Computer Networks 203 (2022) 108676

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A blockchain-based Fog-oriented lightweight framework for smart public vehicular transportation systems

Thar Baker [a,*], Muhammad Asim [b], Hezekiah Samwini [c], Nauman Shamim [b], Mohammed M. Alani [d], Rajkumar Buyya [e]

[a] Department of Computer Science, College of Computing and Informatics, University of Sharjah, United Arab Emirates
[b] National University of Computer and Emerging Sciences, Islamabad, Pakistan
[c] Edge Hill University, UK
[d] Seneca College, Toronto, Canada
[e] CLOUDS Lab, School of Computing and Information Systems, The University of Melbourne, Australia

ARTICLE INFO

Keywords:
Blockchain
Vehicular networks
Fog
Federated learning
Vehicular clouds
Beyond 5G technologies

ABSTRACT

Rapid urbanization is putting a strain on the transport systems of cities worldwide. The effects of this trend include prolonged traffic jams and increasing environmental pollution from rising CO_2 emissions. As city planning requires innovative ways of dealing with the rapid urbanization trend, technological solutions were proposed such as cloud computing, smart vehicles, and Vehicular Ad hoc NETwork (VANET). In this paper, we take advantage of next-generation network technologies to propose a responsive and lightweight framework for smart transportation system which employs blockchain for authentication using fog computing's improvement over cloud computing for distributed applications to provide an efficient and secure transportation system. We take into account the future technologies of 5G and Beyond 5G (B5G) and argue that the integration of B5G technologies, federated learning, blockchain, and edge computing provides the perfect platform necessary for a smart transportation system The evaluation of the proposed framework is done by comparing it to the current cloud-based approach in iFogSim, a popular simulation tool for fog computing research. The evaluation of blockchain-based authentication was done using a customized implementation of blockchain executed in an experimental setup. The simulation results showed that the proposed framework provides superior performance in terms of security, latency, and energy consumption of the system.

1. Introduction

As the world population grows to 7.78 billion human beings [1], urban populations continue to rise rapidly and transportation in urban areas becomes more and more challenging. The United Nations Population Fund reports that more than half of the world's population now live in cities and towns [2]. The figure is expected to rise as more people migrate to urban areas. This rapid urbanization has a great impact on public transportation systems. Challenges such as excessive traffic congestion, lack of parking spaces, longer travel times and environmental pollution from CO_2 emissions are attributed to urban transportation [3]. The BBC reported that in 2017 UK drivers wasted an annual average of 31 h in rush-hour traffic [4]. London was also reported as having the second-worst traffics jam in Europe (after Moscow). Transportation related issues costed the European Union (EU) an estimated 4% of GDP in 2011 [5]. Traditional approaches to solve road traffic challenges, such as expansion of roads and construction of new lanes, are expensive and less desirable as they are usually outpaced by the rate of urbanization. Rapid urbanization demands innovative approaches to solve transportation challenges in cities and towns. Three technological developments; Vehicular Ad hoc NETworks (VANETs), Intelligent Transportation Systems (ITS), Fog Computing along with cellular networks advancement in 5G and beyond — hold promise to present an alternate approach to dealing with road traffic challenges.

ITS have been proposed to improve urban transportation. ITS systems integrate information and communication systems with existing transportation infrastructure to provide sustainable and efficient transportation systems. Technologies used in ITS systems include mobile technology, Internet of Things (IoT), cloud computing, Global Positioning System (GPS) technology and connected vehicles. A common example of ITS can be found in Transport Network Companies (TNCs)

* Corresponding author.
E-mail addresses: tshamsa@sharjah.ac.ae (T. Baker), muhammad.asim@nu.edu.pk (M. Asim), samwinih@edgehill.ac.uk (H. Samwini), i191610@nu.edu.pk (N. Shamim), m@alani.me (M.M. Alani), rbuyya@unimelb.edu.au (R. Buyya).

https://doi.org/10.1016/j.comnet.2021.108676
Received 18 March 2021; Received in revised form 6 November 2021; Accepted 30 November 2021
Available online 20 December 2021

such as Uber and Lyft. TNCs use mobile technology and cloud services to connect passengers and drivers for transportation services. These services, however, do not consider all factors impacting the optimum routes for trips. Ensuring the use of the optimum route for each trip would ensure that trips are more efficient, time-wise and economical. This can reduce the contribution of TNCs to urban traffic jams. This model has proven to be more efficient than traditional taxi. Another study in [6] examined traffic congestion in major cities in the United States to analyse the effect that Uber has had on traffic congestion. The researchers concluded that the emergence of Uber has significantly reduced traffic congestion.

In [7], researchers compared the efficiency of UberX drivers to traditional taxi drivers in five cities in the United States – Boston, Los Angeles, New York, San Francisco and Seattle – based on the capacity utilization rate. The research found that Uber drivers were more efficient than traditional taxi drivers. The researchers identified four factors that may account for this; one of which is the driver–passenger matching technology used by Uber. Although this study is done in only five cities in one country, their results are consistent with other studies done elsewhere. Also, TNCs have provided avenue for transportation in situations where traditional taxis have been known to be scarce. One study [8] found that Uber has made it easier to get transportation when it is raining.

Although they provide a good means of transportation, it may be argued that the efficiency of TNCs can be further improved. First, the use of cloud servers increases the delay experienced by the users. As discussed earlier, relying on cloud servers which are geographically remote from the user affects the performance of low-latency applications. Also, the problem of matching passengers to drivers requires location-awareness which is not supported by cloud computing. TNCs use GPS location information and Nearest Vehicle Dispatch algorithms to determine location and match drivers and passengers, but GPS has challenges of availability and accuracy especially in urban areas [9]. Also, Nearest Vehicle Dispatch does not consider how fast it will take a driver to arrive at a given location in real time. A driver may be closer to a passenger but will take longer to reach the passenger's location due to traffic or other conditions on the road.

TNCs use mobile-cloud architectures to provide transportation services to passengers. A passenger makes a request using a mobile or web application. Based on the location of the incoming request, near-by drivers are prompted and asked to accept the request. The passenger is alerted when his/her request is accepted and the driver moves to the location of the passenger for the trip to begin. The system relies on cloud-based applications to process requests and match passengers to drivers. Also, in selecting a route from one point to another, the system relies on the phone's GPS technology to get the coordinates. The route for a trip is chosen from available routes by the driver based on the system's map; there is often no up-to-date information on the road condition such as traffic jams, weather, etc. The shortest route may take the longest time due to the road condition at the time of the trip, including change to the conditions that can happen after the trip begins. Challenges associated with cloud computing such as high-latency and security issues affect the overall system performance. Also, relying solely on GPS for location information/identification presents numerous challenges including unavailability and inaccuracy especially in urban areas with high-rise buildings [9].

Although Cloud Computing has played an important role in the development of ITS and particularly in the operations of TNCs, the remote location of cloud servers poses a challenge for applications that require location-awareness and low-latency. Fog computing is a new distributed computing paradigm that extends processing, communication and storage resources to the edge of the network. It has been proposed as a solution to the inability of traditional cloud computing to support delay-sensitive and location-aware applications. Fog computing sits and serves as a bridge between cloud data centres and end devices to make cloud services and resources available at the edge of the network (closer to end devices) using fog nodes. Fog nodes may be gateways, routers or dedicated devices. Fog computing presents itself with advantages including low-latency access to computation resources, reduction in network traffic/ pressure on traditional cloud and scalability.

5G technologies offer Device to Device (D2D) and Machine to Machine (M2M) communication paradigm, which mostly means industry 4.0 and the Internet of Things [10]. Beyond 5G (B5G) or 6G networks will cater the challenges that arises from 5G networks. It follows the 5G vision, but, in an evolutionary manner. B5G presents a modified view of transmission network in terms of computing networks capable of making decisions by using Artificial Intelligence (AI) and Machine Learning (ML) [11]. The future networks specifically focus on solving digital challenges of both rural and urban worlds, which supposed to be handled by the 5G networks. B5G offers an evolution of 5G technologies with conceptual and technological integration of new technologies that are advantageous for a smart transport system in a number of ways. For example, 5G and B5G visions and research present a more connected, efficient and technological advanced network presence. A smart transportation system would be much feasible in such conditions [12]. Many of the use cases proposed for B5G including detailed discussion about the suitability of B5G for traffic prediction, safety, and security such as vehicle to everything communication (V2X), intelligent transport system [12–14].

A global view of present traffic and road conditions along with predicted situations is what an ideal smart transport system would need. However, developing such a model is a very challenging and complex task. For such predictive models, conventional machine learning is not suitable as it would be very difficult if not impossible to provide such a large training data. Federated Learning (FL) [15] on the other hand allows training global models by training the algorithm on local data sets and later exchanging the meta parameters without sharing the actual dataset. B5G offers better user experience and allows more connected vehicles with better sensory and traffic related data sharing. This combined with FL would allow a better and globally aware prediction model. The work of [16] discusses the application and efficacy of FL and Deep Learning (DL) in B5G networks. Similarly, [17] provides a discussion about the use of FL in many B5G use cases including intelligent transportation system. In addition, the use of blockchain technology can help supporting the development of efficient and secured 5G-enabled applications that rely on FL. The characteristics of blockchain such as unforgeability, privacy, distributed nature, evidence traceability, and transparency make blockchain an excellent solution to various security problems of FL such as, decentralize authentication. Moreover, a number of studies have advocated the use of blockchain for FL to decentralize the machine learning process. The combination of 5G, blockchain and FL offer new business models and diverse vertical applications. These emerging technologies have several desirable advantages for today's needs in terms of security and privacy of data, high quality-of-service, and seamless network connectivity [18–21].

The main purpose of this work is to present a conceptual model of a future smart transportation system that is ubiquitous, transparent, secure, reliable, efficient and easily scalable. We take into account the future technologies offered by 5G and B5G and argue that integration of B5G technologies, FL, blockchain, edge computing, artificial intelligence and Internet of Things (IoT) provide the perfect platform necessary for a smart transportation system.

The main contributions of this paper are as follows:

- We present an integrated Smart Vehicular Transportation System. The system has three layers including a cloud layer for business intelligence analytics and hand-over features for handing over users and rides among fog devices.

- The propose system is a responsive and lightweight framework for TNC systems that is based on the use of fog nodes to process and match user ride requests. Fog devices receive requests directly from users and match them locally to drivers within the same region. This provides for better responsiveness in comparison to a solely cloud-based system.
- We propose a blockchain-based decentralized mechanism to authenticate fog nodes and smart vehicles in order to allow only legitimate entities to communicate with the proposed framework.
- We compare the performance of the user request processing component of the system in a fog-based approach to a cloud-based implementation. The proposed Fog-oriented implementation achieves better performance compared to the cloud based approach.

The rest of the paper is structured as follows: Section 2 presents the related work; Section 3 introduces the research problem being discussed in this paper; Section 3.2 gives an overview of the proposed smart transportation system; Section 4 describes the proposed framework in formal detailed model; Section 5 describes the simulation setup, whereas Section 6 presents the results and critical discussion, and finally Section 7 concludes the paper and discusses some direction for future work.

2. Related work

In this section, four categories of the literature related to the proposed work are discussed to place the paper in a context within the intended research community.

2.1. Public vehicle systems

Ma et al. [22] developed a taxi-sharing system that schedules taxis to pick up passengers based on time, availability of space in the car and monetary considerations. Although this system produced better results compared to other systems, it is a cloud-based centralized system and thus would suffer from the drawbacks of cloud-based transport systems that causes high delay in comparison to distributed models. Another disadvantage is that the system is not traffic-aware. This leads to low efficiency in road use and added congestion. In addition, the proposed system does not takes security into consideration.

In [23] the authors developed a distributed public vehicle scheduling system. Their system is based on a three-layer vehicular network architecture; sensors, fog layer, and cloud layer. Sensors send the collected data about the vehicle to the fog layer for processing, while the cloud layer is where client requests and received and forwarded to the fog layer as well. The system presents the Public Vehicle Path problem, a member of the Dial-a-Ride Problem [24]. The Public Vehicle Path problem attempts to match requests with vehicles at minimal cost to both the service provider and the rider. Public Vehicles receive and fulfil ride requests. The proposed system has a reduced delay, in comparison to [22]. However, due to its lack of traffic awareness, its performance is not comparable to our proposed system that is built around consistent traffic awareness. In addition, [23] proposed system does not employ encryption, which makes it susceptible to eavesdropping and modification attacks.

2.2. Fog computing in ITS

According to [25], ITS was proposed with the aim of using data available from the transportation system to improve the systems and ensure the safety of users and as an efficient way to manage security in transportation. The proposed ITS suffered from high delay in comparison to our proposed system. Mainly because of its full reliance on a centralized cloud system. In addition, it did not focus on security at all. This meant that there were no proper authentication mechanisms, and no encryption.

