MŰEGYETEM 1782

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Telecommunications and Media Informatics

Blockchain-assisted federated learning in 5G-powered vehicular systems

Scientific Students' Association Report

Author:

Marcell Szabó

Advisor:

László Toka PhD

2023

Contents

Abstract **i**

1 Introduction **1**

2 Use case description **4**

3 Related work **6**

3.1 Building of HD maps 6

3.2 Applications of FL in autonomous transportation 8

3.3 Blockchain-based reliability and incentive mechanism 10

4 System model **12**

4.1 Architecture 12

4.2 Theoretical model 14

4.3 Optimization 16

5 Numerical simulation **19**

5.1 Setup 19

5.2 Results 20

6 Conclusion **24**

List of Figures **25**

List of Tables **26**

Bibliography **26**

Abstract

The transformation of transport systems by connecting the vehicles into the network is due. Cooperative and automated driving is becoming a widely researched topic both by academia and industry, hoping that it can solve many issues we currently struggle with. Reducing the occurrence of traffic jams, helping vehicles to emit fewer greenhouse gases, and most importantly, preserving human lives may all be achievable by these advancements.

To enable this, vehicles must share information efficiently. This can be done by leveraging the aggregation of high-definition (HD) maps. These multi-layered maps contain static and dynamic information regarding a certain situation. To obtain a shared understanding between the vehicles without revealing the actual data, federated learning is introduced in our proposed system, because it preserves data privacy because the actual data is not traveling through the 5G network, only the local model parameters.

To realize this idealized system the good intentions of each participant are also required. With the current advancements in distributed ledger frameworks, introducing a consortium blockchain that tracks the reputation and trustworthiness of the participants and incentivizing them to show honest behavior is essential. To assess feasibility, the entire system's economic utility is modeled, optimized, and simulated.

Chapter 1

Introduction

Autonomous transportation is often hailed as one of the leading sectors that may cause the next industrial revolution. At its center, just as once the steam engine was, artificial intelligence is expected to be the nucleus. Considering recent developments, however, this prospect may not be as far down the road as it may seem. With the autonomous vehicle market already sitting at around 100 billion dollars in 2021, and with the prospective growth to 1.8 trillion dollars by 2030, the breakthrough is projected to happen in this decade. The main contributors and drivers of this phenomenon are found in Europe, East Asia, and North America, as they are all advancing toward Level 4 and 5 automation on the Society of Automotive Engineers (SAE) scale, mainly because of the willingness to increase innovation by loosening regulations and funding research project. The regulations are specifically innovation-friendly in California and Arizona, where companies such as Waymo but even Ford, and GM have all been running self-driving testing fleets.

The motivation behind the quick adoption of self-driving is that national and international (EU) regulatory bodies hope that it helps to reduce traffic jams, road fatalities, and also emissions. Simply because automated vehicles are programmed to always adhere to traffic rules and use advanced routing algorithms. The parallel revolution of electric vehicles is also helping to provide the possibility of attaining these goals. In the case of automated vehicles, however, there is an important additional presumption, namely, that unlocking the full potential and realizing the above-mentioned goals requires a wide coverage of cooperating highly autonomous vehicles.

To achieve autonomous transportation there is a consensus, even Tesla is starting to agree [1], that merely sensors and road-side equipment are not enough, rather there is a need for specialized maps that are made with autonomous vehicles in the focus. These are referred to as High Definition (HD) maps, which in contrast to today's SD maps contain centimeters-level road and lane information coupled with traffic rules and even with highly dynamic data like current traffic or weather. Such precision, although much welcomed, it mandates companies to employ a fleet of mapping stations (vehicles) constantly, which makes the building and validation really costly and time-consuming processes. Conclusively, to reduce these constant advancements are being made by both industry and academia.

Apart from building and validation the third crucial aspect, given that traffic circumstances are under constant change, is the automated delivery of the ever-updated HD maps. These include altered or closed roads, incidental traffic jams, or hundreds of other parameters. In this report, this problem is at the center and the goal is to provide an applicable, secure, and fast solution.

