed securely. Additionally, each participant is authenticated using their cryptographic signature. According to Saraswat et al. [6], blockchain's tamper-resistant ledger is crucial for smart city applications involving UAVs, ensuring secure data sharing and analysis. It overcomes the challenges associated with single-point failure and centralized control experienced by FL-based solutions. It is also responsible for participant registration, global model verification, reputation-based aggregator selection, and storing models off-chain and their respective hash keys on-chain. By storing models off-chain, computational costs would be reduced since storing models on-chain would increase the blockchain size, resulting in high computational costs for miners and a high transaction fee for participants. That is why we chose IPFS as a decentralized storage solution to implement an off-chain model of storage. Two components make up the blockchain layer: the on-chain and the off-chain. On-chain components of the system are managed by smart contracts, which maintain participant registrations, addresses of active participants, cumulative reputation scores for all participants, hash keys for uploaded models, and mechanisms for selecting an aggregator based on cumulative reputation scores.

The layers interact continuously through data collection, processing, and model generation. All participants and miners must be registered as the first step. In the UAV layer, Images captured by the UAVs flying over the smart city will be sent to the edge cloud layer as local datasets. As the next step, edge servers retrieve the initial global model from IPFS. Afterward, they train the local models on the dataset collected from the UAV layer and upload differentially private local models to the blockchain where IPFS is used for off-chain model storage. In addition, model hash keys and local model test accuracy are stored in the smart contract. As soon as the blockchain layer transmits the aggregator's address to the Edge Cloud layer, the selected aggregator will be able to perform global model aggregation based on the hash keys stored on-chain and differentially private local models from IPFS.

After validating the global model's performance on their validation dataset, the validators submit their votes to the smart contract. A global model is accepted if two-thirds of the votes are cast favorably. If a model is rejected, the miner with the next highest reputation score is selected for the global model aggregation. The proposed model can be seen in Figure 1, which illustrates how it works. Federated learning enables collaborative training while maintaining privacy. A differential privacy concept is used in the proposed framework. It will incorporate Gaussian random white noise into the gradients of the local model before training. Unlike traditional FL, it will address the privacy leakage issue [65]. As a result, UAVs contribute to training an accurate global model while preserving the privacy of their raw data [66]. Additionally, our proposed Accumulative Reputation based selection (ARS) mechanism ensures that only a trusted and reliable aggregator is selected for global model aggregation.

A global model is then returned to the UAV layer from the edge cloud layer to guide the drones in their tasks. By utilizing the global model, UAVs can provide services to smart cities, ensuring that many smart city applications are operated efficiently. A secure, privacy-persevered, scalable, and reliable distributed learning system is created using UAVs, edge computing, federated learning, blockchain, and IPFS.

B. CHALLENGES AND CONSIDERATIONS

For the proposed system to be implemented successfully, it is vital to discuss potential challenges and considerations. These include regulatory compliance, technological compatibility, scalability, and community acceptance.

1) REGULATORY COMPLIANCE:

It would be crucial to consider local, national, and international laws and regulations while performing UAV-based tasks in smart cities. The use of UAVs may lead to the generation of large quantities of data. Private information may be contained in those data. Thus, privacy regulations such as GDPR and CCPA [5] will significantly impact the management and sharing of these data [67]. Furthermore, blockchains provide a decentralized approach without the control of a central authority. There may be changes in country-specific regulations, which may raise questions regarding blockchain transactions and smart contracts since different countries have different laws and regulations [68].

2) TECHNOLOGICAL COMPATIBILITY:

To successfully integrate advanced technologies such as UAVs, edge computing, federated learning, and blockchain, domain experts will be required. Furthermore, the hardware, software, and data they use are heterogeneous. As a result, different protocols must be compatible between edge devices and blockchain [69]. Among the challenges associated with blockchain is interoperability. Another requirement is the interoperability of different blockchain frameworks [70].

3) SCALABILITY:

The concept of scalability pertains to the ability to handle the growing number of drones and edge computers while maintaining the performance of the global model. Another major concern is developing a scalable system for smart cities [69]. As part of this work, we have tested its scalability in terms of an increasing number of UAVs while maintaining the outstanding performance of the global model. Other scalability limitations should also be considered, such as non-IID data, heterogeneous type of data, and high transaction volumes [71].

4) COMMUNITY ACCEPTANCE:

There is also the issue of community acceptance, including public perceptions of UAVs and privacy concerns. As a result, trust is engendered through transparent data privacy policies, ethical use guidelines, and community engagement [47]. In addition, public awareness of advanced technologies such as blockchain and federated learning is required to gain community acceptance.

C. SYSTEM INITIATION AND REGISTRATION

In this article, We consider a system $\mathcal{S}$ with a set of unmanned aerial vehicles (UAVs) $\mathcal{U} = U_1, U_2, \ldots, U_n$ and a set of edge servers $\mathcal{E} = E_1, E_2, \ldots, E_m$. To participate in the federated learning process, an edge server must call the $\mathcal{R}(\cdot)$ and $\mathcal{M}(\cdot)$ methods of the smart contract $\mathcal{SC}$. Registration is divided into two types, participants and miners, in our proposed framework.

1) Let $\mathcal{P}$ be the set of all participants, where the participants are represented by P_i. A registration function can be denoted by $\mathcal{R}(P_i)$, which registers participant P_i with the smart contract $\mathcal{SC}$. A participant registration process can be formalized as follows:

$$\forall P_i \in \mathcal{P}, \quad \mathcal{R}(P_i) \rightarrow \mathcal{SC} \tag{1}$$

According to the equation above, for every participant P_i in the set $\mathcal{P}$, the registration function $\mathcal{R}(P_i)$ applies, creating a link between each participant and the smart contract $\mathcal{SC}$.

2) Participant P_i can also be a miner. Let $\mathcal{M}(P_i)$ denote the miner registration function, which registers participant P_i as a miner with the smart contract $\mathcal{SC}$. Formally, the miner registration process can be defined as:

$$\forall P_i \in \mathcal{P}, \quad \mathcal{M}(P_i) \rightarrow \mathcal{SC} \tag{2}$$

In the same way, for every participant P_i in the set $\mathcal{P}$, the miner registration function $\mathcal{M}(P_i)$ applies, registering each participant as a miner with the smart contract $\mathcal{SC}$.

In these equations, $\forall$ stands for 'for all', and $\rightarrow$ corresponds to the function mapping that links the miners and participants to the smart contract $\mathcal{SC}$. In our framework, any participant can act as a miner through the process of miner registration. Miners play a critical role in our system because they aggregate local models into global models. We are using the Accumulative Reputation Based Selection (ARS) mechanism to select a miner, and the selected miner will be called an aggregator. This mechanism ensures that the most accurate and reliable participant is selected for aggregation.

D. DIFFERENTIALLY PRIVATE LOCAL MODEL COMPUTATION

By incorporating noise into the computation of local model updates at each edge server, we ensure differential privacy in federated learning. In Algorithm 1, we describe this process as Differentially Private Local Training.

Assume a dataset $D_i = {x^{(i)}, y^{(i)}}_{i=1}^{n}$ consisting of n samples. Here, each $x^{(i)}$ represents an image captured by the drone's camera, and $y^{(i)}$ denotes the corresponding label. The global model, represented by θ, aims to minimize the loss function L.

Algorithm 1 Differentially Private Local Model Computation

Require: Set of edge servers $E = \{E_1, E_2, \ldots, E_m\}$, initial global model θ_g, set of local datasets $D = \{D_1, D_2, \ldots, D_m\}$, loss function $L(\cdot)$, noise multiplier σ
Ensure: Set of privacy-preserving local model updates $\tilde{\Theta} = \{\tilde{\theta}_1, \tilde{\theta}_2, \ldots, \tilde{\theta}_m\}$

procedure LocalModelTraining($E, \theta_g, D, L, \sigma$)
  $\theta_g \leftarrow$ loadGlobalModelFromIPFS()
  for $i = 1tom$ **do**
    $\theta_i \leftarrow \theta_g$
    $\Delta\theta_i \leftarrow$ trainModel(D_i, L, θ_i)
    $\tilde{\theta}_i \leftarrow \Delta\theta_i + \eta_i$
    $\tilde{acc}_i \leftarrow$ testModel($\tilde{\theta}_i, D_{i,\text{test}}$)
    hashKey$_i \leftarrow f$(storeOnIPFS($\tilde{\theta}_i$))
    f(storeOnContract($\tilde{acc}_i$, hashKey$_i$))
  end for
  return $\tilde{\Theta}$
end procedure

Each edge server E_i ($i \in 1, 2, \ldots, N$) holds a local dataset D_i and calculates a local model update $\Delta\theta_i$ to minimize the loss on its local dataset:

$$\Delta\theta_i = \arg\min_{\theta} \frac{1}{n_i} \sum_{j=1}^{n_i} L(x_i^{(j)}, y_i^{(j)}; \theta), \tag{3}$$

where n_i represents the number of samples at edge server E_i and $(x_i^{(j)}, y_i^{(j)})$ is the j^{th} sample at edge server E_i.