The authors in [26] proposed an architecture for using fog computing for Big Data Analytics in an ITS. Their design consists of a three-dimensional architecture: intelligent computing dimension, real-time big data analytics dimension and the Internet of Vehicles dimension. The computing dimension of the architecture has 4 layers consisting of 3 fog layers and a cloud layer. The first fog layer consists of end devices which have some computing power and are capable of some data processing. The second fog layer, named the intermediate fog, consists of fog nodes at the edge of the network; in routers, roadside units, base stations etc. The intermediate layer was designed to handle more complex data analysis and management of the first fog layer. The last fog layer consists of small data centres for Intelligent Transport management. The purpose of this layer is to facilitate more complex processing. The cloud layer is responsible for complex AI and Big Data processing with minimum impact on the complex processing required for the analysis of large volumes of data with real-time or near-real-time results. Although this system was successful in achieving reduced delay, in comparison to [25], it missed out on many other important features our proposed system presented. First, it lacked authentication and encryption. Second, it did not present any traffic-awareness, which means that road usage efficiency would be low.

In another work [27], a mechanism for managing traffic congestion in Intelligent Transportation Systems using fog computing was developed. The proposed Fast Offset Xpath (FOX) is a route management system which uses fog nodes attached to Roadside Units to manage congestion in a designated region. Vehicles in the region of a fog node (roadside unit) send information on their speed, position, route etc. to the fog node. The fog uses this information to determine the traffic situation in the region under its control. The fog then re-routes vehicles in the region to control congestion. Information is shared among fog nodes within a certain area to provide a general view of congestion in the area. They tested their system via simulations in OMNett++. The results showed reduction in fuel consumption, travel time and CO2 emissions. Although the proposed system adopted a fog-based distributed architecture, and was traffic-aware, it failed in providing any type of security. The proposed architecture did not include any authentication, encryption, or integrity-preservation mechanisms. This makes it a target for many different attacks such as eavesdropping, modification attacks, man-in-the-middle, and injection attacks.

In [28], a crowd-sensing fog-based system for monitoring the condition of road surfaces is presented. The system consists of 4 components; vehicular sensors for detecting potholes and other events, roadside units which are also fog nodes, cloud servers and a control centre — a trusted server for security services. The system uses an efficient certificateless signcryption method to guarantee privacy and ensure data integrity and confidentiality. The system proposed in this work is like the 3-layer architecture in other proposed fog systems except for the control centre which is included for security.

The concept of Vehicular Fog Computing (VFC) was first introduced in [29]. Unlike Fog Vehicular Computing in which only parked vehicles may become fog nodes, in VFC both moving and parked vehicles may provide compute, storage and networking services. Xiao and Zhu [30] suggested an improvement of the proposed system. To avoid network delays, they proposed using vehicular fog nodes as a wireless access point to reduce the number of hops data travel. A further improvement on VFC was proposed by Huang et al. [31]. They present a three-layer architecture for VFC comprising a cloud layer, fog layer and a data generation layer. In their proposed architecture lower layers pre-process data before transmitting to upper layers. The work in [32] investigates the TCP throughput performance of VFC — comparing three routing protocols AODV, DSR and AOMDV. Proposed applications of VFC include traffic control, road condition monitoring and commercial advertisement.

Liu et al. [33] propose a secure intelligent traffic light control system using fog computing. In their system a fog node is attached to each traffic signal light. Vehicles in a region broadcast information

Table 1
Challenges in intelligent transport systems.

Technology	Challenges
Intelligent transport systems	• Dependence on cloud technologies makes it highly susceptible to delays caused by remote locations of the cloud. • Full reliance on GPS due to lack of cloud location awareness. This can be very challenging in areas with many high-rise buildings or urban areas which can result in high inaccuracies. • Security challenges caused by the use of cloud computing. • Employing Nearest Vehicle Dispatch algorithms to connect drivers to ride-hailers. These algorithms rely on the shortest path and ignore the road conditions. • Contributing immensely to traffic jams due to latencies caused by the cloud implementation, reliance solely on GPS, and lack of current traffic awareness in decision making.
Fog-based ITS	• Full reliance on GPS for location awareness. This can be very challenging in areas with many high-rise buildings or urban areas which can result in high inaccuracies. • Lack of global view of the system. Hence, best route choice could be inaccurate. • Authentication challenges due to the distributed nature of the fog in comparison to the centralization requirement of authentication.

to announce their presence. Fog nodes use the information of the vehicles to determine the traffic load in the region and programme traffic signal lights accordingly. The design assumes that vehicles have enough storage and computation resources to solve a CDH puzzle which is used to ensure the integrity of the system. The goal of their design is to prevent malicious vehicles from providing fake location information to traffic lights/fog nodes and thereby get the traffic light to be programmed to their advantage.

To improve data forwarding in vehicle-to-vehicle communication in VANETs, especially in communication coverage holes (areas where communication is difficult or impossible because there are no roadside units and/or there is low density of connected vehicle), a Software Defined Network (SDN) and fog-based intersection routing scheme was proposed in [34]. Intersection-based routing schemes compute the best route to a destination at each intersection along the way till the destination. The challenge with an intersection-based approach to vehicular routing is that there is no global view of the system, and the best route suggested at a given intersection may have challenges which are not captured by the system.

Ning et al. tested a use-case of VFC — a city-wide traffic control system [35]. In their implementation, fog nodes are deployed to cover local areas of a city. The fog layer receives data from smart vehicles such as travel speed, location and weather conditions. Using the information received from vehicles in the region, fog nodes review and update the timing of traffic signal lights to reflect the traffic load on various roads. Also, each fog node sends aggregated data of its region to the cloud layer where a global view of the traffic in the city is created and general control policies are made. In a similar application of VFC, the authors in [36] designed a Vehicular Fog-oriented traffic and road safety management system. However, in this proposed system, the cloud layer is a server responsible for traffic management which made it reasonably efficient in handling real-time traffic.

Table 1 summarizes the challenges in conventional ITS vs. fog-based ITS.

2.3. 5G and Blockchain

The emerging 5G technology provides new standards in telecommunication and overcomes the challenges of traditional mobile networks by providing seamless network connectivity. It supports new business models and diverse vertical applications with high quality-of-service, increased network capacity and enhanced throughput [19,37,38]. However, 5G systems has special communication and security requirements including decentralization, transparency, secure communication, and evidence traceability. The use of blockchain technology can help support the development of efficient and secured 5G-enabled applications. Blockchain has been suggested by many researchers to address several communication and security related issues in 5G such as crowdsourcing system for 5G-enabled smart cities [18,19,39,40].

Blockchain is a distributed ledger technology that relies on peer-to-peer networks to maintain a permanent, tamper-proof, and traceable transactional data. Every blockchain node keeps and maintain a copy of a blockchain ledger. This ledger is regularly updated on the validation of new transactions [41]. Initially, the blockchain was proposed as a cryptocurrency technology by Satoshi Nakamoto called Bitcoin. Later, it was thought to be quite suitable for the cybersecurity ecosystem due to its characteristics, such as immutable data storage and decentralized nature. A hash is used to establish a chain of blocks that constitute the ledger, where the length of the chain plays an important role in resistant to data modification. The longer the chain, the more resilient it is. This is due to the fact that if an adversary changes a transaction in a block, the change will be easily detectable because all the subsequent blocks are linked through hashes.

Although there is a wide range of applications for blockchain technology, we will focus on the use of blockchain in authentication and access control. In 2018, Hammi et al. presented in [42] a blockchain-based authentication mechanism to provide decentralized authentication to IoT devices. The proposed system, titled Bubbles of Trust, was implemented and proven to be efficient and low cost. However, the inter-communication between different systems is not supported, which makes the proposed approach inapplicable to many distributed IoT applications scenarios.

Lau et al. proposed, in 2018 as well, a protocol named Authenticated Devices Configuration Protocol (ADCP) [43]. The proposed protocol is based on blockchain technology to provide digital identification and authentication for IoT devices. Although the implementation seems to work without issues. Further investigation is required to assure that the proposed protocol is hack-proof.

Another blockchain-based authentication mechanism for IoT devices was proposed by Li et al. in [44]. The proposed system assigns a unique identifier for each individual device and records it in the blockchain. This way, devices can identify and authenticate each other without a central authority. The proposed system also included a data protection mechanism by hashing significant data (such as the IoT firmware) into the blockchain where any state changes of the data can be detected immediately. Tuli et al. in [45] introduce FogBus - a service provision tuning facility with a guaranteed integrity through blockchain, however, the use of blockchain in FogBus resulted in high latency (more time) when processing the incoming requests. Being a fog-based system made it perform better in terms of delay reduction. However, the proposed system lacked traffic-awareness which caused its overall efficiency to degrade. Although the system employed authentication mechanism, it lacked any kind of encryption. This makes it highly vulnerable to many attacks.

Non-blockchain-based authentication systems rely mostly on centralization. This centralization has a huge impact on the performance of smart transportation solutions that are being developed. Moreover, the need for delay-sensitive authentication mechanisms has become even more imperative for systems where smart vehicles are often mobile and operate in multiple regions. Thus, our proposed system employs blockchain technology and fog computing offer good grounds to build and manage distributed and decentralized trust and security solutions for time-sensitive fog-enabled systems.

2.4. B5G and federated learning

Beyond 5G (B5G) technologies have been the subject of many studies as early as the standardization of 5G. The main motivation behind that early movement is the realization of the research community of the challenges faced in 5G. These challenges can be summarized in the following points mentioned in [46]:

- Key performance challenges: Throughput, latency, energy efficiency, service creation time, battery lifetime, coverage, and total cost of ownership challenges.
- System-level challenges: privacy-by-design, quality-of-service, simplicity, density, multi-tenancy, diversity, harnessing, harvesting, mobility, location and information-context, open environment, manageability, hardening, resource management, flexibility, identity, flexible pricing, and evolution challenges.

With the highest 5G bitrate currently available being 415 Mbps [47], one of the main focus points of B5G is achieving Tbps bitrates [48]. This high-speed requires further research work on realizable massive-Multi-Input-Multi-Output (MIMO) antennas and equipment, as well as utilizing unused sub-terahertz frequency bands. Although current developments in 5G are trying to address this point, it remains an important research focus for the future.

B5G is aimed to focus at much higher carrier frequencies to try to achieve the intended throughput. Research has shown that Tbps communications will require extremely wide bandwidths that can be realized at carrier frequencies of 300 GHz or higher [48]. This requires interdisciplinary research collaboration within the areas of semiconductors, efficient communication technologies of unprecedented efficiencies reaching 1pJ/bit, and agile antenna arrays with tens to hundreds of elements.

Artificial Intelligence and Machine Learning will also play important roles in the next-generation network. First, AI and ML techniques have been proposed to manage future networks to ensure Quality of Service requirements of sensitive applications are met. Secondly, the technologies which are expected to drive beyond 5G communications, such edge/fog computing and virtualized network functions will drive new AI and ML applications which are not possible today. FL is an area of Machine Learning which can benefit most from the low-latency connections and distributed edge-based computing resources of beyond 5G networks. FL is a machine learning setting where the goal is to train a high-quality centralized machine-learning model while training data remains distributed over a large number of clients. The concept of FL was first introduced by Google in 2016 [49].

One direction of research in FL that received high attention was on-device FL where distributed mobile user interactions are involved and communication cost in massive distribution, unbalanced data distribution and device reliability are some of the major factors for optimization. To achieve this, data is partitioned by user Ids or device Ids, therefore, horizontally in the data space. In [50], Yang et al. extend the concept of FL from covering collaborative learning scenarios among organizations to a general concept for all privacy-preserving decentralized collaborative machine learning techniques.

In 2020, Du et al. published a thorough review of FL for Vehicular IoT and ITS [51]. The paper starts with a brief introduction to the topic of FL, and moves to discuss the technical challenges of applying FL in Vehicular IoT. In addition to [51], other research papers discussed the use of FL in Vehicular IoT and ITS. A short summary of these papers was shown in Table 2.

2.5. Comparison of different systems

In Table 3, a feature comparison is shown to summarize the differences between previously proposed system and our proposed system.

Table 2
Recent studies on federated learning and vehicular IoT [51].