As the task is too immense and impossible for a single entity to manage and since transportation is among the basic needs of human life, causing numerous participants to be commuting at all times, crowdsensing can provide a sensible solution. To use this crowdsourced data to perform ML/DL efficiently and with respect to privacy, Google introduced federated learning in 2015 for next-word prediction in Android devices [2]. Since then FL has been adapted and researched in various fields. For HD map updates the usage of FL is also adequate, as the edge-cloud consists of vehicles and roadside equipment in a specific tile and they are running the learning process with the aggregator being the cloud or another edge device at a higher architectural level. Aggregated models can be transferred back, so the vehicle data remains private.

Nevertheless, the application of FL is not secure nor performant without measures to guarantee the goodwill and reliability of the participants. Otherwise, malicious actors may appear and hinder the learning process, thus causing the FL process never to conclude. To eliminate this threat it is sensible to introduce a reliability-managing blockchain in such a transactional and distributed system. By doing so the participants get associated with a score that corresponds with their truthfulness and reliability, so when the FL aggregation occurs, this grade is taken, and the contribution of the specific participant's contribution is weighted with it. This way it is made sure only truthful information is considered when aggregating the model.

For such a resource-demanding yet crowd-sourced data-based application, the economic implications are non-negligible. Since the materialistic i.e. energy costs that vehicles are paying to perform computations are high, the prospective rewards should be able to compensate the clients. To grant this and conclusively, to boost participation some incentive mechanisms must be involved. As blockchain technology has already been introduced to enable reliability tracking, its tasks can be broadened to incorporate client incentivization. This way, they are receiving both tangible (blockchain-backed rewards) and intangible benefits from using the system, causing their overall balance sheet to turn positive.

In this report, a framework with these key features is introduced. To briefly summarize, it runs on top of the CARLA simulator backend, as that provides the vehicular setting and the data to be used by the FL. The system then continuously runs the HD map creation which receives the data and learns from it by applying FL and using a Hyperledger Fabric-backed distributed ledger to correctly store the calculated reliability metrics and provide incentives. These are maintained and used in every iteration to prevent malicious intent. Upon the termination of the learning process, the vehicles build the HD map from the learned variables.

In the next chapters, first, the use-cases for the proposed system are introduced in Chapter 2, then the reviewed related work and existing solutions are presented in Chapter 3. Following this, the system model is discussed in Chapter 4. In the end, the results are presented and explained, in Chapter 5.

Chapter 2

Use case description

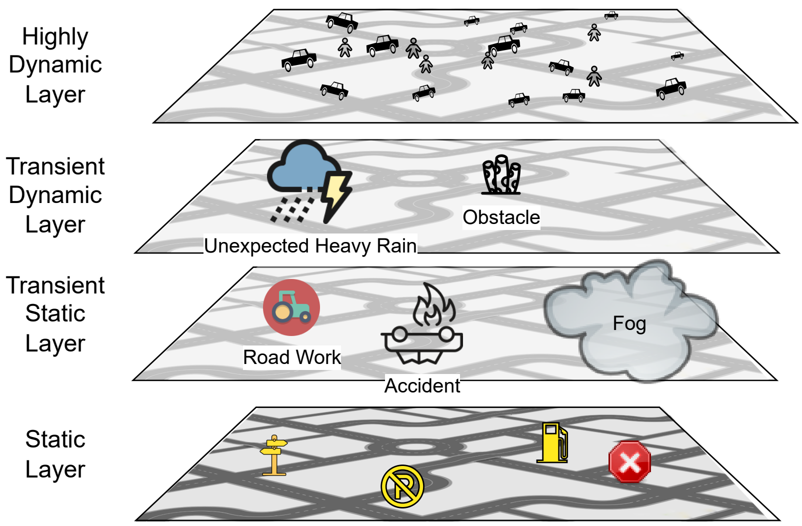


Figure 2.1: Representation of HD map consisting of different layers with static and dynamic information.