Following the fetch of the global model from the Interplanetary File System (IPFS) via the hash key supplied by $\mathcal{SC}$, each edge server E_i starts training on its local dataset $D_i = d_1, d_2, \ldots, d_{n_i}$. The local model update $\Delta\theta_i$ is computed as:

$$\Delta\theta_i = \arg\min_{\theta} \frac{1}{n_i} \sum_{j=1}^{n_i} L(x_i^{(j)}, y_i^{(j)}; \theta), \tag{4}$$

where $(x_i^{(j)}, y_i^{(j)})$ is the j^{th} sample in D_i, $L(\cdot)$ represents the loss function, and n_i is the total number of samples in D_i. To ensure differential privacy, Gaussian noise is added to $\Delta\theta_i$ which is a common approach in differential privacy, resulting in an updated local model with privacy preservation $\tilde{\theta}_i$:

The Gaussian noise is a type of white noise often added to data or computations to disguise original values and maintain privacy [72]. We added Gaussian random noise to a convolutional neural network (CNN) local model during the optimization phase. To implement this, we utilized the TensorFlow-Privacy library's DPKerasSGDOptimizer. Gaussian noise is incorporated into the gradients before the local model parameters are updated. The noise multiplier parameter determines how much noise to add. A noise multiplier parameter has a range of values beginning at zero and does not have a strict upper limit. In practice, it is usually chosen to be a relatively small value since a larger value would increase in noise, and an increase in noise will decrease model accuracy.

$$\tilde{\theta}_i = \Delta\theta_i + \eta_i. \tag{5}$$

we are using the test accuracy acc_i to calculate the Accumulative Reputation Score (ARS) for every edge server, so it is essential to save test accuracy acc_i on the contract $\mathcal{SC}$. after it, the noisy local model $\tilde{\theta}_i$ will be stored on IPFS and its hash key hashKey$_i$ on the contract $\mathcal{SC}$.

This approach protects the computation of local model updates at each edge server by the principle of differential privacy.

E. COMMUNICATION MODEL

Decentralized ledgers such as the blockchain ensure data integrity and privacy. It will store the smart contract $\mathcal{SC}$ that will perform the following operations in our proposed framework:

1) Registrations of participants and miners
2) List of all the active participants
3) Keeps track of current aggregator
4) Accumulative Reputation scores (ARS) of all the participants.
5) Validators save their votes in favor or against the aggregated global model to the smart contract, and the smart contract will make the selection of the miner who has the highest reputation score for global model aggregation
6) IPFS hash keys $\mathcal{H}$ of local and global models. Only hash keys $\mathcal{H}$ will be saved on the smart contract to ensure model integrity, as changing even one value in the model will also change its hash key.

For our local and global models, we use Interplanetary File System (IPFS) as off-chain storage, which will reduce the blockchain size and transaction gas fees, and only hash keys $\mathcal{H}$ will be saved on the smart contract to ensure model integrity. A model can be saved on IPFS, and a hash key can be obtained as follows:

$$M_i \xrightarrow{\text{IPFS}(M_i)} h_i$$

F. GLOBAL MODEL AGGREGATION AND VERIFICATION

Global models are created by aggregating the differentially private local models. The Global Model Aggregation and Verification Algorithm, as presented in Algorithm 3, accomplishes this.

For each participant P_i, let us denote the test accuracy as acc(P_i) and the function $\mathcal{R}(\cdot)$ denotes the reputation score for each participant P_i. Let's also denote the function $\mathcal{A}(\cdot)$ as the aggregate function used by the miner with the best reputation. It takes local model weights as input and combines them into a global model. In our proposed framework, local model weights are combined using FL averaging strategy to produce a global model. Based on this strategy, an average of local model updates from each participant in the federated learning process is calculated. As a result, a global model update is calculated. Algorithm No 3 describes the FL averaging

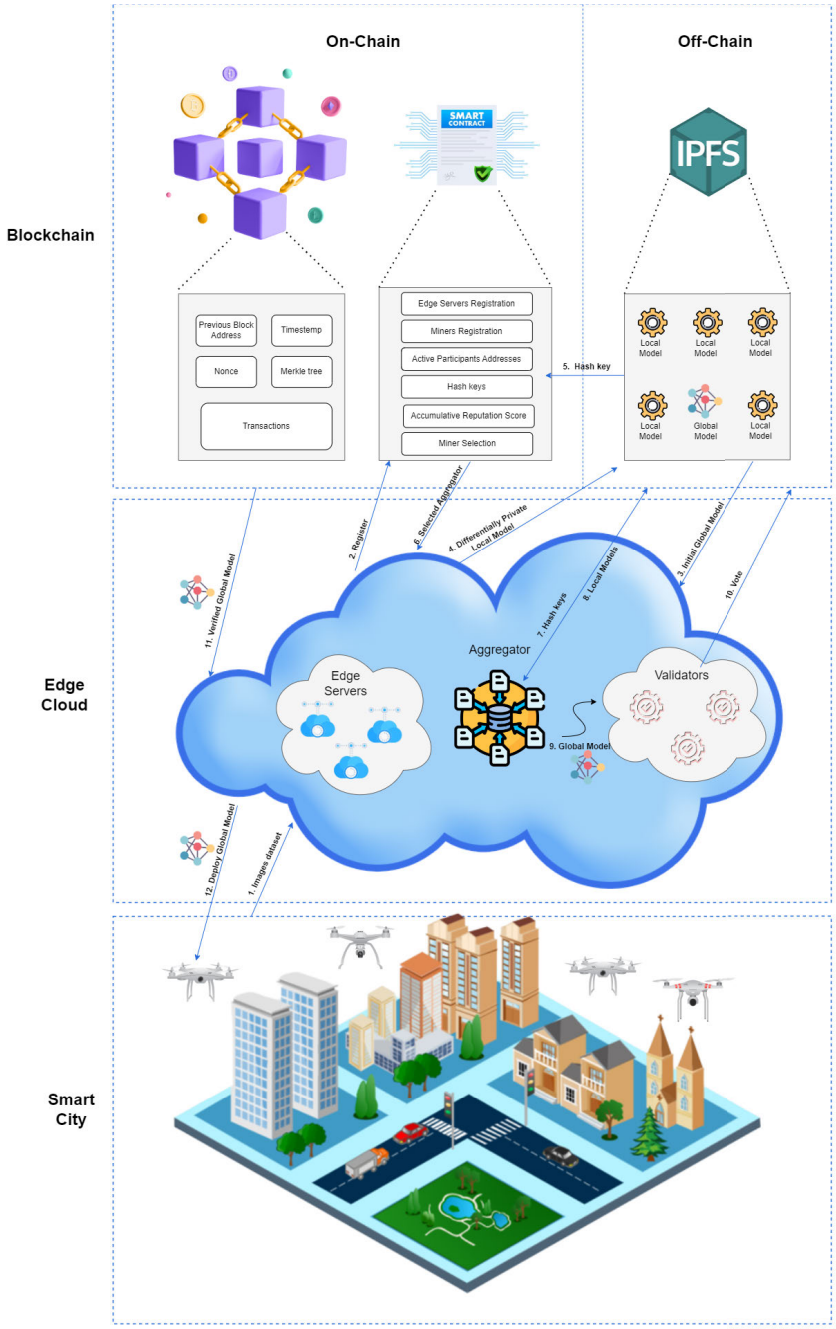

FIGURE 1. The proposed framework.

strategy steps involved in combining local model weights into a global model.

1) Reputation Score Calculation: For each participant P_i in the set of participants $\mathcal{P}$, the reputation score $\mathcal{R}(P_i)$ is calculated based on the harmonic mean of the test accuracy $\text{acc}(P_i)$ and the contribution count of the participant. Contribution counts indicate the number of times that a participant has participated in federated learning. Using the harmonic mean, accuracy, and contribution count are given equal importance, which ensures consistency of participation and high accuracy are recognized. In formal terms, reputation score calculation is as follows:

$$\max_{P_i \in \mathcal{P}} \mathcal{R}(P_i) = \frac{2 \times (\text{acc}(P_i) \times \text{contribution}(P_i))}{\text{acc}(P_i) + \text{contribution}(P_i)} \quad (6)$$