Paper	Summary
[52]	An asynchronous FL scheme with a hybrid blockchain for IoV
[53]	A FL-based estimation of network status for URLLC
[54]	A FL-based image classification in vehicular IoT
[55]	A discussion on possible applications of FL for UAVs
[56]	An asynchronous federated learning scheme for resource sharing in vehicular IoT

3. Problem formulation and system model

3.1. Problem definition

In the TNC case of dial-a-ride, a passenger makes a request for a ride to a cloud server, the request usually has the passenger's location, the pick-up location, destination of the trip and the expected departure time. The server processes the request by sending the closest available driver. Determining the closest driver from a remotely located server in the cloud is problematic. For instance, a very popular approach is to use the Haversine Formula with longitude and latitude coordinates obtained from GPS readings. Although the Haversine Formula presents a mathematically accurate way of measuring the shortest distance between points on a sphere, using it to measure distance between two locations on the earth presents practical problems if this is done with no location-awareness. For example, two points which are on the opposite sides of a hill would be close, even though one may have to navigate around the hill from one to reach the other. Fig. 1 shows the mobile-cloud architecture. Additionally, the idea behind the fog-based framework is to allow a variety of things to communicate and cooperate with each other in order to offer a wide range of services related to public transportation. Thus, a large number of vehicles and fog nodes are expected to participate and produce important data that can be shared between vehicles and fog nodes. However, it is extremely important that only legitimate things should make use of the system. Otherwise, it will be vulnerable to various types of security attacks, such as data and identity theft, data alteration.

The goal of this project was to develop a framework for a smart public transportation system which will provide localized matching of passengers to drivers/available vehicles and provide real-time recommendations to drivers on the best route to their destination in a secure manner.

Use of fog nodes for localized matching raises the challenge of efficiently authenticating the entities which is addressed by means of a blockchain based authentication mechanism. We will first elaborate the distributed authentication challenge and later explain how a blockchain can solve this.

A client needs to be authenticated to use the localized matching service on a fog node. The authentication options can be (a) Authentication via the cloud database (b) Authentication via the fog node. Let us discuss the option *a* first, when a user is successfully authenticated by the cloud database, the next step is to convey this information to the fog node from which the user would get the matching service. This update includes the identity of the user and the time period for which the authentication is valid. Lastly, the user will present his/her identity to the concerned fog node and will be finally authenticated. Another way to do the same is to provide the user with a presentable proof of authentication and update the related fog node about this authentication and the proof. The user can then present this proof to the concerned fog node to be considered as an authenticated user. There are many schemes such as two factors authentication that can be used for this purpose. However, such a scheme would at least consist of three steps to be completed by three different actors. (1) By the user, to acquire a proof of authentication from cloud database (2) By the cloud node, to update the fog node (3) By the fog node, to verify the authentication

Table 3
Comparison of previous system to the proposed system.

Work	Integrated technologies				Security			Delay			Distributed	Traffic-Aware	Power saving
	IoT	Cloud	Fog	Blockchain	Confidentiality	Integrity	Authentication	High	Medium	Low			
Cloud-based ITS [25]		✓						✓					
Fog-based ITS [26]	✓		✓						✓		✓		
Ma et al. [22]		✓						✓					✓
Lai et al. [23]	✓	✓	✓				✓		✓		✓		✓
Brennand et al. [27]	✓		✓							✓	✓	✓	✓
Tuli et. al. [45]	✓	✓	✓	✓			✓	✓			✓		✓
Proposed work	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓

proof when presented. Other than this cost, there are other factors that add to the complexity of this scheme such as knowing the location the fog node by both, the cloud database and the user for updating the respective fog node and presenting authentication proof/identity respectively. For moving user/vehicle would it be sufficient to update one fog node or update multiple or all fog nodes. Similarly, questions like for how long the authentication proof will remain valid? what if a user needs to re-authenticate in the same region after a short period of time? need to be answered. We can safely conclude that authentication via cloud database would be a three step process and requires related challenges to be solved.

Now, let us discuss the option *b* i.e. authentication via fog node. For a fog node to be able to authenticate a user, it would require an updated copy of the cloud database. The authentication process would be very efficient however each update in the cloud database would requires updating of all fog nodes. Other than cost of maintaining database at fog nodes, database synchronization, backup, node failure etc. would pose much serious issues, also some fog nodes may not have resources to maintain the database.

Having considered the trivial options, we now consider the use of blockchain to authenticate clients via fog nodes. We propose a blockchain that resides on each main fog node along with an authentication process that allows users to be authenticated by means of this blockchain in a secure way. The main features of the proposed blockchain are as under, Section 4 provides details of the proposed blockchain, algorithms for blockchain updating/construction, authentication and discussion about consensus mechanism.

Lightweight: For each user the blockchain stores a hash of the user's data along with user's public key (in our implementation both are 256 bits long, the size of such a blockchain for 1 million user's hashes and keys would be only 61.03 mega bytes)

Efficient: Each block of the blockchain consists of a hash table to store user's public key and a hash value. The registration process provides users the block number which contains their public key and hash value. At the time of authentication user provides its public key and the block number. Authentication is performed by retrieving the block using block id, hash table of the retrieved block is searched against given public key. Cost of these operations is O(1), as the said operations are not dependent upon length of the proposed blockchain, the cost of authentication is not affected by the number of users in the system.

Secure: As no actual data is stored on the blockchain and only hashes and primary keys are stored, the privacy and security of the user is not at risk at any time. The authentication request generated by a registered user is signed with user's private key which makes it unfeasible to modify this request on its way to the fog node. To avoid replay attacks, the request message contains a nonce which is validated during the authentication process.

The framework will take advantage of fog computing and blockchain technology to overcome the location-awareness challenges of cloud and improve latency and security.

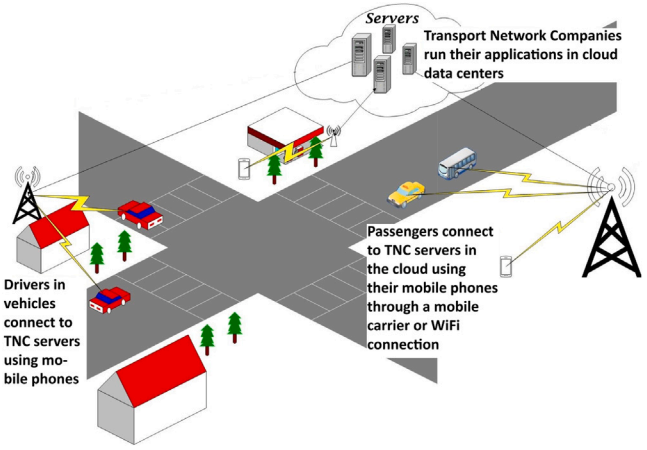


Fig. 1. Mobile-cloud architecture used by Transport Network Companies.

3.2. System model

Fig. 2 presents the framework of the proposed Fog-oriented Smart Transportation System. The system has a three-layer architecture. The system takes advantage of technologies for Beyond 5G networks such as FL, Blockchain and edge analytics. The first layer consists of cloud servers. The second part is the fog layer. The last part has the end devices and sensors which use and provide data for the system. It consists of several fog nodes distributed across a region (e.g. a city), and each connected to a central server in the cloud. Each fog node is allocated a pre-defined region and is aware of/connected to at least one other fog node — its neighbour and to the cloud. An area may also have smaller fog nodes which connect to the main fog node in the region providing networking and some computing resources for the main fog node and the devices connecting to it. The smaller fog nodes may be roadside units at remote locations or vehicles with storage and computing capabilities. Fog nodes also act as federated nodes. Each Road Side Unit uses data from vehicles within their region to train local models to be forwarded to the aggregator agent which may be in the cloud or at the fog layer. The aggregator aggregates the local models from federated nodes to create a global model. FL within this system can be applied to several use cases including traffic management and public safety.

The proposed system also provides a blockchain-based authentication mechanism which is secure and efficient. The scheme is based on public key cryptography, cryptographic hash function and blockchain. The proposed mechanism has the following components:

(1) Blockchain
(2) Consensus Node, Data Node
(3) Authentication Module
(4) Registration Module

Fig. 4 provides details of system components and their interaction while Fig. 5 provides the technical description of the authentication and

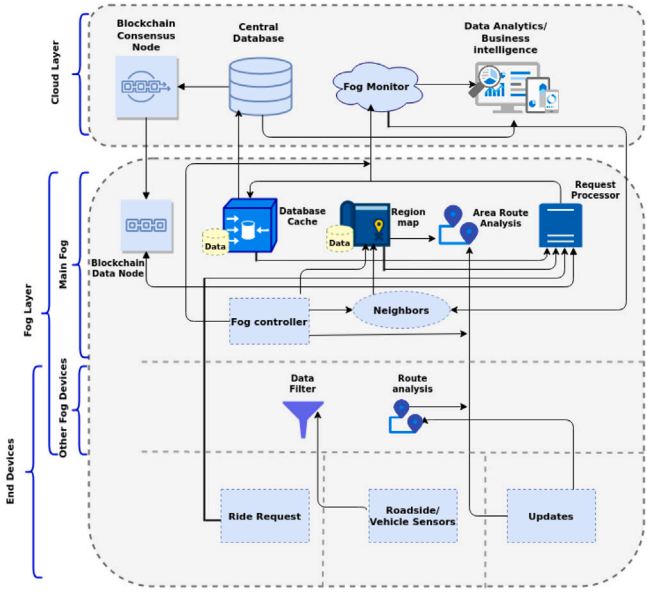


Fig. 2. Framework of the proposed system.

registration process. Details of components and their functionality is presented in Section 4.

3.3. Cloud layer

The cloud has vast computation, storage and networking resources. It is the central point of the system and maintains the resources required for the entire system to run effectively. The cloud will consist of the following components:

- Database
  The database of all the users of the system and their account information. Passengers, drivers, vehicles and payment information, as well as the fog nodes and their regions of control are stored in databases in the cloud. The database is accessed by users when they create accounts on the system. Fog nodes interact with the database to verify users and make copies of the user's record for a limited period.
- Blockchain Consensus Node
  Termed here as $Node_{con}$, is responsible for receiving new data from cloud database, updating blockchain, generating new blocks and distribute updates and blocks to other nodes in the blockchain.
- Fog Nodes Monitor
  The fog monitor actively maintains connection with fog nodes to ensure they are active. The fog monitor will also ensure that fog nodes are not overloaded. The fog node may also have redemarcate regions allocated to fog nodes to deal with overloads.
- Data Analytics/Business Intelligence Module
  The cloud layer maintains global models for various aspects of the transportation system, using local models from fog nodes which are used as federated nodes in distributed machine learning models. Fig. 3 shows a generic FL training framework for the system. The models may be useful for several sectors including traffic management, law enforcement and public safety.

3.4. Fog layer

The fog layer consists of nodes with some processing, storage and networking resources. Fogs are organized into pre-defined regions of

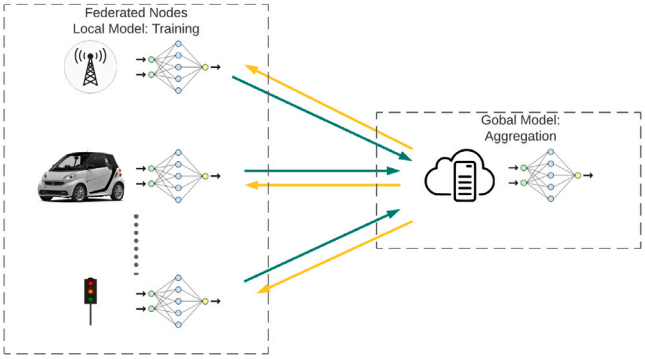


Fig. 3. Federated learning framework for fog-based NGN Intelligent Transportation System.

control. Each region has one main fog node, called the Main Fog, which receives and processes requests from passengers. Fog nodes may be distributed cloud resources available at the Radio Access Network (RAN), in Roadside Units (RSU) or vehicles with computing resources. Cloud-RAN have already been deployed by some telco operators [57]. Network softwarization and virtualization technologies including Software-Defined Networking and Network Function Virtualization used by Next-Generation Network technologies will ensure that context aware, low-latency applications run on fog nodes are reliable. Other fogs nodes within the region may carry out some computation for the main fog at the RAN but do not receive and process ride requests. The fog layer consists of five components as explained below:

- Database Cache
  Each main fog maintains records from the cloud central database. The cache database is updated as new ride requests are made and as vehicles enter the region under the control of the fog. When a user makes a ride request within a region the request is sent to the main fog, the main fog node authenticates the user via the blockchain, and requests the user's record from the cloud. The user's record is kept in the fog's database cache for a limited time period within which any requests from the user will be handled directly by the fog without contacting the cloud. The same process is followed for vehicles that enter a region.
- Blockchain/Blockchain Data Node
  Along with database cache model, we also experimented a blockchain based authentication mechanism for the same purpose, the two models have their own features which fulfils different requirements of such system. The main fog node in each region also acts as a Blockchain Data Node termed as $Node_{data}$. The node is responsible for authenticating the clients in its region. It maintains user's credential as a hash of their data along with user's public key and receives updates and new blocks from the $Node_{con}$
- Region Map
  The main fog node maintains a map of the region under its control. The map is used to monitor the traffic load on the roads in the region. The fog node also maintains a map of at least one other neighbour fog. Data from vehicles, traffic signal lights and cameras within the region is used to train the local model in the FL model (Fig. 3).
- Routes Analysis
  The route analysis module aggregates data from vehicles and sensors to estimate the cost of travelling along the roads in the region. Route analysis may be part of other fog nodes within the region. The main fog uses the data to provide recommendations for drivers as they travel to a destination.