As we witness the gradual transition toward autonomous driving, the development of digital maps for self-driving vehicles becomes imperative, transcending their traditional role of aiding in navigation. Autonomous Vehicles (AVs) demand an exceptionally detailed understanding of their surroundings, with precision down to the centimeter level, to make critical decisions. To achieve this, these vehicles employ an array of sensors to gather real-time data about their environment, subsequently processing it to navigate autonomously. Yet, achieving real-time autonomous navigation, particularly in urban settings, poses a formidable challenge due to the limitations of sensor range and inherent inaccuracies. High Definition (HD) maps play a pivotal role in enabling autonomous vehicles to comprehend their environment with a high degree of accuracy and achieve precise localization [3]. This, in turn, empowers them to overcome the limitations of sensor visibility, exercise context awareness, and effectively interpret local traffic regulations to make safer decisions and proactively plan their routes.

HD maps, primarily designed for self-driving applications, excel in delivering the level of accuracy necessary for vehicles to navigate in real-time. The creation of HD maps typically involves specialized vehicles equipped with highly precise sensor equipment, including Differential GPS (D-GPS), a plethora of cameras, and exceptionally accurate laser scanners. These sensors capture information about current obstacles and the traffic regulations pertinent to the surrounding environment. HD maps, illustrated in Figure 2.1, comprise various layers: the static layer, housing infrequently changing information like road layouts; the transient static layer, containing details that remain constant for a set period, such as roadwork; the transient dynamic layer, which offers data subject to frequent changes; and the highly dynamic layers, delivering real-time data about the surrounding environment, including vulnerable road users.

Given that the road network is far from static and constantly influenced by factors such as traffic congestion, accidents, construction activities, and the status of adaptive traffic signs, it is imperative to keep these maps continuously updated to provide real-time information to autonomous vehicles. This necessitates the rapid delivery of updated HD maps with minimal latency, a challenge that can be mitigated through the implementation of edge computing. The concept of deterministic update intervals is advocated, and subsequent sections delve into the impact of shorter update cycles on the overall system.

Notably, HD maps are location-specific and entail the transmission of substantial data volumes. Vehicles within the same geographical area often request identical HD map data, referred to as a "tile," for autonomous driving. This recurrent transmission of large HD map datasets over the core network places significant stress on capacity-constrained backhaul links. To alleviate this burden and substantially reduce latency, a practical solution involves processing and storing HD maps on Multi-access Edge Computing (MEC) servers situated at the vehicular network edge, often referred to as Roadside Units (RSUs). This enables vehicles to access the required HD maps from their associated RSUs through vehicle-to-infrastructure (V2I) communication without the need to traverse the core network. Alternatively, the emerging 5G edge infrastructure, offering broadcast communication as of Release 16 [4], can be harnessed to provide the necessary computing capabilities for edge computing within close proximity to users.

Chapter 3

Related work

3.1 Building of HD maps

HD map creation faces a lot of challenges, due to its complicated structure and serious requirements. Mapping methods have been evolving, but full automation is still out of reach.

The authors of [5] give an overview of the current process and techniques, with the whole pipeline is depicted in Figure 3.1.