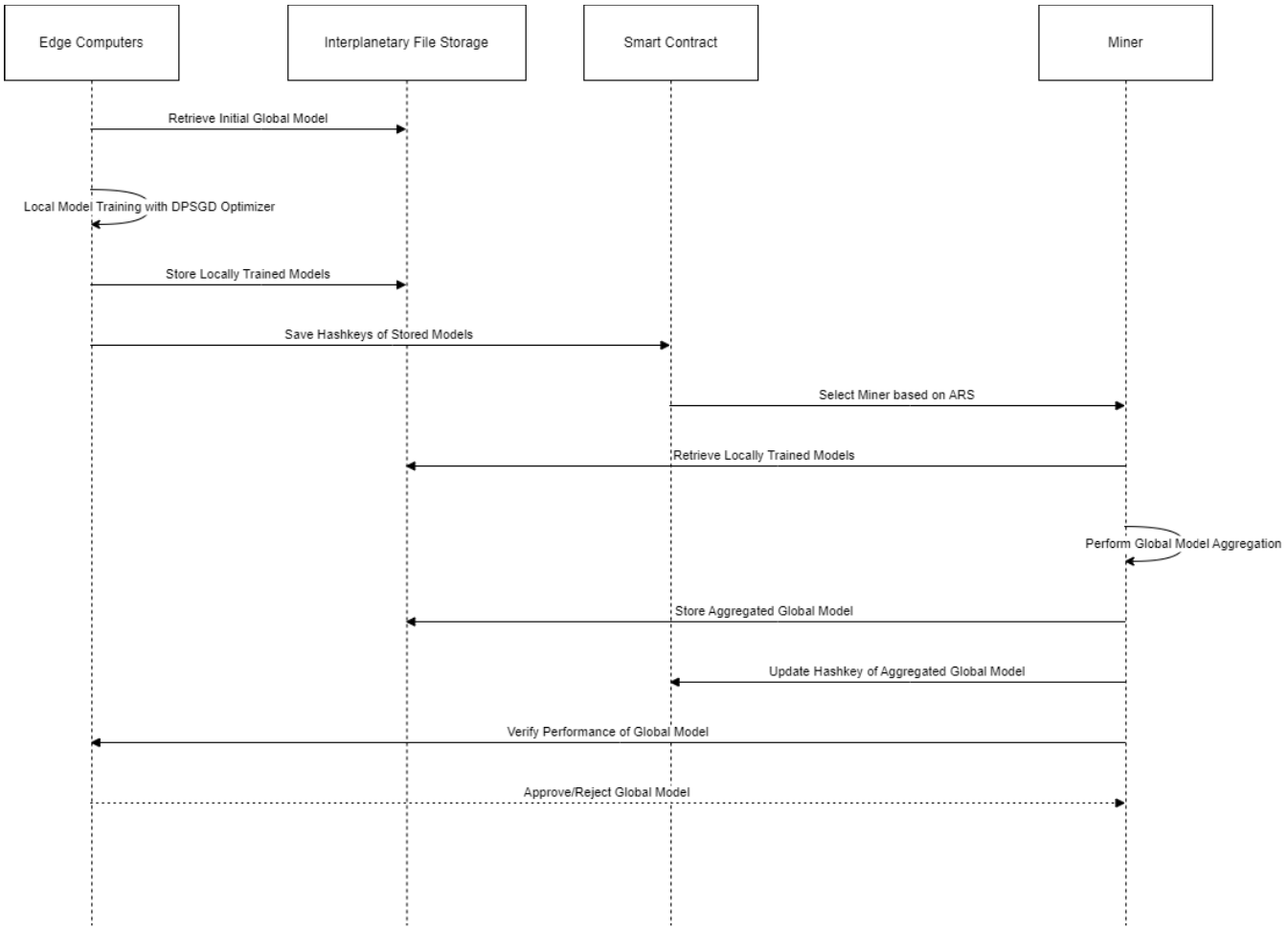

FIGURE 2. Sequence diagram of the proposed framework.

Algorithm 2 Global Model Aggregation

Require: Ethereum address of the miner with highest reputation score $m_{\max}$, set of locally trained models $\Theta = \theta_1, \theta_2, \ldots, \theta_m$

Ensure: Updated global model θ'_g

procedure GlobalModelUpdate($m_{\max}$, Θ)
  for $i = 1tom$ **do**
    hashKey$i \leftarrow f$(getHashFromContract($m$max))
    $\theta_i \leftarrow f$(loadModelFromIPFS(hashKey$_i$))
  end for
  $\theta'_g \leftarrow$ aggregateModels(Θ)
  hashKey$_g \leftarrow f$(storeOnIPFS(θ'_g))
  f(storeOnContract(hashKey$_g$))
  f(triggerValidation())
  return θ'_g
end procedure

Using this equation, reputation scores are calculated for each of the participants P_i of the set of participants $\mathcal{P}$. An individual's reputation score is the harmonic mean of their test accuracy $/text[acc](P_i)$ along with their contribution count. Any miner who has the highest Accumulative Reputation Score (ARS) will be selected as an aggregator.

2) Global model: The participant P_j with the highest reputation score $\mathcal{R}(P_j)$ aggregates the models globally. The global model G is calculated as:

$$G = \mathcal{A}\left(\{M_i | P_i \in \mathcal{P}\}\right)$$

where M_i is the local model from participant P_i.

3) Model verification: Verifying global model performance requires a 2/3 majority after model aggregation. A formal definition of the model verification problem is as follows:

$$V(G) = \frac{1}{|\mathcal{P}|} \sum_{P_i \in \mathcal{P}} V_i(G) \geq \frac{2}{3}$$

where $V_i(G)$ is a binary variable indicating whether participant P_i verifies the global model or not (1 for verification, 0 otherwise), and $|\mathcal{P}|$ is the total number of participants.

Algorithm 3 Federated Averaging Strategy for Global Model Aggregation

Require: Set of local model updates $\Delta\Theta = \Delta\theta_1, \Delta\theta_2, \ldots, \Delta\theta_m$, set of participants $\mathcal{P} = P_1, P_2, \ldots, P_m$

Ensure: Updated global model θ'_g

procedure GlobalModelUpdate($\Delta\Theta, \mathcal{P}$)
  Initialize global model θ_g
  for each iteration **do**
    for $i = 1tom$ **do**
      Receive local model update $\Delta\theta_i$ from participant P_i
    end for
    Calculate average of local model updates $\theta'_g = \frac{1}{m}\sum_{i=1}^{m}\Delta\theta_i$
    Send updated global model θ'_g to each participant P_i
  end for
  return θ'_g
end procedure

Global model verification is an integral part of our framework to ensure consistent model performance. It is performed by the validators following the global model aggregation step by the aggregator. Except for the aggregator, all active miners are validators that verify the performance of the global model on the validation dataset. Afterward, they can vote for or against the smart contract. The model will be accepted if 2/3 of the votes are in favor of it. Upon rejection, the next miner with the highest ARS will be selected as an aggregator. The aggregator calculates the new global model, which will be revalidated by the validators. This process continues until a global model is accepted. Using a two-thirds majority rule, the smart contract handles model acceptance, rejection, and selection of the next miner. The 2/3 majority rule ensures the accuracy and reliability of the global model.

In Figure 2, we present a diagram that shows the sequence of events that will take place in our framework. As a result of this framework, the challenges stated above have been fully addressed. Combining local model generation, a robust communication model and global model aggregation can allow us to achieve federated learning while ensuring differential privacy.

V. EXPERIMENTS

A. HARDWARE AND SOFTWARE CONFIGURATION

The Hardware we used to conduct experiments is a computer having an 11th generation intel(R) core(TM) i9 3.50GHz CPU and 64GB RAM. The 64-bit operator system installed on it was Windows 11. All the experiments were performed on Anaconda Jupyter Notebook with Python 3.12 as the development environment. We simulated edge computers and the federated learning process using the Flower Framework [73]. We used TensorFlow by Google for the local models and TensorFlow Privacy by Google for the differential privacy integration. Solidity programming language was used to develop the smart contract while Ganache Tool was used to build a local Ethereum blockchain.

B. LOCAL MODELS

The experiments were conducted using two different local models for different datasets. In Model No. 1, there are five convolutional layers, three max-pooling layers, three batch normalization layers, and two fully connected layers for classification. A detailed description of the architecture of local model number 1 can be found in figure 3.

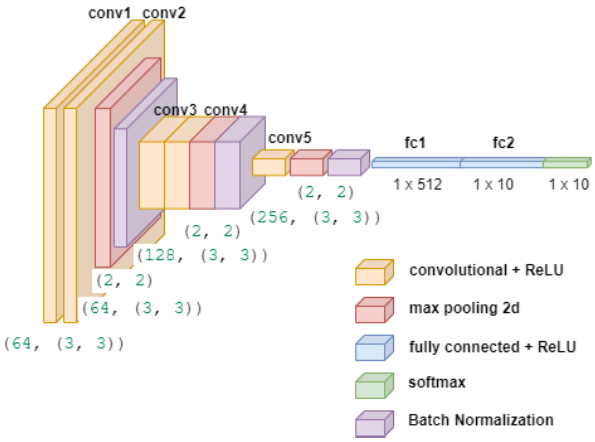

FIGURE 3. Local Model No 1.