- Request Processor
  The Request Processor is a module responsible for accepting and processing ride request from passengers. The module matches passengers to the vehicles closest to them based on records in the database at the time of the request. The module may forward requests to neighbour fogs if the request cannot be serviced by a vehicle in the region.
- Neighbour List
  Each fog node maintains record of its immediate neighbours. The information kept may include the location of the neighbour fog, the region it controls and the routes to reach the region. This information is used to forward ride requests and to handover vehicles which are completing a request. A vehicle servicing a request across regions is handed over to the next fog as it moves from one region to another.
- Fog Controller
  The Fog Controller monitors and assigns tasks to other fog devices within a region controlled by the Main Fog. Roadside Units and Vehicles with computing, storage and network resources within the region are assigned tasks to assist the Main Fog in processing data. Tasks include computing the cost of travel on a road or checking the traffic jam in an area of the region.

3.5. End devices and sensors

There are three categories of end devices:

- Mobile devices
  Passengers and drivers will interact with the system using mobile devices. The mobile application shall send requests to the fog node closest to it. Users can access the application anytime, anywhere through RAN.
- Smart Vehicles
  Smart vehicles will send sensed data to fog nodes to help determine the traffic load on the road on which the vehicle is travelling. Smart Vehicles are identified and recorded in the database.
- Roadside sensors
  Sensors along the roads send data such as number of vehicles on the road, average speed of vehicles, weather conditions etc. to fog nodes. Sensors send data regularly to fog nodes. The data is used as input for an algorithm to create a general view of the traffic situation in an area controlled by a fog node.

4. System overview

This section presents a detailed view of the system, including algorithms and some data structures used.

4.1. Users

Each main fog node maintains a record of recurrent users, $U = \{u_1, u_2, \ldots u_n\}$. Users are removed from the list after they have left the fog's region or after they have been inactive for a predefined time period, T_e. A user is added to U when they send a ride request to the fog node. The fog node requests the user's record from the cloud database. A request, $R_i(id_i, t_o, l_o, l_d)$ is received from user, u_i. Where id_i is the user id of the user, t_o is the pickup time, l_o is the pickup location for the request and l_d is the destination for the request.

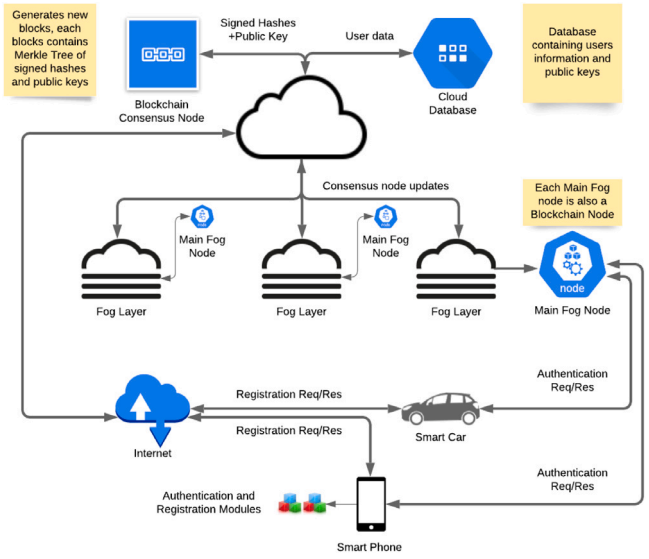


Fig. 4. IoTM Framework with Blockchain based authentication. Main components are (a) Cloud layer: Consists of Blockchain consensus and database nodes, responsible for maintaining user data and hashes for authentication (b) Fog Layer: Consists of fog devices, each region has one main fog node that also acts as Blockchain node (c) Device Layer: Consists of user application residing in smart phones/cars etc. performs registration and authentication using specific modules, there are however other modules as well.

4.2. Blockchain

The proposed system uses a custom Blockchain (BC), which stores user related data for authentication. Each block consists of a hash table that contains (a) public key of a user (b) a hash value. The hash is calculated over user data and public key, and is then signed with user's private key. The size of the hash table has a fixed upper limit $Size_{limit}$, each block has a unique identity ID_{block}, a hash pointer to the previous block and a unique sequence number N_{seq}. The proposed blockchain has following main components.

4.2.1. Registration module

The registration module is part of the user application and is used to register a user, vehicle or fog node in the system. The module generates a pair of public and private keys (key_{pu}, key_{pr}) for each new entity. The keys are generated once and later used for registration and authentication. Each new user/vehicle/fog node at the time of registration provides three items (a) user related data such as identity information, email etc., (b) signature sig i.e., data hash and public key, signed with user's private key (c) public key of the user. We used Elliptic Curve Digital Signature Algorithm (ECDSA) and Secure Hash Algorithm 3 (SHA3-256) for public/private keys, hashes and signatures. At the time of registration, a registration request containing *data, sig* and *public key* is sent to the cloud node as described in Algorithm 1. For a new user, the cloud node upon receiving the request adds the user in the cloud database along with the public key, the signed hash along with public key is sent to the Blockchain node, the process is explained in Algorithm 1. The Blockchain node $Node_{con}$ adds the user in latest block and returns the block's sequence number N_{seq} and a hash identifier to the user. This sequence number and hash identifier will be later used for authentication.

4.2.2. Authentication module

Authentication is performed by $Node_{data}$, an existing user can request authentication by sending an authentication request message. Algorithm 2 provides details of authentication process, following are the main steps of the process.

Algorithm 1 Registration Request

procedure REQUEST($data, key_{pu}, key_{pr}$)
  $hash \leftarrow Hash(data + KEY_{pu})$
  $sig \leftarrow Encrypt_{KEY_{pr}}(hash)$
  $message \leftarrow data, sig, key_{pu}$
  $result \leftarrow Send(message)$
  return *result*

procedure REGISTER USER ($message$)
  $data, sig, key_{pu} \leftarrow message$
  if $exists(data)$ **then**
    $result \leftarrow ErrorMessage$
  else
    $result \leftarrow AddtoCloudDB(data, key_{pu})$
    $result \leftarrow AddtoBlockChain(sig, key_{pu})$
  return *result*

(1) User sends $key_{pu}, block_{id}, block_{seq}$ and a message that consists of hash of the data along with hash of public key key_{pu}, the message is signed with key_{pr}
(2) $Node_{data}$ calculates hash of provided key_{pu} and compares it with hash provided, if it matches, the public key is valid
(3) Assuring the authenticity of the authentication request by validating the *nonce*. The *nonce* provided inside the signed request and the one provided with request message should be both identical and valid. Identical *nonce*s will ensure integrity of the request, while, the validity of the *nonce* is checked to counter replay attacks on authentication requests.
(4) $Node_{data}$ searches blockchain for hash of key_{pu} if found, retrieves respective data hash from blockchain. It then compares two hashes and returns the result

The mechanism is efficient as it requires only single message to be sent from the user, the elements required for forming the authentication request message have already been provided to the client from the $Node_{con}$ at the time of registration, the only computation performed to form this message is to sign with private key. On the other hand the authenticating fog-blockchain node performs the following tasks in sequence

- Signature Verification: this is done by decrypting the user data with the key provided, this also retrieves the user data and hash of the public key
- Sender Identification: this is done by computing the hash of the public key and comparing it with the hash of the key retrieved if the two hashes are same it is verified that the public key corresponds to the correct private key, this also thwarts the replay attack attempt, now the hash of the public key is searched using provided block sequence number and block identification. If the user is valid the blockchain will contain a record of the user in the specified block, if a record exists it is verified that sender is valid
- Authentication: step-3 outlined above

The total cost of computation involved can be represented as $cost = cost(E/D) + cost(S) + cost(H) + cost(C)$. Here cost indicates computation cost of an operation, E, D, H, C represents encryption, decryption, blockchain search, hash calculation and value comparisons. Only a single decryption operation is performed during the signature verification, where the data to be verified is a fixed length hash, the cost of this operation would be $O(1)$. Blockchain search is performed on a single block retrieved using the block sequence number and block identity, the block data is stored in a hash table using a Python dictionary so the search time is constant i.e. $O(1)$ and does not depend upon the size of the block, rest of the computation is constant and does not depend upon the number length of the blockchain or any other parameter. So the authentication method works with a constant cost irrespective of number of users.

Algorithm 2 Blockchain based authentication

Input: *Sig; the signed hash of public key and user data, nonce; a time stamp, public and private keys ; generated during registration phase, Sequence number; the sequence number of the block in blockchain where the user hash was stored*

procedure REQAUTH($sig, key_{pu}, key_{pr}, N_{seq}, nonce$)
  $temp \leftarrow Encrypt_{KEY_{pr}}(sig, nonce, N_{seq})$
  $message \leftarrow temp, key_{pu}, nonce$
  $result \leftarrow Send(message)$ ▷ to Regional main fog
  return *result*

Input: *Authentication request message*

procedure AUTHENTICATE($message$)
  $temp, key_{pu}, nonce' \leftarrow message$
  $sig, nonce, N_{seq} \leftarrow Decrypt_{KEY_{pu}}(temp)$
  if isInValid($nonce', nonce$) **then**
    $result \leftarrow ErrorMessage$
  else
    if $isValid(sig, key_{pu}), N_{seq}$ **then**
      $result \leftarrow SuccessMessage$
    else
      $result \leftarrow ErrorMessage$
  return *result*

4.2.3. Consensus and Blockchain updates

The Blockchain used here achieves consensus by means of a special node N_{con} residing in Cloud Layer. All updates on the Blockchain are performed by N_{con}, these updates and new blocks are then forwarded to other Blockchain nodes in the system. The details of this process is as, upon successful registration of a user the database node forwards (see Algorithm 1) its public key and *sig* to the N_{con} node which adds this to the hash table of the latest incomplete block. These updates are not immediately forwarded to other nodes until a fixed number of updates $Update_{limit}$ has been made. As the update limit approaches the latest block is sent to all the nodes in the system and the update counter is reinitialized. If during updates size of the block approaches to size limit $Size_{limit}$ the block is immediately dispatched to other nodes and a new block is started. The consensus node resides in the cloud with the database node and it bears the pros and cons of any cloud node. The consensus mechanism used here follows the model of permissioned/private blockchain where the consensus mechanism is handled by few authorized nodes, this approach suits business organizations where collaborating parties have common interest with partial trust and enforcement of business agreements and policies is a key issue. Also permissioned blockchain usually has custom requirements which are difficult to achieve through permission less or public blockchain e.g. secrecy of business data, client oriented responses, privacy, security, efficiency etc. Hyperledger Fabric [58] is an example of permissioned blockchain, it uses an Ordering Service Node (OSN) to achieve consensus. The more common, public blockchain such as Ethereum [59] and distributed consensus mechanisms such as PoW may not suit the proposed mechanism because these blockchains are designed to handle situations where new data/transactions are generated by public and chances of fraud/misuse are high therefore consensus mechanisms has to make sure that only valid blocks/records are added to the blockchain. In our case the source of new data to the blockchain is the cloud database and a new record becomes immediately available after user's registration, as only registration data is required for the authentication further transactions from the user are not needed by

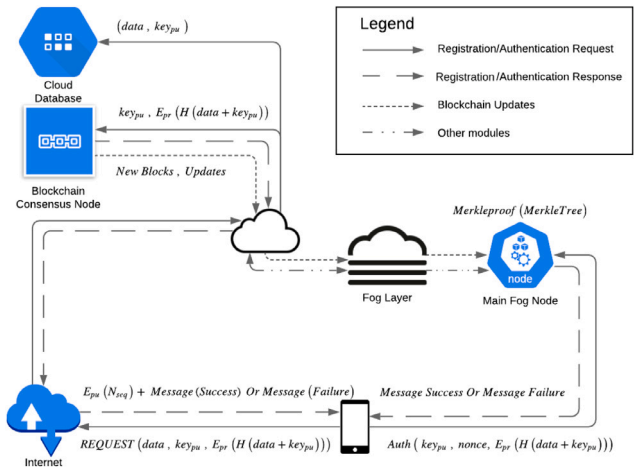


Fig. 5. Authentication and registration process, the working is explained using different line styles for registration and authentication request, respective responses, updates from blockchain consensus node and all other communications.

the blockchain. We consider it more efficient and feasible to directly added blocks to the blockchain using authorized consensus node that can directly gets updates from cloud database, if needed the proposed system can be scaled up by adding further consensus N_{con} nodes.

Each time an update is received by $Node_{data}$, the sequence number of this arriving block is compared with last received block's sequence number. If the two sequence numbers are same the old block is replaced by new one in the local blockchain, the arriving block is simply added otherwise. Algorithm 3 outlines the process of Blockchain construction.