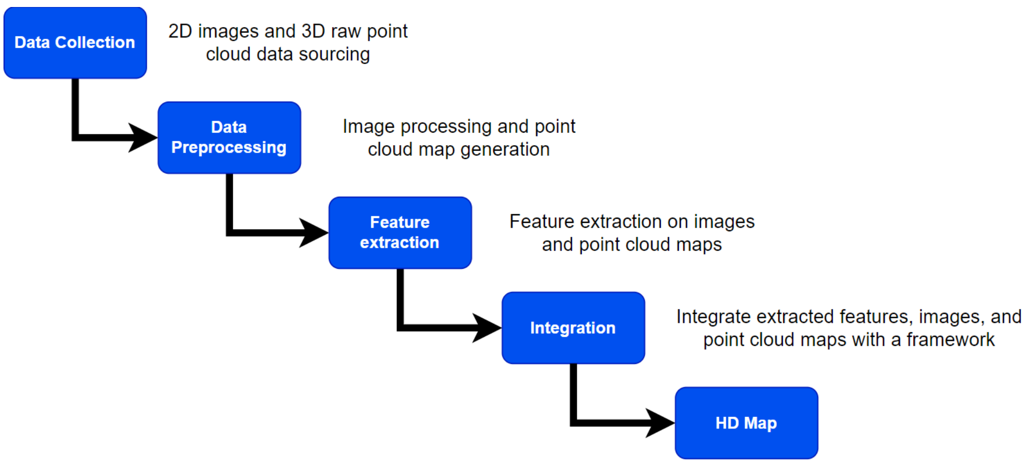


Figure 3.1: HD map generation steps

The first step in the process is data collection/sourcing. This is usually done by a Mobile Mapping System (MMS), depicted in Figure 3.2, which is a vehicle fitted with numerous sensors, including global navigation satellite system (GNSS), inertial measurements unit sensors (IMU), light detection and ranging (LIDAR), camera, and radar. These provide a huge amount of environmental, locational, and traffic information at the same time so a given section only has to be traversed once.

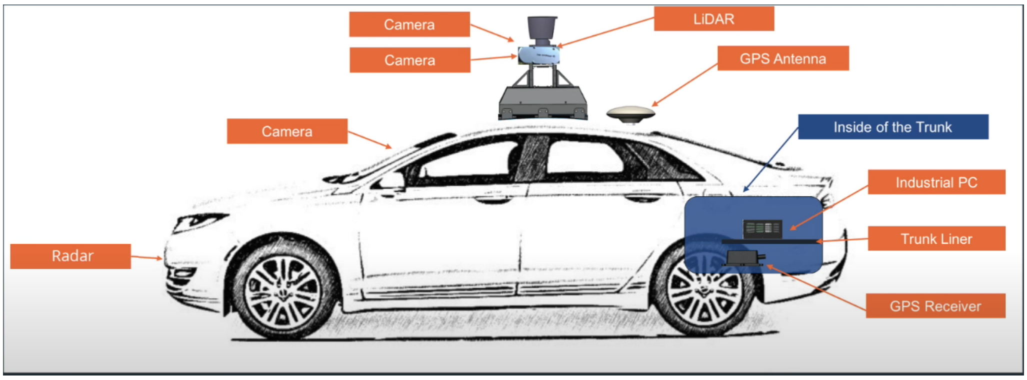


Figure 3.2: Mobile Mapping System used by Baidu

Therefore, most incumbent players use this method because they tend to have a fleet of MMS-es that can provide detailed data from large road sections. Researchers and academia, on the other hand, are unable to use this method, given the huge upfront costs of even one MMS, so, they focus on smaller areas like university campuses. Alternatively, they can also use publicly available, but detailed datasets like the KITTI or the Lyft dataset.

In the next step, the sensor data is fused into an initial point cloud map. These are accurate and precise but they have to get through a multi-step process, called *point cloud registration*, to get rid of noise and align some overlapping elements.

The last, and most time-consuming part to complete the HD map is feature extraction. Features like roads/lanes and their markings, and pole-like objects (traffic lights, signs) all need to be distinguished, validated, and loaded into specific layers. As these tasks make the essence of the HD map, it needs to be precise so manual work is indispensable as of now, but the emergence and adoption ML have shown some realignment toward semi-automated and automated solutions.

Paper [6] shows how Baidu produces its HD maps, which is very similar to the process described above. They initially gather vast data through their fleet of mapping vehicles, then these are processed by deep learning and computer vision algorithms to segment and classify the point clouds into separable objects. To verify the automated processes they employ manual verification trying to make sure that the final HD map is precise. Lastly, the maps are produced, released, and maintained by the company, so frequent updates are ensured.