The first layer consists of 64 convolutional filters of three-by-three sizes, while the second layer is also a convolutional layer of the same size. A third layer is a 2D max pooling layer of two by two to reduce the less important features and improve the focus of computation on the most important features. In the fourth layer, a batch normalization layer is applied to normalize the output of the layer within a batch of data, in order to improve the stability and speed of training. Five and six layers are composed of convolution layers with 128 filters of 3 by 3. In the seventh layer, a 2D max pooling layer of size 2 by 2 is applied. As the eighth layer, there is a batch normalization layer that is used to normalize the output within a batch. The ninth layer is a convolution layer consisting of 256 filters of 3 by 3 dimensions. Layers ten and eleven are devoted to 2D max-pooling of 2 by 2 and batch normalization. In the last two layers, there are two fully connected layers, measuring 1 by 512 and 1 by 10, respectively. A Relu activation function was used in the hidden layers and a softmax function was used in the last layer of classification.

In the second model, we used a ResNet-50 pre-trained model that is trained on ImageNet data to extract features. ImageNet is a large dataset containing millions of labeled images spanning thousands of categories [80]. The architecture of local model No 1 is presented in figure 4. As part of this second model, ResNet-50 is used as an extractor of

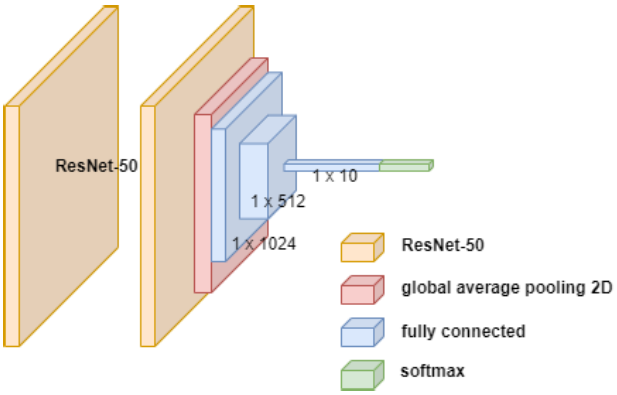

FIGURE 4. Local Model No 2.

features and combined with 4 layers of classification. Each of the four layers of classification consists of a global average pooling 2D layer, a densely connected layer of size 1 by 1024, 1 by 512, and 1 by 10 respectively. For the hidden layers, Relu activation was used, and for the classification layer, the softmax function was used. The reason behind adding a global model average pooling layer is to reduce overfitting by reducing the number of parameters and focusing on the important features. In this case, we prefer the global average pooling 2D over the max-pooling 2D layer. This is because we wish to retain the spatial information, which can be achieved by calculating the average of each feature map and preserving the spatial structure, and it is also often used in the classification layers.

C. A CASE STUDY: FEDERATED LEARNING FOR FOREST FIRE DETECTION IN A SMART CITY USING UAVS AND EDGE COMPUTING

1) SCENARIO CONTEXT

In today's smart city environment, disaster management is of utmost importance. This case study aims to examine the practical utility of the proposed UAV-based federated learning framework in a smart city scenario, focusing on disaster management related to forest fire detection. The setup involves UAVs equipped with high-resolution cameras flying over smart cities and capturing images of forested areas. Edge computers receive these images as private datasets that participate in the FL process. Convolutional neural networks (CNNs) are trained on these private datasets by edge computers.

2) SIMULATION AND DATASET

This case study utilizes the FLAME dataset [74] available on IEEEDataPort. There are 47,992 RGB images in the Flame dataset that were captured by drone cameras in Northern Arizona. Images were captured using different drones and cameras. The dataset consists of two classes: Fire and No Fire, and each image size is downscaled to (254,254,3), then the pixel value is normalized between 0 and 1 to improve model training efficiency. The sample images from the FLame dataset are presented in figure 5.

FIGURE 5. FLAME dataset sample images.

In this simulation process, we divided this dataset into 75% for the edge computers and 25% to test and evaluate the global model's performance. As their local dataset, 75% of the samples from the flame dataset were equally divided among the number of edge computers. Additionally, each edge computer took 10% validation samples from its local datasets to monitor the performance of its local model. This 10% validation was also used to monitor and save their local model accuracy to smart contracts for the reputation mechanism discussed in the proposed framework section. We simulated 10 edge computers and used Model No. 1 presented in Figure No 3 as the local model but with minor changes. To prevent overfitting, we added a 0.2 dropout layer after each batch normalization layer. A Rectified Linear Unit (ReLU) is used in the hidden layers and a sigmoid is used in the output layer to normalize the output to a fire or no fire probability. The local model is trained by the optimizer with 0.001 as the learning rate. Binary cross-entropy is used as a loss function to measure classification performance. Moreover, each local model is trained with 32 batch sizes and 20 epochs.

A trained local model is stored in IPFS by each edge computer and its hash keys are stored in the smart contract. The Accumulative Reputation-based Selection (ARS) mechanism is a key innovation in this framework. The ARS system is governed by a smart contract enabled by blockchain technology and allows the selection of a suitable edge computer as an aggregator based on its historical performance and reliability. Through this reputation-based mechanism, the risk of selecting an inefficient or compromised aggregator is significantly reduced, thereby improving the overall security and performance of the system. Once the global model has been aggregated by the selected aggregator and validated by the validators, it can either be used for the next round, or deployed to 5G-enabled UAVs for forest fire detection. When a UAV detects a fire, it immediately alerts the edge computer, which promptly notifies the local authorities.

3) RESULTS AND METRICS

After 5 FL rounds, the global model's accuracy stands at a staggering 98.98%, with a loss of just 0.24. Precision and recall stand at 98.91% and 99.48%, respectively, leading to an F1 Score of 99.19%. The confusion matrix of the global model is presented in Figure 6 where 0 reflects no fire and 1 reflects fire prediction. The high recall rate ensures almost all fire instances are correctly identified, making the system exceedingly reliable. The high F1 score of 99.19% indicates a balanced system capable of identifying and precisely locating potential fire outbreaks. It is evident from these results that the proposed work is effective.

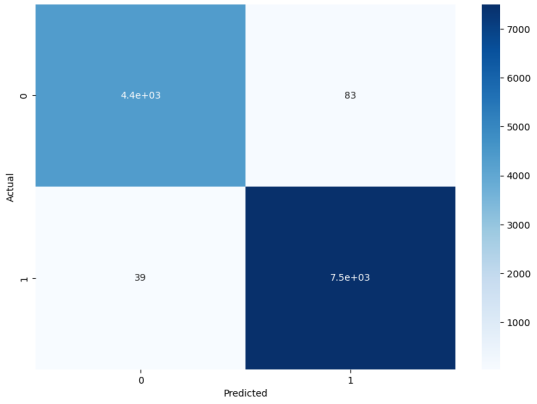

FIGURE 6. Global model Confusion Matrix on FLAME dataset.

D. EVALUATION OF THE PROPOSED APPROACH

1) PUBLIC DATASETS

We evaluated and tested our proposed strategy using the Mnist [75] and CIFAR-10 [76] datasets. The Mnist dataset is a well-known dataset that is used in machine learning and computer vision tasks. "Mnist" refers to the Modification National Institute of Standards and Technology dataset. It contains images of handwritten digits from 0 to 9 along with their respective labels. The images are all grayscale and measure 28 pixels by 28 pixels. In order to evaluate and test ML algorithms, specifically image classification algorithms, the Mnist dataset is most commonly used.

As another popular dataset for machine learning and computer vision applications, the CIFAR-10 is also a popular choice. This is an acronym for "Canadian Institute for Advanced Research". As part of the CIFAR-10 dataset, there are 60,000 RGB images that are 32 by 32 in size, presenting ten different classes of classification. The classes represent different categories such as cats, birds, cars, etc.

Additionally, the CIFAR-10 dataset is widely used as a benchmark for evaluating machine learning models, particularly deep neural networks. It has a greater diversity of classes than Mnist, making it a more complex and challenging dataset.

2) METRICS WITH GLOBAL ROUNDS

We conducted the first experiment to evaluate the suggested strategy's performance in terms of accuracy, loss precision, recall, and f1-score. We used local model No 1 and trained it on the MNIST dataset.

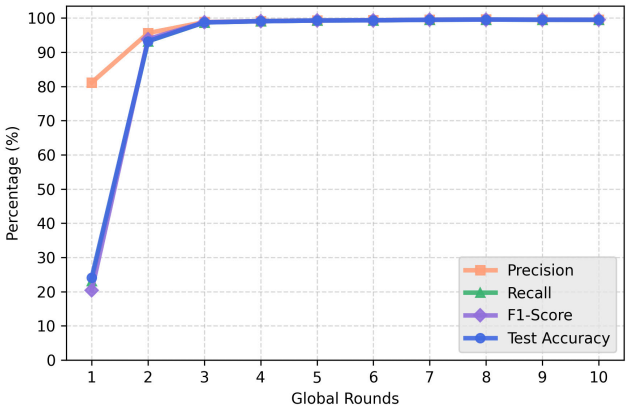

FIGURE 7. Metrics with Global Rounds.