Algorithm 3 Construct Blockchain

```
procedure AddtoBlockchain(sig, key_pu)
    block ← getBlock(latest))
    block.tree ← addNode(sig, key_pu)
    block.tableSize ← block.tableSize + 1
    if block.tableSize == Size_limit then
        sendUpdate(allnodes)
        startNewBlock()
    else
        if Update_limit then
            sendUpdate(allnodes)
    hash ← Hash(data + KEY_pu)
    blockId ← block.id
    return blockId + hash
```

4.3. Vehicles

Vehicles within a fog region send status updates, $s_i(id, l, t, sp, dr, a)$. Where for each smart vehicle s_i, id is the vehicle's identifier, l is the current location of the vehicle, t is the time the update was sent, sp is the speed at which the vehicle is travelling, dr is the direction of travel and a is the availability of the vehicle to accept ride requests.

4.4. Representation of maps

Each fog, F_i models the map of its region as a directed Graph, $G_i(V_i, E_i)$ with roads represented by edges, E_i and intersections as vertices, V_i. Two fogs, F_i, F_j are neighbours when they share at least one edge (road) in common. Thus for two neighbour fogs:

$$E_i \cap E_j \neq \varnothing$$

Let $E_{ij} = \{e_1, e_2 \dots e_n\}$, then $f_i \in V_j \wedge f_j \in V_i$. The edges $e_1 \dots e_n$ are described with the last vertices they connect to in the graph of a given node and the neighbour fog they lead to. Thus, each fog represents its neighbours as vertices in its graph.

For each edge in a fog region, the main fog computes the estimated speed of travel by vehicles with Eq. (1).

$$Cost_{e_1} = 1 - \frac{\bar{x}(s_1, s_2, \dots s_n)}{Max_{e_1}} \tag{1}$$

where $\bar{x}(s_1, s_2, \dots s_n)$ is the average travel speed of the vehicles on e_1 and Max_{e_1} is the maximum speed allowed on e_1. The speed at which vehicles are travelling on the edge is received from vehicles travelling on it along with their direction of travel. The cost of travelling in a given direction is calculated by finding the average speed of vehicles travelling on in the given direction, dividing by the speed limit of the edge, and subtracting from one. Note that a cost of zero implies that vehicles on the given edge are moving at the speed limit which should imply the limited traffic. The computation may be done by the main fog or by other fog nodes within the region under its control such as RSUs or vehicles with fog nodes on-board. Neighbours maintain connection with each other and exchange maps to facilitate route finding across fog regions.

4.5. Request processing

Fogs receive ride requests from users for processing to produce a match. A match is a pair of a driver and passenger for a trip requested by the passenger. For a request to be processed the user's record must be in the record of recurrent users of the fog node. If a requesting user is not a recurrent user the fog sends a request to the cloud for the user's record before their ride request is be processed.

Algorithm 4 takes a ride request as input and processes it to produce a match. The ride request is stored as r. If the user is not a recurrent user the fog will request the user's data from the cloud. The array $D[\,]$ are the drivers close to the pickup location. σ is a minimum allowable distance a driver must be from a passenger to be considered as a potential match.

Algorithm 4 Request Processing Algorithm

```
Require: Ride Request
Ensure: Match
if passenger is recurrent then
    r ← request;
else
    request passenger's record from cloud;
    if passenger has no account in cloud database then;
        ask user to create an account;
        create passenger's account;
    add passenger to recurrent users;
    r ← request;
if there are available drivers close to pickup location then
    D[] ← drivers closest to passenger;
    for d in D[] do
        if distance(p, d) < σ then
            send r to d;
    if d accepts request then
        create match;
        notify passenger;
else
    forward request to neighbour fog closest to passenger;
```

Once a user's record is verified, the fog node searches for drivers within a defined radius, d_o of the pickup location specified in the request. The request is sent to all the drivers within the defined radius who are available to fulfil the request. The user is notified once a nearby driver accepts the request. Where there is no driver nearby, the request is forwarded to the neighbour fog closest to the user. The

Request Processing Algorithm is shown in Algorithm 4. The condition in the expression (2) is used to locate nearby drivers.

$$2r\arcsin\left(\sqrt{\sin^2\left(\frac{\phi_v-\phi_u}{2}\right)+\cos(\phi_u)\cos(\phi_v)\sin^2\left(\frac{\theta_v-\theta_u}{2}\right)}\right)\leq d_o \quad (2)$$

where r is the radius of the earth, ϕ_v is the latitude of the vehicle's location, ϕ_u is the latitude of the user's location, θ_v is the longitude of the vehicle's location, θ_u is the longitude of the user's location, d_o is a maximum distance from the user or pickup location.

After a driver accepts a request, a match $M(l_o, t_o, id_u, id_v)$ is made, where l_o is the pickup location, t_o is pickup time, id_u is the passenger's user id and id_v is the vehicle id.

Algorithm 5 Route Recommendation Algorithm

Require: Graph, source, destination
Ensure: Best route to destination, cost of best route to destination

```
for vertex v ∈ Q do
    dist[v] ← ∞
    prev[v] ← NULL
    add v to Q
dist[source] ← 0
while destination ∈ Q do
    u ← vertex in Q with minimum distance, dist[u]
    remove u from Q
    for neighbour v of u do
        alt ← dist[u] + cost(u, v)
        if alt < dist[v] then
            dist[v] ← alt
            prev[v] ← u
return dist[ ], prev[ ]
```

4.6. Route recommendation

After a driver picks up a passenger, the fog node provides recommendations on the best route to reach the destination of the trip. Each main fog computes the weight/cost of travel on each road using Eq. (1). Fog nodes compute the shortest/best route to a destination based on the weights using Algorithm 5. Algorithm 5 is a variant of Djikstra's Shortest Path Algorithm to find the shortest path between two points.

In the algorithm the shortest distance from a given *source* to a *destination* in graph G. Where Q is an instance of the graph for the map of fog node running the computation. At the start of the algorithm each vertex v is assigned a cost of infinity from the source. The cost of travelling between two vertices $cost(u, v)$ is determined using Eq. (2).

4.7. Handover

To ensure constant connection with drivers and vehicles during a trip, B5G-enabled fog nodes handover trips and users to neighbour fogs as vehicles approach a neighbour fog. When a vehicle's current location is on an edge that is shared with a neighbour fog, and the vehicle's direction is towards the neighbour fog, the fog node in the region the vehicle is leaving sends data on the trip to the neighbour fog the vehicle is entering. The process is show in Algorithm 6. The handover is expected to be seamless to the moving vehicle due to the high communication speeds, and minimal latency provided by B5G.

Algorithm 6 takes the location of the vehicle s and its direction of travel. If the s is located on an edge which is shared with another fog and the direction of travel is towards the other fog, the vehicle is handed over to the neighbour fog. Otherwise, its location is updated.

Algorithm 6 Handover Algorithm

```
if (s.location is on an edge in E_ij) ∧ (direction is towards vertex F_j) then
    send trip, user and vehicle data to F_j
    remove user from recurrent users;
    remove vehicle from vehicles
else
    update location
```

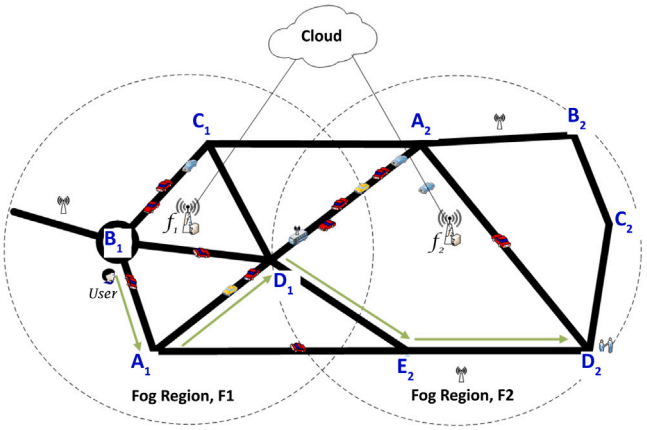


Fig. 6. A typical scenario.

4.8. Typical scenario

This section describes a typical scenario from the receipt of a ride request from a rider to the rider arriving at their desired destination, as shown in Fig. 6. A user, u_1 within B5G-enabled fog region $F1$ wants a ride from their current position, point B_1 to location D_2 in B5G-enabled fog region $F2$. u_1 sends a request to the main fog node in its current location, f_1. f_1 after receiving the request first checks if u_1 has a record in its recurrent users list. If u_1 is not a recurrent user, the B5G-enabled fog node sends a request to the cloud database to obtain a copy of the user's record. Once the user's record is in the recurrent users list the request is processed.

The first stage in the request processing is to find drivers who are close to the pickup location. Hence, the user's request will be sent through the B5G network to all drivers who are close to the pick up location and free to go. Drivers may accept or reject a request. When the B5G-enabled fog node receives the responses from drivers, the driver who can reach the pickup location quickest (based on the cost from their location to the pickup location) is selected and a match is made. Alternatively, the B5G-enabled fog node may send the list of drivers to the user using the B5G network. In this case the user selects their preferred driver based on information provided them, such as, the type of vehicle, how quickly they can reach the pickup location and the rating of the driver. Once a match is made the chosen driver and the passenger are notified. This notification is expected to reach with minimal latency, due to the communication speed provided by the B5G network. The driver proceeds to the pickup location to pick up the passenger. The scenario is also shown in the sequence diagram in Fig. 7.

In this example, the destination of the trip is located in another B5G-enabled fog region. In this scenario the originating fog knows the destination because the other fog region is within its neighbour region. The two fogs are neighbours because they share four edges, and thus exchange information regularly. At the start of the trip, the originating fog node sends a request to the neighbour with the destination for the best route from its exit to the destination of the request utilizing the B5G network. f_2 replies with edge $F1E_2$ (note that fog region $F1$ is a vertex in the graph of $F2$ and vice versa, although they each maintain

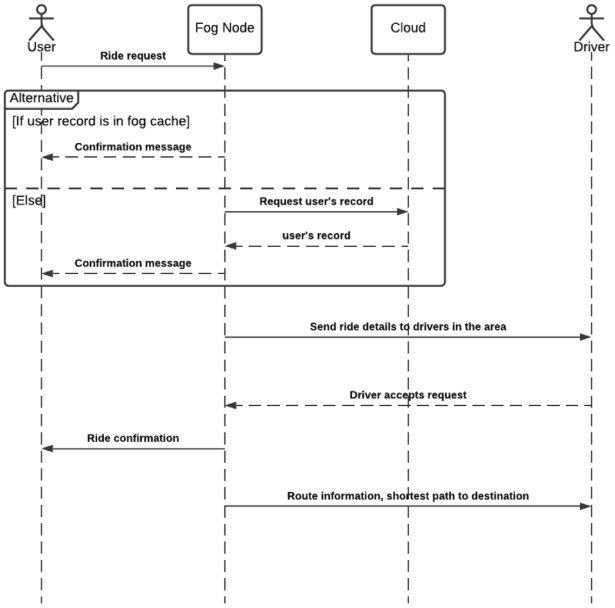


Fig. 7. Sequence diagram of a typical scenario.

a list of all locations within the other). f_1 then computes the shortest distance from the pickup location (B_1) to D_1F2. The route for the shortest distance is provided to the driver before the journey begins.

While on the trip, regular updates are sent to the B5G-enabled fog node. When the vehicle reaches the edge D_1F2, B5G-enabled fog f_1 utilizes the B5G network to forward information on the trip (the user, the vehicle and the driver) to the neighbour B5G-enabled fog f_2 (Algorithm 6) and notifies the driver and rider of the handover. Fog f_1 then removes the user from its list of recurrent users. From this point fog f_2 takes over the trip and sends the shortest path from E_2 to D_2 to the driver. Once the vehicle arrives at D_2 the trip is completed.

In Algorithm 7, Q_i is the map/graph of fog region F_i.

Algorithm 7 Destination Query Algorithm

```
if destination ∈ Q_i then
    run Algorithm 5 to determine shortest path to destination;
else if destination ∈ Q_j then
    request best entry point from neighbour;
    run Algorithm 5 using best route to neighbour as destination;
else
    send request to cloud to locate the destination;
```

It is important to note that in a real-world application other components are necessary. Modules such as billing and system security are not considered here because they are beyond the scope of this project. However, they can be included in such a system as shown in other work [23]. Also, a more complex scenario than the above may arise. For instance a passenger's destination may be within a fog region which is not a neighbour of their current fog region. In such a scenario, the fog node contacts all of its neighbours using the B5G network to search for the location from the locations of their neighbours. If the location is not found the fog node forwards the request to the cloud. This process is shown in Algorithm 7.