As a specific field study, the authors of [7] and [8] provided a detailed description of the HD mapping of Taiwan and gave an overview on how to lower the costs of HD map production. Their proposed strategy is based on five pillars, which are: the usage of technical guidelines and standards in HD map production, flexible data acquisition, automated HD map format conversion tool, (semi)automated HD map production tool and finally the upscaling of HD maps. They argue that by achieving these, we can substantially lower the cost of HD map production and verification thus accelerating the adoption of autonomous vehicles.

In Taiwan, they followed these steps to produce HD maps on five routes in a research park by simulating crowdsourced data by using four third-party platforms. The data also kept track of the environmental changes they performed and has been published in an OpenDRIVE format. For further cost reduction, they examined that format conversion, which was sufficiently performant for road networks with the usage of the ASSURE map tool. They also tested their semi-automated production tool for evaluation. The results showed that, when compared to verified HD maps the overall network was consistent, however, the road edges and traffic lights had some modeling errors. Ultimately they conclude that these techniques helped them to lower the costs of HD map production from 1000000 NTD[1]/km to 350000 NTD/km.

3.2 Applications of FL in autonomous transportation

Given the data privacy conscientiousness and the network-preserving nature of federated learning, it has become a prime technique for performing different ML/AI-related tasks in the field of autonomous transportation. One of these use-cases, and perhaps the most prominent one, is the applicability of FL for HD map creation. As sensor and video data result in high volumes, its transportation would strain the available network bandwidth and ultimately also cause latency issues. In terms of privacy, sharing location or other sensitive information may also be undesirable for customers, which would understandably prevent their participation [9].

The authors in [10] first introduced a three-layered system, constituting the cloud, edge, and vehicular layers, and provided a system that employs FL for dynamic map fusion. The role of the central server is to provide initial parameters, and pre-trained models, if necessary. After receiving these the edge vehicles are continuously performing local fusion on their image and point cloud information and upload the weights to the edge server. This then performs a three-stage fusion by first partitioning objects in local maps, then generating them onto the global map, and lastly, it removes the overlapping ones. The result is then sent back to the vehicles.

[1] New Taiwanese Dollar

To expand the work to more complete HD maps, the same authors improved the system, and in [11] they provided a complex FLCAV solution. They identified multiple tasks like object and road detection, weather and sign classification, or trajectory prediction and built a framework that uses the same approach as the previous article. The difference is that here, during the local FL training, actual DNNs are utilized. They also consider the resource implications of the proposed system. The framework is built on top of, and trained in CARLA [12] (see later in section 5.1 but validated on a real-world dataset. Here it achieved roughly 58% accuracy for object detection, which means that some additional transfer learning and real-world testing are indispensable.

The authors of [13] focus on energy-efficient map data distribution. They propose an algorithm that minimizes the power consumption while receiving HD maps. An RSU serves a vehicle only if the energy required to receive data from the RSU and for basic movement is less than the remaining energy of the vehicle. For such vehicles, the data is divided among all the RSUs in proportion to the received power to provide the service. The authors in [14] propose a joint power control and spectrum assignment policy to maximize the sum data rate of the overall network for HD map dissemination. They study the interference effect on data transmission and formulate a model that describes the interference control problem in V2X-enabled HD map dissemination. The authors suggest cooperative delivery of HD maps through V2I and V2V communications by dividing the HD map into many data blocks based on data volume and infrastructure environment. The work in [15] discusses HD map caching for autonomous driving in vehicular networks with unknown vehicle requests and trajectories. The proposed caching algorithm defines a reward function that considers the historical tile request data.