Throughout all our experiments, we assumed that the dataset had been collected by UAVs and was prepared for local model training at edge computers. In the simulation, we use 10 edge computers, 20 local epochs, and 10 global rounds. MNIST data samples were evenly divided into a number of participants, and each edge computer built its local model by considering Model No 4 as the local model. Adam was used as an optimizer for each local model, and the categorical cross-entropy loss function was used as a loss function for each local model. The sample images were normalized by dividing them by 255 during the training process for each local model. Moreover, as a pre-processing step, we applied a hot encoding scheme to the labels of the images due to the use of categorical cross-entropy as the loss function. The results of this experiment are presented in figure 7 as accuracy, precision, recall, and f1-score of the global model, and its loss with respect to global rounds is illustrated in figure 8 on testing images of MNIST dataset. It illustrates the test loss of the global model in terms of the number of global rounds. On the MNIST test dataset, figure 8 indicates that the proposed framework consistently decreases the loss of global models as the number of global training rounds increases. Considering these results, it is evident that the proposed ARS approach is effective.

It is estimated that global model accuracy, precision, recall, and f1-score were 24.12 percent, 81.12%, 23.11%, and 20.37% respectively, after the first round. After the second round, the global model accuracy, precision, recall, and f1-score were 90%+. In less than five rounds, we were able to achieve 99.53% accuracy, 99.54% precision, 99.52% recall, and 99.53% f1-score in all considered metrics of the global model. In the first round, the global model had a test loss of 2.98 and it decreased to 0.03 in less than five rounds and it ended up with a minimum loss of 0.02.

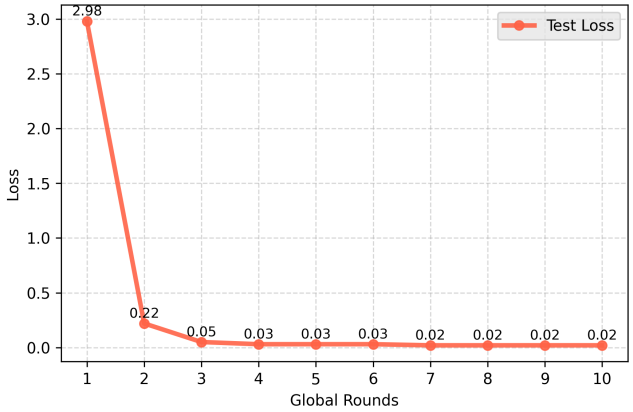

FIGURE 8. Test Loss vs Global Rounds.

3) METRICS VS NUMBER OF EDGE COMPUTERS

This section aims to examine the scalability of the proposed work and its impact on the evaluation metrics. We conducted ten experiments to test the scalability of the global model convergence process by taking into account different numbers of edge computers. To conduct these experiments, Model No 3 and the MNIST dataset were used. A total of 10 rounds and 20 local epochs are carried out in each experiment. In each experiment, samples from the MNIST dataset were equally divided between participants.

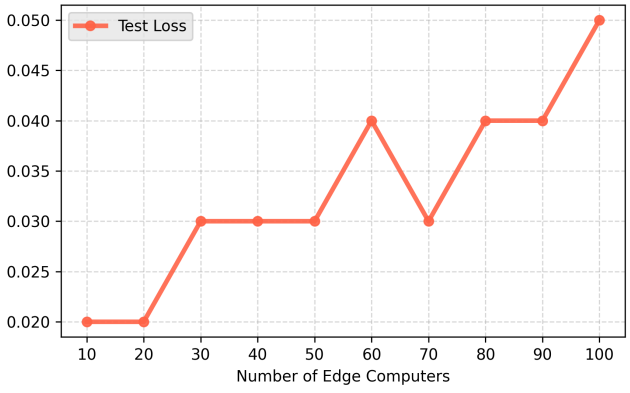

FIGURE 9. Loss vs Number of Edge Computers.

The pre-processing steps of normalization and one hot encoding scheme are similar to those used in the previous experiment. Edge computers trained local models with 20 epochs, 128 batch sizes, Adam as an optimizer, and categorical cross-entropy as a loss function. The loss of the global model on MNIST testing images is illustrated in figure 9. All ten experiments were conducted using 10 edge computers in the first experiment, 20 in the second experiment, 30 in the third experiment, and 100 in the tenth experiment. Apart from changing the number of edge computers from 10 to 100, the configuration of these experiments remained the same. We were able to achieve 99% accuracy, precision, recall, and f1-score of the global model in all these experiments on the testing dataset, and 0.02 to 0.05 loss of global model testing in all these experiments. Our proposed strategy is effective in scalability with constant accuracy, precision, recall, and f1 score. As a result of the experiments, when the number of edge computers was increased, it took more rounds to achieve the same accuracy that was achieved with fewer rounds with fewer edge computers. Because of this, the number of rounds was increased from 10 to 15 for 90 and 100 edge computers, respectively. In 12 rounds, it gives 99%+ accuracy, precision, recall, and f1-score for 90 and 100-edge computers. We obtained the same results with fewer than 5 rounds for fewer edge computers. This is because if we increase the number of edge computers, we will have fewer dataset samples for each one, resulting in poor local model training.

According to our findings, our proposed strategy is scalable and capable of accommodating many edge computers while maintaining good accuracy. Furthermore, we conclude that the number of rounds must be increased if we increase the number of edge computers without increasing the number of samples in the local dataset.

4) ACCURACY VS NUMBER OF GLOBAL ROUNDS

This experiment is designed to determine the effect of a number of rounds on test accuracy. This experiment uses model No 3 as the local model and MNIST as the local dataset. This experiment includes 50 global rounds, 10 edge computers, and 20 local epochs. Furthermore, each local model training was performed with 128 batch sizes, Adam as an optimizer, and categorical cross entropy as a loss function. The pre-processing steps are the same as those we employed in our previous experiments.

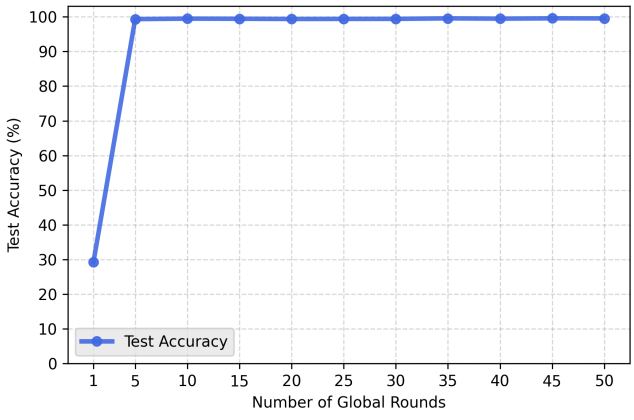

FIGURE 10. Accuracy Vs Number of Global Rounds.

The results of this experiment are demonstrated in 10 and in 11 of the global model on the MNIST test dataset. In the first round, the accuracy and loss were 29.26% and 2.18 respectively. after the 5th round, the test accuracy and loss of the global model were 99.33% and 0.02 respectively. Even though the number of rounds was increased, accuracy

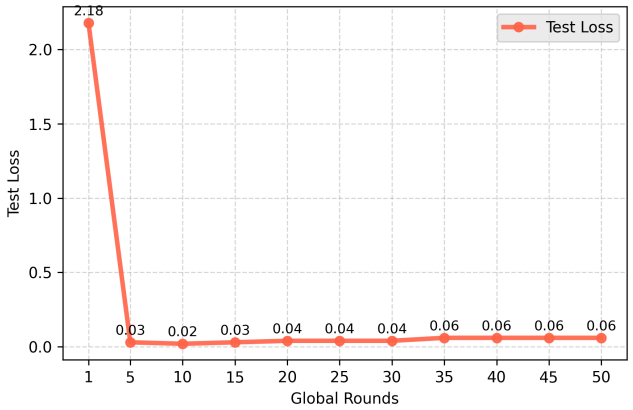

FIGURE 11. Loss vs Number of Global Rounds.

and loss did not improve much, but, at the 45th round, we obtained the maximum accuracy of 99.56%. The further increase in the number of rounds increases the loss, which increases from 0.02 to 0.05. As a result, we conclude that the number of rounds should be fine-tuned to achieve minimal loss for the convergence of the global model.

5) DIFFERENTIALLY PRIVATE LOCAL MODEL SHARING

As a way to tackle privacy leakage while sharing local model weights by integrating Differential Privacy into local model training. this experiment uses 30 rounds, 20 edge computers, and 50 local epochs. Differentially Private Stochastic gradient descent (DPSGD) was used as an optimizer for the local model. The other parameters of the local model have a batch size of 250, a categorical cross-entropy as a loss function, and a learning rate of 0.25. Similarly to the previous experiments, this dataset was pre-processed.

Our noise integration was based on Gaussian noise. It is a form of white noise frequently introduced to data or computations to mask original values and ensure privacy preservation [72]. In our approach, we introduced Gaussian random noise to a local model of a convolutional neural network (CNN) during the optimization phase. This was achieved using the DPKerasSGDOptimizer from the TensorFlow-Privacy library.