5. Simulation setup

According to our knowledge and literature, there is no single simulator that allows building custom blockchain scenarios along with B5G, IoT, cloud, and Fog Computing components. Most of the recent work related to B5G, integrated with Blockchain and AI, are tested and evaluated in a limited customized environment. Thereby, to evaluate the proposed authentication mechanism, we implemented a custom version of blockchain on top of an open source blockchain developed in Python. The customized version is added in a Python flask application to support HTTP requests, the application supports API calls to (a) add new transaction, which in our case is client registration request (b) create new block, which in our case is done by the cloud-blockchain node (c) authenticate, which is used by clients to send authentication request over HTTP (d) chain, to view the entire blockchain in a browser as JSON object.

Table 4
Experiment parameters.

Parameter	Value
Hashing algorithm	SHA3-256
Digital signature algorithm	ECDSA
Step size	25
Number of passes	100
Total registered clients	2475
Total authentication requests	123 750

To evaluate the proposed mechanism, we measured the time it took in authenticating clients. In a single pass we attempted to authenticate all the clients registered in the system. We started by authenticating few registered clients and increased the number of clients after each pass. For example initially there were x registered clients, we authenticated all of them and then added y more clients and started again with updated blockchain of size $(x+y)$. A total of 100 passes were created, the size was varied by a fixed value or step size, for each pass we measured the minimum, maximum and mean authentication time along with standard deviation. Table 4 provides the details of various algorithms used, the number of registered client and total authentication requests handled by the system.

The passenger application is simulated to evaluate the proposed framework. The simulation is run for two scenarios. In the first scenario all processing is done in the cloud–fog nodes (gateways) do not carry out any processing. In the second scenario, processing modules are placed in fog nodes as proposed in the framework. The simulations are run using iFogSim [60]. To evaluate the proposed framework, sections of it are modelled as applications in iFogSim. The system is evaluated for latency (delay), network usage and energy consumption of devices in the system. Both scenarios use the same physical topology, as shown in Fig. 8.

The mobile application has 3 modules, the client module, control module and the matching as shown in Fig. 9. In the simulation setup we assume that all users within the fog region are regular users and thus fog nodes do not interact with the cloud database for the fog-only mode. Future investigations will look into the effect of using the cloud database. A user enters details of their ride request using an application on their smartphone (the client module). The user's input and output devices are modelled as sensors and actuators respectively in iFogSim. The client module pre-processes the request and forwards it to the router or gateway device. The Fog Controller Module serves an interface between the fog device/gateway and the user devices. It receives user requests, extracts the appropriate input information and passes the request to the Matching module. The Matching Module runs Algorithm 4 and returns a match for the request. To model situation where users' requests are sent to the cloud or neighbour fogs the relationship between the ride request tuple and the input tuple entering and leaving the Controller Module respectively is set as a fractional selectivity of 0.6. This implies that only 60% of requests are forwarded to the Matching module in the simulation.

Two simulation modes are run. In the first scenario the gateway routes the request to the cloud/data centre for processing. In the second scenario the request is processed by the gateway device and

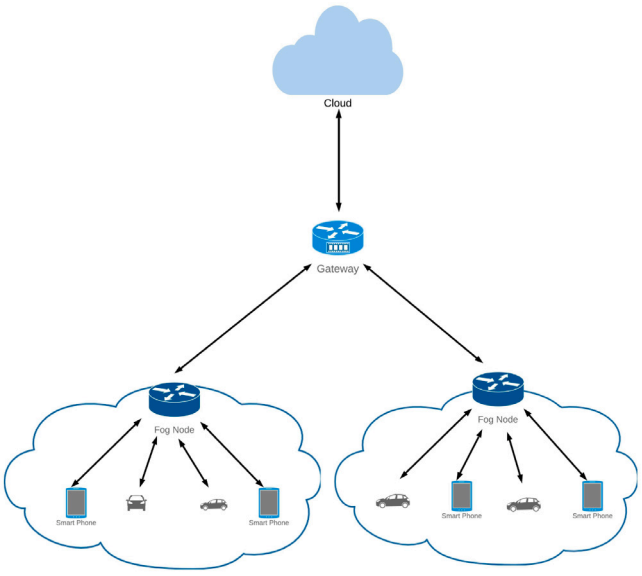


Fig. 8. Simulation topology.

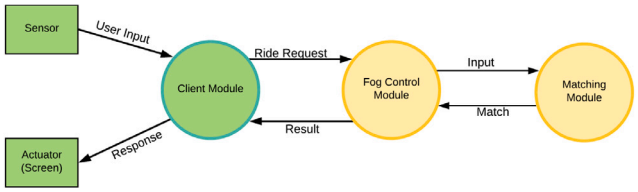


Fig. 9. Model for ride request application.

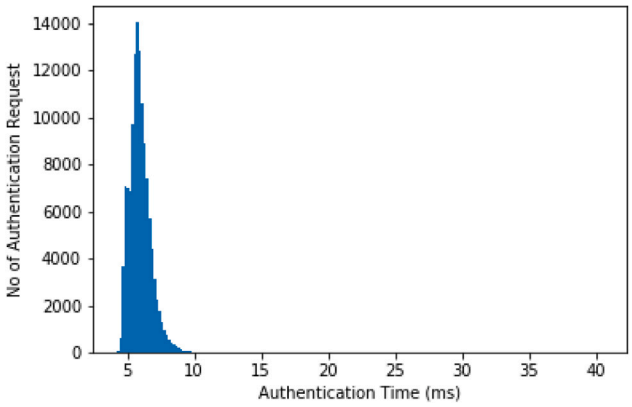


Fig. 10. Authentication time.

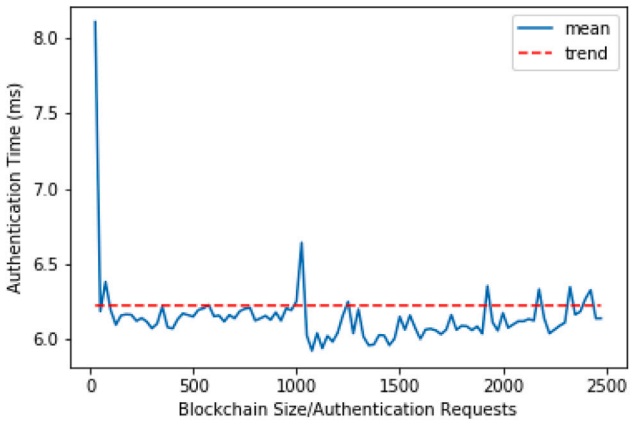


Fig. 11. Performance of proposed authentication mechanism.

the proxy-server — it is not forwarded to the cloud server. iFogSim provides two module placement strategies that make this possible: cloud-only placement module and edge-ward placement module. In the simulations, cloud placement module is used for cloud based processing of users' requests and edge-ward placement is used for processing by devices at the edge of the network. In cloud-only placement all modules of the application are placed in the cloud data centre. All user requests are sent to the cloud, processed and the results sent back to the user (actuator). With edge-ward placement iFogSim attempts to place modules as close as possible to the user depending on the availability of processing power on edge devices. When a device has limited resources for a module, the module is placed on the next available edge device.

To compare the performance of the system for cloud-based or fog-based scenarios the application loop delay, network usage and energy consumption of devices is measured for a setup with 1 cloud data centre, 1 proxy, 2 gateways and varying number of mobile devices connected to a gateway from 2 to 20 users.

6. Results and discussion

In this section the results from the simulations described above are presented and discussed. Four metrics were considered during the simulations. They are authentication delay, application delay, network usage and energy consumption of devices. For each metric we compare the performance for the cloud-based approach to the fog-based approach. The results are also compared to results obtained by other researchers in previous related work.

6.1. Authentication efficiency

Fig. 11 shows the mean authentication time at different blockchain sizes for 100 passes. The time is measured in milliseconds while the blockchain size is taken as number of registered clients. The initial spike in the plot is due to the fact that the blockchain authentication related data structures are initialized when the first authentication request is received, the small variations along the trend line are due to the delay caused by creation of new blocks. The trend line indicates that the performance does not fluctuate due to increase in size of the blockchain consequently increased number of authentication requests, Table 5 provides the results of experiments. In each experiment a total of 100 passes with different blockchain size and number of authentication requests processed were carried out. Fig. 13 shows the change in blockchain size and number of cumulative authentication requests processed till then for each pass. With each pass 25 more clients are registered and then all the clients are authenticated to calculate the authentication time after this update. In 100 passes a total of 2500 clients were registered (indicated by red line), the blue line shows sum of all authentication requests for each pass. Experimental data shows that proposed authentication is not affected by the change in size of the blockchain and number of authentication requests. Figs. 10 and 11 show the performance of proposed mechanism. The plot in Fig. 11 is generated using mean authentication time measured at various loads as described in Section 4, the histogram in Fig. 10 indicates that authentication time of majority of the requests range between 04 to 08 ms, where as the number of times authentication requests took more than 09 ms is negligible. While Table 4 shows the parameters for the experiments, Table 5 shows the min, max, mean, and standard deviation, of the authentication time for various experiments (see Fig. 12).

We compare our proposed blockchain based authentication scheme with similar existing approaches. Table 6 outlines the features and parameters of these approaches, which are common to our work along

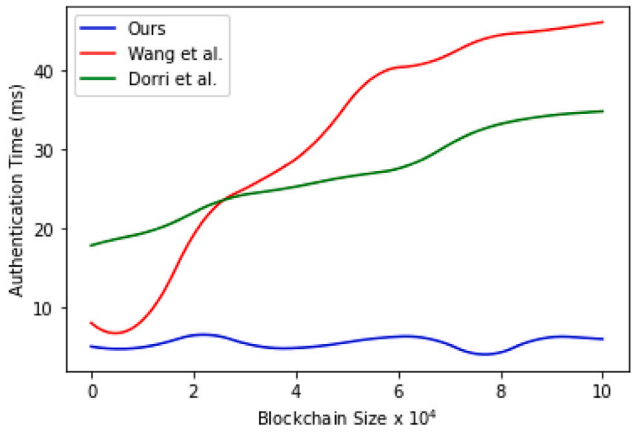


Fig. 12. Performance comparison of our proposed mechanism with Wang et al. [61] and Dorri et al. [62] details provided in Table 5.

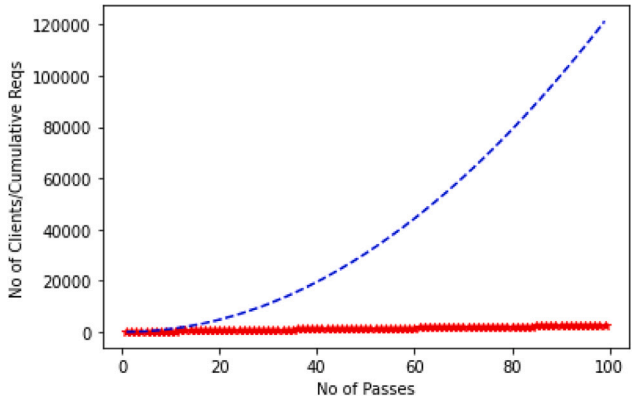


Fig. 13. A step-wise increase in size of the blockchain with each pass for 100 passes and sum of authentication requests handled by the system till up-to each pass.

Table 5
Experiment results.

Experiment	Min	Max	Mean	Div
1	5.92	8.10	6.14	0.2218
2	5.85	7.77	6.19	0.2219
3	5.88	8.15	6.15	0.2587
Combined	5.85	8.38	6.09	0.2924

with the performance comparison. The following paragraphs discuss and compare each of these approaches with our work. The work presented in [61] has many common attributes to our work. In [61], the authentication cost or authentication time increases with the increase in the size of the blockchain. Similarly the authentication time of proposed mechanism in [62] which uses a light weight blockchain [64], increases with blockchain size, this is due to the use of a decentralized consensus mechanism and special nodes called Overlay Block Managers (OBMs) that execute the consensus mechanism. As the network grows, the number of OBMs increases and leads to increased communication for developing a consensus about the validity of the credential presented by a client. In our case, the consensus mechanism is centralized and is controlled by a special node called Consensus Node, which resides with the cloud node and broadcasts updates to all data nodes. Also, our approach uses a blockchain lookup for authentication rather than developing a consensus which is very time consuming in comparison to the lookup operation.

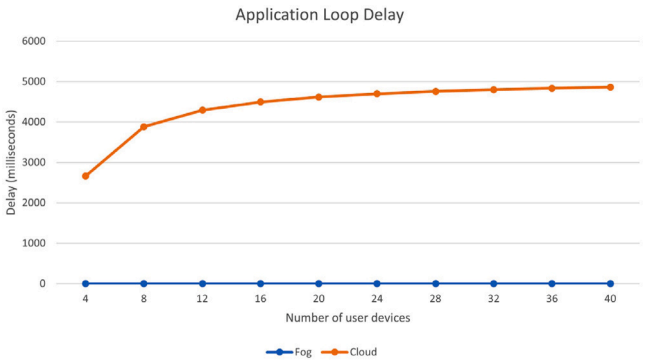


Fig. 14. Application delay.