The integration of Gaussian noise occurs in the gradients before updating the local model parameters. The extent of noise to be added is determined by the noise multiplier parameter. This parameter is bounded by a lower zero limit and lacks a strict upper boundary. In practice, it is often set to a relatively small value to prevent excessive noise, as greater noise would lead to reduced model accuracy. It is a trade-off between privacy and accuracy. How much noise should be added depends on the accuracy threshold limit and the problem or situation's requirements. The amount of noise to be added is defined by the parameter named noise multiplier. We used a 0.3 value as a noise multiplier and achieved 98.0%+ accuracy, precision, recall, and f1-score, which is still better than the results of the related work presented in the table No 2. We examine less than 2% drop in global model accuracy after integrating Differential Privacy, but this is not a significant loss in the context of achieving privacy and masking the impact of individual records on the overall results.

6) ANALYSIS OF TIME CONSUMPTION

This section aims to analyze the time consumption associated with the proposed strategy. It is also essential to consider the impact of increasing the number of edge computers and global rounds on time consumption. In these experiments, edge computers were increased from 10 to 100, and global rounds were increased from 1 to 10. Time module in Python is used to monitor the computation time. To train the local model, we considered Model No. 1 presented in figure 3 and used MNIST as the local dataset. As in the previous experiments, the same pre-processing steps are followed, and the MNIST dataset is divided equally among all edge computers. This experiment's results are shown in figure 12.

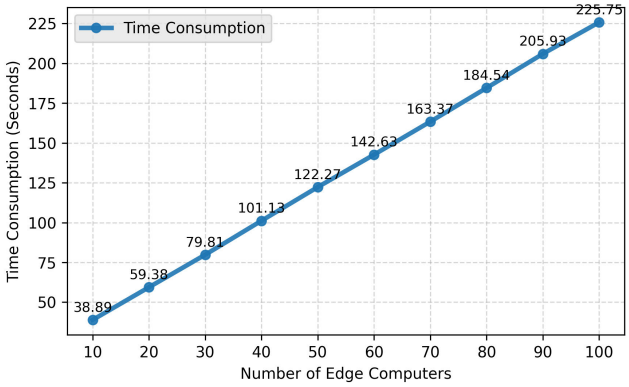

FIGURE 12. Time Consumption and Number of Edge Computers.

The training time of the local model was not included in the analysis, and only the proposed strategy was considered. The reason for not including the local model training time is that our proposed work does not rely on any specific local model, and the local model training time is not constant and depends on the size of the local dataset and the architecture of the local model. The time required for 10-edge computers was 38.89 seconds, 59.38 seconds for 20-edge computers, and 225.81 seconds for 100-edge computers. Taking this analysis into account, we found that increasing the number of global rounds did not affect the time taken by the proposed strategy, but increasing the number of edge computers linearly increased the time as shown in figure 12.

7) COMPARATIVE ANALYSIS

This section aims to analyze the proposed work in light of state-of-the-art research. We performed two experiments on two different datasets by using two different local models.

We consider the local model 3 and the MNIST dataset in the first experiment. This experiment follows the same pre-processing steps as the previous one. Simulated edge

TABLE 2. Comparative Analysis on MNIST dataset.

Ref	Y[1]	LM[2]	L[3]	A[4]%
A. McGibney et al. [50]	2023	FCN[5]	-	72.0
Kalapaaking et al. [48]	2023	LeNet	-	95.7
Li et al. [55]	2022	CNN[6]	-	95.0
Tursunboev et al. [58]	2022	CNN[6]	-	98.9
Ridhawi et al. [77]	2022	CNN[6]	0.12	96.2
H. Zhang et al. [57]	2021	CNN[6]	-	95.0
H. Yan et al. [78]	2020	CNN[6]	-	94.9
Proposed	2023	CNN[6]	0.02	99.53

[1] Year
[2] Local Model
[3] Loss
[4] Accuracy
[5] Fully Connected Network
[6] Convolutional Neural Network

TABLE 3. Comparative Analysis on CIFAR-10 dataset.

Ref	Y[1]	LM[2]	L[3]	A[4]%
Liu et al. [79]	2023	MobileNet	-	82.0
Kalapaaking et al. [48]	2023	AlexNet	-	73.3
Ridhawi et al. [77]	2022	CNN[6]	2.04	73.8
Proposed	2023	ResNet-50 + FCN[8]	0.24	92.74

[1] Year
[2] Local Model
[3] Loss
[4] Accuracy
[5] Fully Connected Network
[6] Convolutional Neural Network

computers were given equal shares of the training samples of the MNIST dataset. During this experiment, 10 rounds were conducted, 20 local epochs were created, and 20 edge computers were simulated. The local model 3 was trained with 128 batch size, Adam optimizer, and categorical cross-entropy as loss function.

This experiment's results and comparison to recent work are presented in table 2 regarding global model accuracy, precision, recall, f1-score, and loss for the test images of MNIST. We achieved better results than the recent work, which focused only on accuracy and did not consider other important evaluation metrics. As a result of this experiment, the highest accuracy, precision, recall, and f1-score were 99.53%, 99.54%, 99.52%, and 99.53% respectively, and achieved the lowest global model loss with the value of 0.02. The Confusion Matrix of the global model on MNIST test samples is presented in figure 13.

As part of the second experiment, the local model 4 and the dataset for CIFAR-10 were considered. The simulation consisted of three rounds, five local epochs, and ten edge computers. The CIFAR-10 training samples were evenly divided and given to the edge computers for local model training. For the training of the local model, 64 batch sizes were used, an SGD optimizer was applied, and the

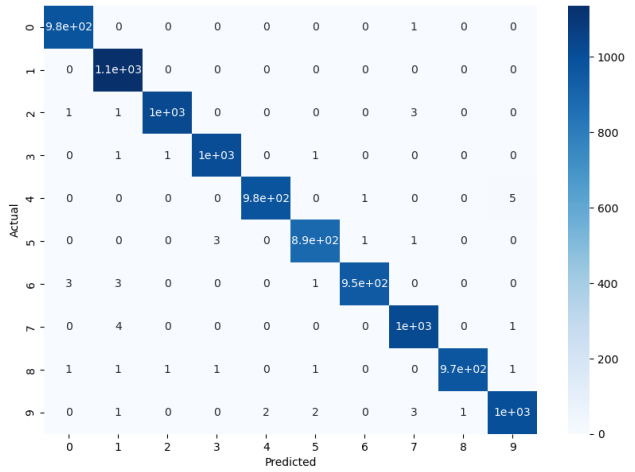

FIGURE 13. Confusion Matrix of the global model on MNIST Dataset.

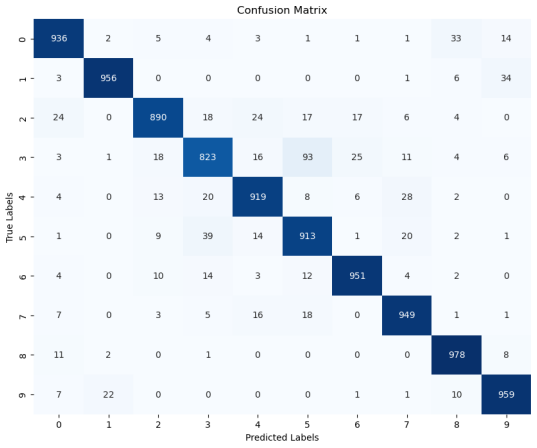

FIGURE 14. Confusion Matrix of the global model on CIFAR-10 Dataset.

loss was computed using sparse categorical cross-entropy. Comparisons between the study results and those of recent related work are presented in table 3. We achieved more than 90 percent accuracy, precision, recall, and F1-score in just two global rounds, with a global model loss of less than 0.28. After the third global round, the accuracy, precision, recall, and f1-score were 92.74%, 92.76%, 92.74%, and 92.71% respectively, and obtained a 0.24 loss. The confusion matrix of these results is illustrated in figure 14. The results are clearly superior to those of recent state-of-the-art studies.

VI. CONCLUSION

This framework stands out because of its unique integration of differential privacy, the InterPlanetary File System (IPFS), and Solidity smart contract, all orchestrated by an Accumulative Reputation Score (ARS). Several challenges encountered in UAV-based FL for smart cities are addressed by the proposed framework, including single-point failure, privacy leakage, scalability, and global model verification. By utilizing differential privacy, participants' local data is protected, while the decentralized nature of the framework minimizes the risk of single-point failure.

The proposed framework was carefully assessed, and we achieved better results than the existing work. The highest accuracy, precision, recall, f1-score, and loss achieved by the proposed framework were 99.53%, 99.54%, 99.52%, 99.53%, and 0.02 on the MNIST dataset, and 92.74%, 92.75%, 92.74%, 92.71%, and 0.24 on CIFAR-10 dataset respectively. The results showed that by exploiting the distributed nature of federated learning and the effectiveness of IPFS for off-chain model storage, the proposed strategy outperforms compared to the existing work.

In the future, we intend to propose a secure authentication mechanism to enhance security, a participant selection mechanism to Improve fairness and incentivize participation, a model compression mechanism to Reduce communication overhead, and a fair incentive mechanism to Encourage participation and improve global model performance for the framework are under consideration.

ACKNOWLEDGMENT

The authors sincerely thank the ICESCO Chair for Big Data Analytics and Edge Computing, Quaid-i-Azam University, Islamabad, Pakistan, and Shaheed Zulfikar Ali Bhutto Institute of Science and Technology University, Islamabad.

REFERENCES

[1] C. Feng, K. Yu, M. Aloqaily, M. Alazab, Z. Lv, and S. Mumtaz, "Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13784–13795, Nov. 2020.

[2] K. Yu, Z. Guo, Y. Shen, W. Wang, J. C. Lin, and T. Sato, "Secure artificial intelligence of things for implicit group recommendations," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2698–2707, Feb. 2022.

[3] H. Li, K. Yu, B. Liu, C. Feng, Z. Qin, and G. Srivastava, "An efficient ciphertext-policy weighted attribute-based encryption for the Internet of Health Things," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 5, pp. 1949–1960, May 2022.

[4] Y. Sun, J. Liu, K. Yu, M. Alazab, and K. Lin, "PMRSS: Privacy-preserving medical record searching scheme for intelligent diagnosis in IoT healthcare," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 1981–1990, Mar. 2022.

[5] P. Satia, "Drones: A history from the British middle east," *Humanity, Int. J. Human Rights, Humanitarianism, Develop.*, vol. 5, no. 1, pp. 1–31, Mar. 2014.

[6] D. Saraswat, A. Verma, P. Bhattacharya, S. Tanwar, G. Sharma, P. N. Bokoro, and R. Sharma, "Blockchain-based federated learning in UAVs beyond 5G networks: A solution taxonomy and future directions," *IEEE Access*, vol. 10, pp. 33154–33182, 2022.

[7] C. Kumar and S. Mohanty, "Current trends in cyber security for drones," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2021, pp. 1–5.

[8] P. Kurien. *Five Industries Benefiting From Drone Inspections*. IBM Blog. Accessed: Aug. 19, 2023. [Online]. Available: https://www.ibm.com/blog/five-industries-benefiting-from-drone-inspections/

[9] D. Schwartz. *Why Drones Are the Future of Outdoor Search and Rescue*. Accessed: Aug. 19, 2023. [Online]. Available: https://www.outsideonline.com/outdoor-adventure/exploration-survival/drones-search-rescue/

[10] *Zipline Instant Delivery and Logistics*. Accessed: Aug. 19, 2023. [Online]. Available: https://www.flyzipline.com/

[11] *Amazon Prime Air Prepares for Drone Deliveries*. Accessed: Aug. 19, 2023. [Online]. Available: https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries

[12] *Joby Aviation Welcomes New 75M Investment From Uber as it Acquires Uber Elevate and Expands Partnership*. Accessed: Aug. 19, 2023. [Online]. Available: https://www.jobyaviation.com/news/joby-aviation-welcomes-new-75m-investment-from-uber-as-it-acquires-uber-elevate-and-expands-partnership/?uclick_id=482470a4-7505-471c-b00f-ae98fa48256f

[13] *Using Unmanned Aircraft System Tech*. Accessed: Aug. 19, 2023. [Online]. Available: https://www.bechtel.com/newsroom/press-releases/bechtel-among-first-companies-to-use-unmanned-aircraft-system-technology-in-construction/

[14] P. Stone, R. Brooks, E. Brynjolfsson, R. Calo, O. Etzioni, G. Hager, J. Hirschberg, S. Kalyanakrishnan, E. Kamar, S. Kraus, K. Leyton-Brown, D. Parkes, W. Press, A. Saxenian, J. Shah, M. Tambe, and A. Teller, "Artificial intelligence and life in 2030: The one hundred year study on artificial intelligence," 2022, *arXiv:2211.06318*.

[15] Mainblades. *EasyJet Makes Drone Inspection a Reality in Aviation MRO*. Accessed: Aug. 19, 2023. [Online]. Available: https://mainblades.com/article/easyjet-makes-drone-inspection-a-reality-in-aviation-mro/

[16] CB Insights Res. *21 Companies Using Drone Tech Today From Retail to Insurance*. Accessed: Aug. 19, 2023. [Online]. Available: https://www.cbinsights.com/research/report/corporations-drone-technology/

[17] *Drone COVID Vaccine Deliveries*. Accessed: Aug. 19, 2023. [Online]. Available: https://about.ups.com/us/en/our-stories/innovation-driven/drone-covid-vaccine-deliveries.html

[18] W. Y. B. Lim, J. Huang, Z. Xiong, J. Kang, D. Niyato, X.-S. Hua, C. Leung, and C. Miao, "Towards federated learning in UAV-enabled Internet of Vehicles: A multi-dimensional contract-matching approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5140–5154, Aug. 2021.

[19] Y. Wang, Z. Su, N. Zhang, and A. Benslimane, "Learning in the air: Secure federated learning for UAV-assisted crowdsensing," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1055–1069, Apr. 2021.

[20] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[21] F. A. Ghaleb, M. A. Maarof, A. Zainal, B. A. S. Al-rimy, A. Alsaeedi, and W. Boulila, "Ensemble-based hybrid context-aware misbehavior detection model for vehicular ad hoc network," *Remote Sens.*, vol. 11, no. 23, p. 2852, Dec. 2019.

[22] M. Driss, D. Hasan, W. Boulila, and J. Ahmad, "Microservices in IoT security: Current solutions, research challenges, and future directions," *Proc. Comput. Sci.*, vol. 192, pp. 2385–2395, Jan. 2021.

[23] M. A. Khan, M. A. Khan Khattk, S. Latif, A. A. Shah, M. Ur Rehman, W. Boulila, M. Driss, and J. Ahmad, "Voting classifier-based intrusion detection for IoT networks," in *Proc. Adv. Smart Soft Comput. (ICACIn)*. Cham, Switzerland: Springer, 2022, pp. 313–328.

[24] C. Feng, K. Yu, A. K. Bashir, Y. D. Al-Otaibi, Y. Lu, S. Chen, and D. Zhang, "Efficient and secure data sharing for 5G flying drones: A blockchain-enabled approach," *IEEE Netw.*, vol. 35, no. 1, pp. 130–137, Jan. 2021.

[25] Y. Qu, S. R. Pokhrel, S. Garg, L. Gao, and Y. Xiang, "A blockchained federated learning framework for cognitive computing in industry 4.0 networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2964–2973, Apr. 2021.

[26] Y. Zheng, H. Duan, X. Yuan, and C. Wang, "Privacy-aware and efficient mobile crowdsensing with truth discovery," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 1, pp. 121–133, Jan. 2020.

[27] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 1387–1395.

[28] B. Brik, A. Ksentini, and M. Bouaziz, "Federated learning for UAVs-enabled wireless networks: Use cases, challenges, and open problems," *IEEE Access*, vol. 8, pp. 53841–53849, 2020.

[29] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, sS. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," 2018, *arXiv:1811.03604*.

[30] W. Zhang, Q. Lu, Q. Yu, Z. Li, Y. Liu, S. K. Lo, S. Chen, X. Xu, and L. Zhu, "Blockchain-based federated learning for device failure detection in industrial IoT," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5926–5937, Apr. 2021.

[31] M. Chen, O. Semiari, W. Saad, X. Liu, and C. Yin, "Federated deep learning for immersive virtual reality over wireless networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[32] Y. Zhang, T. Gu, and X. Zhang, "Poster abstract: A ChainSGD-reduce approach to mobile deep learning for personal mobile sensing," in *Proc. 19th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2020, pp. 325–326.

[33] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 691–706.

[34] H. Li, D. Liu, Y. Dai, T. H. Luan, and S. Yu, "Personalized search over encrypted data with efficient and secure updates in mobile clouds," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 1, pp. 97–109, Jan. 2018.

[35] S. H. Alsamhi, F. A. Almalki, F. Afghah, A. Hawbani, A. V. Shvetsov, B. Lee, and H. Song, "Drones' edge intelligence over smart environments in B5G: Blockchain and federated learning synergy," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 295–312, Mar. 2022.

[36] C. Zhu, X. Zhu, J. Ren, and T. Qin, "Blockchain-enabled federated learning for UAV edge computing network: Issues and solutions," *IEEE Access*, vol. 10, pp. 56591–56610, 2022.

[37] D. A. Sungheetha and D. R. Sharma R, "Real time monitoring and fire detection using Internet of Things and cloud based drones," *J. Soft Comput. Paradigm*, vol. 2, no. 3, pp. 168–174, Jul. 2020.

[38] S. Khanna and C. R. Murthy, "Communication-efficient decentralized sparse Bayesian learning of joint sparse signals," *IEEE Trans. Signal Inf. Process. over Netw.*, vol. 3, no. 3, pp. 617–630, Sep. 2017.