A similar concept of using blockchain for authenticating IoT devices is also proposed by [44]. It presents a proof of concept of using blockchain on an experimental IoT setup that supports and validates our idea of using blockchain for authenticating clients. [63] provides a way to not only authenticate, but also to control the access to IoT devices using a permissioned blockchain. The work is similar to our approach except that the blocks are added by means of developing a consensus (PBFT) algorithm. To compare our work with [63] it is important to understand that both use custom implementation of blockchain in different programming languages — also, the code execution environments are not the same. The verification process in this work takes a minimum of 28 ms, maximum 40 ms, and on the average is 33 ms.

Fig. 11 shows that in both of the compared works the cost of authentication in terms of time increases with blockchain size, this is due to the decentralized consensus mechanisms whereas in proposed mechanism the authentication cost does not varies with size of the blockchain this indicates that the proposed system is scalable and more efficient. Though, there are differences in implementations and cryptographic algorithms yet it is very exciting that our implementation provides much smaller authentication delays.

6.2. Application delay

A major goal of moving the processing and storage from the cloud to the network edge is to reduce the time end devices have to wait for a response after they send input for processing. TNC applications require quick response to user request to maintain high user satisfaction. The complete application delay was measured by measuring the delay of the application loop from the user input to client, control module, matching module and back to control module. Simulation results (Fig. 14) show a big difference between the application latency for running the user request application in the cloud and on fog nodes. The result is also consistent with similar comparisons made for other applications in other research [65]. Compared to the delay from a cloud-based approach the delay from the fog-based approach is negligible. The cloud datacentre is a major bottleneck in the cloud-based implementation as all the requests from users are sent to it. Consequently, as the numbers of users grow the performance is affected and thus user-experience is impacted. On the other hand, the simulations in this project were done for up to 40 end users. Further studies may be needed to show the behaviour of the system for much larger number of users.

6.3. Network usage

Fig. 15 shows the network use of the application for both cloud and fog approaches. As the number of users increases the amount of data used by the system increases linearly for both approaches. The results show a consistent difference between the cloud and fog network use.

Table 6
Performance comparison.

Feature	[61]	[62]	[44]	[63]	Our
Blockchain	Ethereum	Custom	Hyperledger Fabric	Hyperledger Fabric	Custom
Digital signature	Yes	Yes	Yes	Yes	Yes
Consensus	Decentralized	Decentralized	Decentralized	Decentralized	Centralized
Cryptography	PKI	PKI	PKI	PKI	PKI
Application	IoV	IoV	IoT	Smart Factory	Transport System
Cloud	Yes	Yes	Yes	Yes	Yes
Fog	No	No	No	No	Yes
Evaluation	Simulator (Venis)	Simulator	Hardware	Custom Implementation (JUICE, Solidity, Java)	Custom Implementation (Python, Flask)
Performance	Authentication delay increases with increase in blockchain size	Authentication delay increases with increase in blockchain size	Only validates the concepts of blockchain based authentication	Authentication Delay Min = 28 ms, Max = 40 ms, Avg = 33 ms	Authentication time independent of blockchain size. Authentication delay Min = 5.85, Max = 8.38, Avg = 6.09

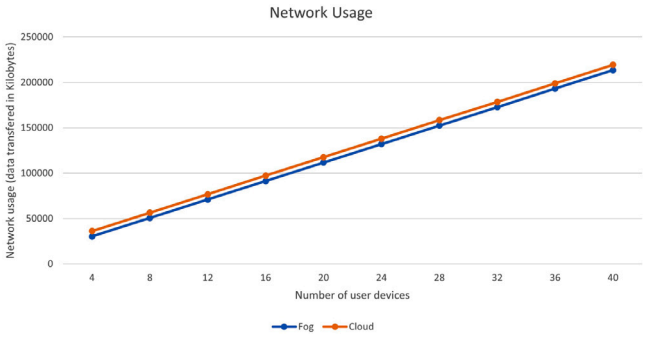


Fig. 15. Network use/number of connected devices.

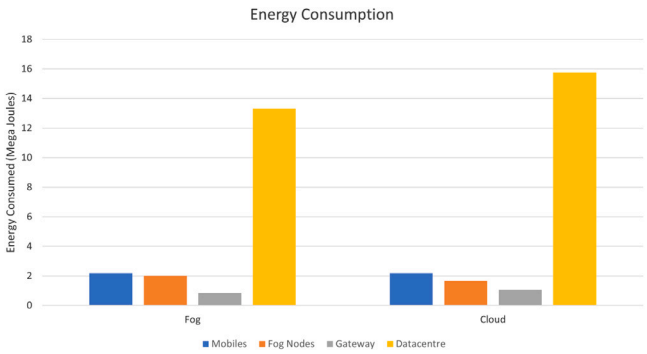


Fig. 16. Energy consumption.

This result diverges from results from other studies such as [60] where fog-based deployment show a much slower growth. The difference may be as a result of the use of a gateway/proxy between fog regions and the cloud. The Gateway device aggregates requests to the cloud and thus reduce the overhead network capacity required for a collection of requests. In a highly distributed system the network usage for cloud deployment would be significantly higher.

6.4. Energy consumption

Fig. 16 shows the energy consumption of the different groups of devices in the simulation for the two modes of deployment. The figure shows energy consumed when running 40 user devices (smartphones). The total energy for mobile devices remains constant. This is because mobile devices run the client module in both cases. Energy consumed by fog nodes reduced in the cloud deployment because fog nodes carry out only networking functions in the cloud mode. Also, the gateway energy increases in cloud mode compared to fog mode because when running in the fog nodes gateways do not forward network traffic to the cloud. Moreover, it is observed that energy consumption of the cloud increases with the change in approach. A further observation is the significant energy consumption of the cloud under fog deployment. This is unexpected since no data is sent to the cloud when the system is run on fog devices. However the consumption is due to the idle state power use which is significantly higher for the cloud. The results show that processing in the fog devices cost significantly less energy than processing in the cloud, for the same application. Given that any framework that needs less energy, processing and memory is considered to be light, hence, this obviously proves that the proposed framework is lightweight.

7. Conclusions and future work

A major challenge for TNCs is to guarantee user satisfaction by ensuring prompt response to requests. Although they provide an effective means of transportation in urban areas, some studies suggest that TNCs may be contributing to the worsening case of traffic jams in cities. Furthermore, the reliance solely on GPS for location information and the use of cloud-based servers to process data present challenges of latency and location-awareness in such systems. On the other hand, Fog Computing and blockchain, as a complement to Cloud Computing provides tremendous opportunity for improving latency, location-awareness and scalability for transport applications. Along with advances in vehicular sensors and networking, Fog and blockchain present opportunity to meet requirements which have been impossible to meet in current systems. The main purpose of this work is to rely on the future technologies of 5G and B5G that integrate with FL, blockchain, and edge computing to present a conceptual model of a futuristic smart transportation system. The proposed framework combines aspects of Intelligent Transportation with the operation model of Transportation Companies. The framework was based on a three-layer architecture with a cloud layer, fog layer consisting of fog nodes organized into areas/regions, and end devices such as mobile phones and smart vehicles. A blockchain-based decentralized mechanism was included to authenticate fog nodes and smart vehicles in order to allow only legitimate entities to communicate with the proposed framework. The framework also included algorithms for matching passengers with drivers, handing over passengers moving from one fog region to another and determine the best route to reach a given destination. Furthermore, the project proposed an approach for determining the level of traffic congestion on a given road using travel speeds of vehicles

on the road. The proposed system covers most of the ITS and fog-based ITS mentioned in Table 1.

In addition to the above, this paper also presented partial comparison of the proposed approach to a cloud approach using simulations. The simulations were to compare performance of the two approaches in terms of network usage, energy consumption and application delay. The ride request application was used in the simulations, using the iFogSim simulator. The results of the simulation showed better performance of a fog-based approach compared to cloud-based approach, particularly for application delay. The results were consistent with some other work done by other researchers in some cases. However, further evaluation would be required to confirm the conclusions as some as simulations were based on a specific scenario and topology and included a limited number of devices.

As mentioned above, simulations and resulting comparisons were done based on network usage, energy consumption and application delay. Hence, further evaluation of the proposed framework can be conducted to further establish its efficiency in comparison to existing and other proposed approaches. Additionally, the FL model for smart transportation presented will be enhanced and evaluated. The use of FL transportation present several other application use cases which must be investigated further. Moreover, the impact of the communication overhead from interactions between fog nodes and the cloud was not considered in this study and will be considered in the future. Another aspect of evaluation can be done for the impact of mobility on the system performance which was not possible in our implementation. Evaluations can also be done using real life data sets, similar to what was done in a few other studies.

CRediT authorship contribution statement

Thar Baker: Conceptualization, Data curation, Formal analysis, Supervision, Software, Validation, Project administration, Writing – original draft, Writing – review & editing. **Muhammad Asim:** Visualization, Data curation, Validation, Formal analysis, Writing – original draft, Writing – review & editing. **Hezekiah Samwini:** Formal analysis, Data curation, Validation, Writing – original draft, Writing – review & editing. **Nauman Shamim:** Formal analysis, Writing – original draft, Writing – review & editing. **Mohammed M. Alani:** Formal analysis, Writing – original draft, Writing – review & editing. **Rajkumar Buyya:** Visualization, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is partially supported by the University of Sharjah SEED Research Project, United Arab Emirates (2102150209).

References

[1] Worldometer, World population clock: 7.8 billion people (2020) - worldometer, 2020, (Accessed on 06/07/2021), https://www.worldometers.info/world-population/#::text=World.

[2] UNFPA, State of World Population 2007, Tech. Rep., United Nations Population Fund, 2007, p. 108, [Online]. Available: www.unfpa.org.

[3] J.A. Guerrero-ibanez, S. Zeadally, J. Contreras-Castillo, Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies, IEEE Wirel. Commun. 22 (6) (2015) 122–128.

[4] BBC, UK must tackle 'astonishing' cost of congestion, 2018, [Online]. Available: https://www.bbc.co.uk/news/uk-42948259.

[5] EU-Mobility and Transport, Transport in the European Union Current Trends and Issues, Tech. Rep., European Commission, Brussels, 2018, p. 144, [Online]. Available: https://ec.europa.eu/transport/sites/transport/files/2018-transport-in-the-eu-current-trends-and-issues.pdf.

[6] Z. Li, Y. Hong, Z. Zhang, An empirical analysis of on-demand ride-sharing and traffic congestion, in: International Conference on System Sciences, Hawaii, 2017, pp. 4–13.

[7] J. Cramer, A.B. Krueger, Disruptive change in the taxi business: The case of uber, Amer. Econ. Rev. 106 (5) (2016) 177–182.

[8] A. Brodeur, K. Nield, Has Uber Made It Easier to Get a Ride in the Rain? Tech. Rep., University of Ottawa, Department of Economics, University of Ottawa, Ottawa, 2017.

[9] K. Chadha, The global positioning system: challenges in bringing GPS to mainstream consumers, in: 1998 IEEE International Solid-State Circuits Conference. Digest of Technical Papers, ISSCC. First Edition (Cat. No.98CH36156), IEEE, San Francisco, CA, USA,, 1998, pp. 26–28.

[10] W. Saad, M. Bennis, M. Chen, A vision of 6G wireless systems: Applications, trends, technologies, and open research problems, IEEE Netw. 34 (3) (2020) 134–142.

[11] Q. Bi, Ten trends in the cellular industry and an outlook on 6G, IEEE Commun. Mag. 57 (12) (2019) 31–36.

[12] R. Sahal, S.H. Alsamhi, K.N. Brown, D. O'Shea, C. McCarthy, M. Guizani, Blockchain-empowered digital twins collaboration: Smart transportation use case, Machines 9 (9) (2021).

[13] S. Dang, O. Amin, B. Shihada, M.-S. Alouini, What should 6G be? Nature Electron. 3 (1) (2020) 20–29.

[14] Y.-C. Liang, D. Niyato, E.G. Larsson, P. Popovski, Guest editorial: 6G mobile networks: Emerging technologies and applications, China Commun. 17 (9) (2020) 90–91.

[15] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H.B. McMahan, et al., Towards federated learning at scale: System design, 2019, arXiv preprint arXiv:1902.01046.

[16] C. She, R. Dong, Z. Gu, Z. Hou, Y. Li, W. Hardjawana, C. Yang, L. Song, B. Vucetic, Deep learning for ultra-reliable and low-latency communications in 6g networks, IEEE Netw. 34 (5) (2020) 219–225.

[17] S. Savazzi, M. Nicoli, M. Bennis, S. Kianoush, L. Barbieri, Opportunities of federated learning in connected, cooperative, and automated industrial systems, IEEE Commun. Mag. 59 (2) (2021) 16–21.