[39] D. Li, Z. Luo, and B. Cao, "Blockchain-based federated learning methodologies in smart environments," *Cluster Comput.*, vol. 25, no. 4, pp. 2585–2599, Aug. 2022.

[40] Y. Liu, J. Nie, X. Li, S. H. Ahmed, W. Y. B. Lim, and C. Miao, "Federated learning in the sky: Aerial-ground air quality sensing framework with UAV swarms," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9827–9837, Jun. 2021.

[41] L. Yin, J. Feng, H. Xun, Z. Sun, and X. Cheng, "A privacy-preserving federated learning for multiparty data sharing in social IoTs," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2706–2718, Jul. 2021.

[42] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[43] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K.-Y. Lam, "Local differential privacy-based federated learning for Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8836–8853, Jun. 2021.

[44] W. Boulila, M. K. Khlifi, A. Ammar, A. Koubaa, B. Benjdira, and I. R. Farah, "A hybrid privacy-preserving deep learning approach for object classification in very high-resolution satellite images," *Remote Sens.*, vol. 14, no. 18, p. 4631, Sep. 2022.

[45] B. Zhao, K. Fan, K. Yang, Z. Wang, H. Li, and Y. Yang, "Anonymous and privacy-preserving federated learning with industrial big data," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6314–6323, Sep. 2021.

[46] S. A. H. Mohsan, M. A. Khan, F. Noor, I. Ullah, and M. H. Alsharif, "Towards the unmanned aerial vehicles (UAVs): A comprehensive review," *Drones*, vol. 6, no. 6, p. 147, Jun. 2022.

[47] D. C. Nguyen, M. Ding, Q.-V. Pham, P. N. Pathirana, L. B. Le, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12806–12825, Aug. 2021.

[48] A. P. Kalapaaking, I. Khalil, M. S. Rahman, M. Atiquzzaman, X. Yi, and M. Almashor, "Blockchain-based federated learning with secure aggregation in trusted execution environment for Internet-of-Things," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1703–1714, Feb. 2023.

[49] X. Liu, Y. Deng, and T. Mahmoodi, "Wireless distributed learning: A new hybrid split and federated learning approach," *IEEE Trans. Wireless Commun.*, vol. 22, no. 4, pp. 2650–2665, Apr. 2023.

[50] T. Ranathunga, A. McGibney, S. Rea, and S. Bharti, "Blockchain-based decentralized model aggregation for cross-silo federated learning in Industry 4.0," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 4449–4461, Mar. 2023.

[51] I. Sharma, A. Sharma, and S. K. Gupta, "Asynchronous and synchronous federated learning-based UAVs," in *Proc. 3rd Int. Symp. Instrum., Control, Artif. Intell., Robot. (ICA-SYMP)*, Jan. 2023, pp. 105–109.

[52] C. Feng, B. Liu, K. Yu, S. K. Goudos, and S. Wan, "Blockchain-empowered decentralized horizontal federated learning for 5G-enabled UAVs," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3582–3592, May 2022.

[53] Y. Wang, Z. Su, T. H. Luan, R. Li, and K. Zhang, "Federated learning with fair incentives and robust aggregation for UAV-aided crowdsensing," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3179–3196, Sep. 2022.

[54] I. Abunadi, M. M. Althobaiti, F. N. Al-Wesabi, A. M. Hilal, M. Medani, M. A. Hamza, M. Rizwanullah, and A. S. Zamani, "Federated learning with blockchain assisted image classification for clustered UAV networks," *Comput., Mater. Continua*, vol. 72, no. 1, pp. 1195–1212, 2022.

[55] H. Li, H. Zhang, D. Liao, X. Zhu, Y. Dai, and S. Verma, "A data sharing method for Internet of Drones based on federated learning," in *Proc. 5th Int. ACM Mobicom Workshop Drone Assist. Wireless Commun. 5G Beyond*, Oct. 2022, pp. 91–96.

[56] Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Z. Li, L. Lyu, and Y. Liu, "Privacy-preserving blockchain-based federated learning for IoT devices," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1817–1829, Feb. 2021.

[57] H. Zhang and L. Hanzo, "Federated learning assisted multi-UAV networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 14104–14109, Nov. 2020.

[58] 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," *Appl. Sci.*, vol. 12, no. 2, p. 670, Jan. 2022.

[59] W. Chen, B. Liu, H. Huang, S. Guo, and Z. Zheng, "When UAV swarm meets edge-cloud computing: The QoS perspective," *IEEE Netw.*, vol. 33, no. 2, pp. 36–43, Mar. 2019.

[60] Q.-V. Pham, M. Le, T. Huynh-The, Z. Han, and W.-J. Hwang, "Energy-efficient federated learning over UAV-enabled wireless powered communications," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4977–4990, May 2022.

[61] Y. Yu, X. Bu, K. Yang, H. Yang, X. Gao, and Z. Han, "UAV-aided low latency multi-access edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 5, pp. 4955–4967, May 2021.

[62] L. Zhang and N. Ansari, "Optimizing the operation cost for UAV-aided mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 6085–6093, Jun. 2021.

[63] B. Yang, H. Shi, and X. Xia, "Federated imitation learning for UAV swarm coordination in urban traffic monitoring," *IEEE Trans. Ind. Informat.*, vol. 19, no. 4, pp. 6037–6046, Apr. 2023.

[64] S. Huang, J. Gui, T. Wang, and X. Li, "Joint mobile vehicle-UAV scheme for secure data collection in a smart city," *Ann. Telecommun.*, vol. 76, nos. 9–10, pp. 559–580, Oct. 2021.

[65] X. Wu, Y. Zhang, M. Shi, P. Li, R. Li, and N. N. Xiong, "An adaptive federated learning scheme with differential privacy preserving," *Future Gener. Comput. Syst.*, vol. 127, pp. 362–372, Feb. 2022.

[66] W. Y. B. Lim, S. Garg, Z. Xiong, Y. Zhang, D. Niyato, C. Leung, and C. Miao, "UAV-assisted communication efficient federated learning in the era of the artificial intelligence of things," *IEEE Netw.*, vol. 35, no. 5, pp. 188–195, Sep. 2021.

[67] D. Wilkinson and M. Ooijevaar, "Data protection, privacy & big data 1," in *Drone Law and Policy*. Evanston, IL, USA: Routledge, 2021, pp. 183–212.

[68] R. de Caria, "Blockchain and smart contracts: Legal issues and regulatory responses between public and private economic law," *Italian LJ*, vol. 6, p. 363, 2020.

[69] M. Alamri, N. Jhanjhi, and M. Humayun, "Blockchain for Internet of Things (IoT) research issues challenges & future directions: A review," *Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 1, pp. 244–258, 2019.

[70] H. F. Atlam and G. B. Wills, "Technical aspects of blockchain and IoT," in *Advances in Computers*. Amsterdam, The Netherlands: Elsevier, 2019, pp. 1–39.

[71] S. Bragagnolo, M. Marra, G. Polito, and E. G. Boix, "Towards scalable blockchain analysis," in *Proc. IEEE/ACM 2nd Int. Workshop Emerg. Trends Softw. Eng. Blockchain (WETSEB)*, May 2019, pp. 1–7.

[72] Z. Bu, J. Dong, Q. Long, and S. Weijie, "Deep learning with Gaussian differential privacy," *Harvard Data Sci. Rev.*, vol. 2, no. 3, pp. 1–31, Jul. 2020.

[73] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. Hei Li, T. Parcollet, P. P. B. de Gusmão, and N. D. Lane, "Flower: A friendly federated learning research framework," 2020, *arXiv:2007.14390*.

[74] A. Shamsoshoara, F. Afghah, A. Razi, L. Zheng, P. Fulé, and E. Blasch, Apr. 16, 2021, "The FLAME dataset: Aerial imagery pile burn detection using drones (UAVs)," *IEEEDataPort*, doi: 10.21227/qad6-r683.

[75] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[76] A. Krizhevsky, V. Nair, and G. Hinton. (2010). *CIFAR-10 (Canadian Institute for Advanced Research)*. [Online]. Available: http://www.cs.toronto.edu/kriz/cifar.html

[77] I. A. Ridhawi, M. Aloqaily, A. Abbas, and F. Karray, "An intelligent blockchain-assisted cooperative framework for Industry 4.0 service management," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 4, pp. 3858–3871, Dec. 2022.

[78] H. Yang, J. Zhao, Z. Xiong, K.-Y. Lam, S. Sun, and L. Xiao, "Privacy-preserving federated learning for UAV-enabled networks: Learning-based joint scheduling and resource management," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3144–3159, Oct. 2021.

[79] T. Liu, T. Zhang, J. Loo, and Y. Wang, "Deep reinforcement learning-based resource allocation for UAV-enabled federated edge learning," *J. Commun. Inf. Netw.*, vol. 8, no. 1, pp. 1–12, Mar. 2023.

[80] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012.