[18] L. Tan, H. Xiao, K. Yu, M. Aloqaily, Y. Jararweh, A blockchain-empowered crowdsourcing system for 5g-enabled smart cities, Comput. Stand. Interfaces 76 (2021) 103517.

[19] D.C. Nguyen, P.N. Pathirana, M. Ding, A. Seneviratne, Blockchain for 5G and beyond networks: A state of the art survey, J. Netw. Comput. Appl. 166 (2020) 102693.

[20] M. Ali, H. Karimipour, M. Tariq, Integration of blockchain and federated learning for internet of things: Recent advances and future challenges, Comput. Secur. (2021) 102355.

[21] N. Tariq, M. Asim, F.A. Khan, T. Baker, U. Khalid, A. Derhab, A blockchain-based multi-mobile code-driven trust mechanism for detecting internal attacks in internet of things, Sensors 21 (1) (2021) 23.

[22] S. Ma, Y. Zheng, O. Wolfson, Real-time city-scale taxi ridesharing, IEEE Trans. Knowl. Data Eng. 27 (7) (2015) 1782–1795.

[23] Y. Lai, F. Yang, L. Zhang, Z. Lin, Distributed public vehicle system based on fog nodes and vehicular sensing, IEEE Access 6 (2018) 22011–22024.

[24] J.-F. Cordeau, G. Laporte, The dial-a-ride problem: models and algorithms, Ann. Oper. Res. 153 (1) (2007) 29–46.

[25] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, C. Chen, Data-driven intelligent transportation systems: A survey, IEEE Trans. Intell. Transp. Syst. 12 (4) (2011) 1624–1639.

[26] T.S.J. Darwish, K. Abu Bakar, Fog based intelligent transportation big data analytics in the internet of vehicles environment: Motivations, architecture, challenges, and critical issues, IEEE Access 6 (2018) 15679–15701.

[27] C.A.R.L. Brennand, F.D. da Cunha, G. Maia, E. Cerqueira, A.A. Loureiro, L.A. Villas, FOX: A traffic management system of computer-based vehicles FOG, in: 2016 IEEE Symposium on Computers and Communication (ISCC), IEEE, Messina, Italy, 2016, pp. 982–987.

[28] S. Basudan, X. Lin, K. Sankaranarayanan, A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing, IEEE Internet Things J. 4 (3) (2017) 772–782.

[29] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, S. Chen, Vehicular fog computing: A viewpoint of vehicles as the infrastructures, IEEE Trans. Veh. Technol. 65 (6) (2016) 3860–3873.

[30] Y. Xiao, C. Zhu, Vehicular fog computing: Vision and challenges, in: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), IEEE, Kona, HI, USA, 2017, pp. 6–9.

[31] C. Huang, R. Lu, K.-K.R. Choo, Vehicular fog computing: Architecture, use case, and security and forensic challenges, IEEE Commun. Mag. 55 (11) (2017) 105–111.

[32] F.H. Rahman, A. Yura Muhammad Iqbal, S.S. Newaz, A. Thien Wan, M.S. Ahsan, Street parked vehicles based vehicular fog computing: TCP throughput evaluation and future research direction, in: 2019 21st International Conference on Advanced Communication Technology (ICACT), IEEE, PyeongChang Kwangwoon_Do, Korea (South), Korea (South), 2019, pp. 26–31.

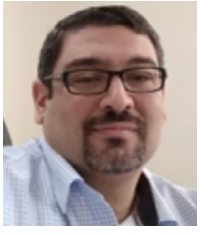
[33] J. Liu, J. Li, L. Zhang, F. Dai, Y. Zhang, X. Meng, J. Shen, Secure intelligent traffic light control using fog computing, Future Gener. Comput. Syst. 78 (2018) 817–824.
[34] N. Noorani, S.A.H. Seno, Routing in VANETs based on intersection using SDN and fog computing, in: 8th International Conference on Computer and Knowledge Engineering, ICCKE 2018, IEEE, Mashhad, Iran, 2018, pp. 339–344.
[35] Z. Ning, J. Huang, X. Wang, Vehicular fog computing: Enabling real-time traffic management for smart cities, IEEE Wirel. Commun. 26 (1) (2019) 87–93.
[36] Y. Lai, H. Lin, F. Yang, T. Wang, Efficient data request answering in vehicular ad-hoc networks based on fog nodes and filters, Future Gener. Comput. Syst. 93 (2019) 130–142.
[37] A. Chaer, K. Salah, C. Lima, P.P. Ray, T. Sheltami, Blockchain for 5G: Opportunities and challenges, in: 2019 IEEE Globecom Workshops (GC Wkshps), IEEE, 2019, pp. 1–6.
[38] A. Rago, P. Ventrella, G. Piro, G. Boggia, P. Dini, Towards an optimal management of the 5G cloud-RAN through a spatio-temporal prediction of users' dynamics, in: 2020 Mediterranean Communication and Computer Networking Conference (MedComNet), IEEE, 2020, pp. 1–4.
[39] W. Serrano, The blockchain random neural network for cybersecure IoT and 5G infrastructure in smart cities, J. Netw. Comput. Appl. 175 (2021) 102909.
[40] U. Khalid, M. Asim, T. Baker, P.C. Hung, M.A. Tariq, L. Rafferty, A decentralized lightweight blockchain-based authentication mechanism for IoT systems, Cluster Comput. (2020) 1–21.
[41] M.A. Jan, P. Nanda, X. He, Z. Tan, R.P. Liu, A robust authentication scheme for observing resources in the internet of things environment, in: 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, IEEE, 2014, pp. 205–211.
[42] M.T. Hammi, B. Hammi, P. Bellot, A. Serhrouchni, Bubbles of trust: A decentralized blockchain-based authentication system for IoT, Comput. Secur. 78 (2018) 126–142.
[43] C.H. Lau, K.-H.Y. Alan, F. Yan, Blockchain-based authentication in IoT networks, in: 2018 IEEE Conference on Dependable and Secure Computing (DSC), IEEE, 2018, pp. 1–8.
[44] D. Li, W. Peng, W. Deng, F. Gai, A blockchain-based authentication and security mechanism for IoT, in: 2018 27th International Conference on Computer Communication and Networks (ICCCN), IEEE, 2018, pp. 1–6.
[45] S. Tuli, R. Mahmud, S. Tuli, R. Buyya, FogBus: A blockchain-based lightweight framework for edge and fog computing, J. Syst. Softw. 154 (2019) 22–36, http://dx.doi.org/10.1016/j.jss.2019.04.050.
[46] E. NetWorld2020 and others, 5g: Challenges, research priorities, and recommendations, Joint White Paper September, 2014.
[47] Infographic: The cities with the fastest 5G speeds, 2021, [Online; accessed 15. Oct. 2021], https://www.statista.com/chart/24771/cities-with-the-fastest-5g-download-speeds-globally,
[48] R. Aguiar, J. Sebanstien-Bedo, A. Garcia Armada, B. Evans, A. Galis, H. Karl, White paper for research beyond 5G, 2016.
[49] J. Konečný, H.B. McMahan, F.X. Yu, P. Richtárik, A.T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, 2016, arXiv preprint arXiv:1610.05492.
[50] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Trans. Intell. Syst. Technol. (TIST) 10 (2) (2019) 1–19.
[51] Z. Du, C. Wu, T. Yoshinaga, K.-L.A. Yau, Y. Ji, J. Li, Federated learning for vehicular internet of things: Recent advances and open issues, IEEE Open J. Comput. Soc. 1 (2020) 45–61.
[52] Y. Lu, X. Huang, K. Zhang, S. Maharjan, Y. Zhang, Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles, IEEE Trans. Veh. Technol. 69 (4) (2020) 4298–4311.
[53] S. Samarakoon, M. Bennis, W. Saad, M. Debbah, Distributed federated learning for ultra-reliable low-latency vehicular communications, IEEE Trans. Commun. 68 (2) (2019) 1146–1159.
[54] D. Ye, R. Yu, M. Pan, Z. Han, Federated learning in vehicular edge computing: A selective model aggregation approach, IEEE Access 8 (2020) 23920–23935.
[55] B. Brik, A. Ksentini, M. Bouaziz, Federated learning for UAVs-enabled wireless networks: Use cases, challenges, and open problems, IEEE Access 8 (2020) 53841–53849.
[56] Y. Lu, X. Huang, Y. Dai, S. Maharjan, Y. Zhang, Differentially private asynchronous federated learning for mobile edge computing in urban informatics, IEEE Trans. Ind. Inf. 16 (3) (2019) 2134–2143.
[57] S. Kitanov, T. Janevski, State of the art: Mobile cloud computing, in: 2014 Sixth International Conference on Computational Intelligence, Communication Systems and Networks, 2014, pp. 153–158.
[58] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, et al., Hyperledger fabric: a distributed operating system for permissioned blockchains, in: Proceedings of the Thirteenth EuroSys Conference, 2018, pp. 1–15.
[59] G. Wood, et al., Ethereum: A secure decentralised generalised transaction ledger, Ethereum Project Yellow Pap. 151 (2014) (2014) 1–32.
[60] H. Gupta, A. Vahid Dastjerdi, S.K. Ghosh, R. Buyya, IFogSim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments, Softw. - Pract. Exp. 47 (9) (2017) 1275–1296.
[61] X. Wang, P. Zeng, N. Patterson, F. Jiang, R. Doss, An improved authentication scheme for internet of vehicles based on blockchain technology, Vol. 7, IEEE, 2019, pp. 45061–45072.
[62] A. Dorri, M. Steger, S.S. Kanhere, R. Jurdak, Blockchain: A distributed solution to automotive security and privacy, IEEE Commun. Mag. 55 (12) (2017) 119–125.
[63] C. Lin, D. He, X. Huang, K.-K.R. Choo, A.V. Vasilakos, BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0, J. Netw. Comput. Appl. 116 (2018) 42–52.
[64] A. Dorri, S.S. Kanhere, R. Jurdak, Towards an optimized blockchain for IoT, in: 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI), IEEE, 2017, pp. 173–178.
[65] P.K. Kandregula, Comparing Fog Based Smart Parking System to Cloud Based Smart Parking System (Ph.D. thesis), Texas A&M University, 2018, [Online]. Available: https://search.proquest.com/docview/2248788552?pq-origsite=gscholar.

Thar Baker is Associate Professor in the Department of Computer Science at The University of Sharjah (UoS) in UAE. Before joining UoS, Thar was Reader in Cloud Engineering and Head of Applied Computing Research Group (ACRG) in the Faculty of Engineering and Technology at Liverpool John Moores University (LJMU, UK). He received his Ph.D. in Autonomic Cloud Applications from LJMU in 2010, and became a Senior Fellow of Higher Education Academy (SFHEA) in 2018. Dr Baker has published numerous refereed research papers in multidisciplinary research areas including parallel and distributed computing, algorithm design, green and sustainable computing, and energy routing protocols.

Muhammad Asim is an Associate Professor at the Department of Computer Science, National University of Computer and Emerging Sciences, Pakistan. Having attained a Ph.D from Liverpool John Moores University, he researches in the fields of Cloud Computing, Computer Networks, Network Security, Internet of Things and Wireless Sensor Networks.

Hezekiah Samwini received the B.Sc. degree in Telecommunication Engineering from Kwame Nkrumah University of Science and Technology and the MSc. in Computing and Information Systems from Liverpool John Moores University. His research interests include Distributed Systems, Security in Cloud Computing and Big Data Analytics.

Nauman Shamim is a Ph.D. student in the Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Pakistan. He received his MS degree in Computer Science from National University of Sciences and Technology (NUST). His research interests include Cyber Security, Blockchain, and machine learning.



Mohammed Alani is a Professor of Cybersecurity and Networking. He has worked in many institutions in the Middle East since he received his Ph.D. in Computer Engineering in 2007. His major fields of interest include big data applications and machine learning applications in cybersecurity. He has authored four books in various fields of security and networking in addition to many papers in reputable high-impact venues.

Rajkumar Buyya is a Redmond Barry Distinguished Professor and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He is also serving as the founding CEO of Manjrasoft, a spin-off company of the University, commercializing its innovations in Cloud Computing. He served as a Future Fellow of the Australian Research Council during 2012–2016. He serving/served as Honorary/Visiting Professor for several elite Universities including Imperial College London (UK), University of Birmingham (UK), University of Hyderabad (India), and Tsinghua University (China). He has authored over 725 publications and seven text books including "Mastering Cloud Computing" published by McGraw Hill, China Machine Press, and Morgan Kaufmann for Indian, Chinese and international markets respectively. He also edited several books including "Cloud Computing: Principles and Paradigms" (Wiley Press, USA, Feb 2011). He is one of the highly cited authors in computer science and software engineering worldwide (h-index = 137, g-index = 304, 100,000+ citations).