A new architecture is proposed in [16] that combines Multi-access Edge Computing (MEC), and Software Defined Networking (SDN) to enable HD map-assisted autonomous driving. A two-tier server structure is proposed with cloud and MEC servers to achieve resource utilization and network scalability. The applications and services are deployed on the MEC server using Network Function Virtualization (NFV) at the network edge. A MEC system framework is proposed in [17] for HD map applications. The authors introduce the application mode, functional modules, HD map data distribution workflow, and communication process between the autonomous vehicle client and the server.

In terms of system applicability and dimensioning, the authors in [18] did widespread simulations using a multi-level architecture, similar to the one in [10]. They also utilized a queuing theory-based approach to model the changing (mobility-influenced) demands. In their findings, an HD map building FL system is most optimal when using 1 km^2 tile areas with as high bandwidth as possible and as low trigger time, which they defined as the time interval that an HD map layers gets refreshed, as possible.

3.3 Blockchain-based reliability and incentive mechanism

The authors in [19] introduce a complex system, where a network task publishers in a federated learning scheme are tracking the reputation of the clients in a dedicated blockchain. These metrics are then used for node selection in successive iterations to exclude malicious clients. They are calculated by the current task publisher based on the computation time and responsiveness of the client. Following the initial calculation, poisoning detection schemes like RONI or FoolsGold [20] are also executed to get a more exact value. This *local reputation vector* is then combined with other task publishers' historical data and as a result, a composite reputation value is reached. This is then stored in the blockchain and also used for upcoming worker selections in future FL iterations. The main shortcoming of this model is that a latency analysis is not given, so the overall applicability in a highly dynamic scenario, like HD map updating, is not reviewed.

In a slightly different approach, the authors of [21] built a system with general IoT devices as participants of an FL scheme. Here the manufacturers submit their models to a blockchain, which is then used to keep the model up to date as the FL is being executed. When the local models are trained and the weights are sent, they propose a novel consensus mechanism for miners to validate the aggregated model and calculate the reputation of the customers. If a local update is deemed to be in the top majority then it is accepted and the corresponding customers' reputation is increased and also recorded in the blockchain otherwise it is decreased. Once every update is validated the aggregated model in committed to the blockchain. The main drawback of this approach is that in a mobile HD map maintaining setting, the models of a given tile should not be stored on a global, or on an otherwise wide-scale blockchain as it creates overhead. Instead, as the traditional FL suggests, the weights are propagated upwards to get better-fitting models. The complicated consensus mechanism poses another drawback as it would delay the execution of the FL process decisively.

For providing incentivization in a federated setting the authors in [22] propose a system that is based on contract theory so that mobile users are compensated for joining a system. Here they provide detailed theoretical descriptions based on proofs for their mechanism: the utilities for both the clients and the system are thoroughly introduced and optimized. They also show that this approach can result in improved learning accuracy. This solution, however, is not directly applicable to the defined use case, because here the client and system utilities are different due to the system building on FL being for the overall betterment of all the participants. Therefore, there are non-materialistic benefits for the clients to remain engaged, i.e. they receive the contribution of others when creating an HD map.

Chapter 4

System model

In this section, I propose a comprehensive model that introduces both the architectural aspects and its economic considerations. These jointly describe the systems' thorough operation and protocols, while also mathematically analyzing the introduced incentivization mechanism to model economic feasibility. In order to ascertain real-life applicability, some realistic bounds and constraints are also considered while analyzing the output of the numeric optimization, enabling the formulation of some guidelines and best practices for creating such federated-networked system with optimized performance.

4.1 Architecture

Designing a comprehensive and feasible navigation system that helps autonomous transportation involves solving numerous core requirements. These can be separated into economic and technical issues, namely that the system should provide useful, highly precise, and actual information in the face of High Definition (HD) maps that encompass static, like road geometry, and dynamic data, like weather or other commuting parties, respectively. Given the excessive variability and size of the data that would be needed for such a task, user participation is beneficial and desired because a centralized solution is impractical and supposedly infeasible. The traffic infrastructure users could collaborate in a decentralized fashion to provide data on their surroundings and so aggregated HD maps could be constructed. For this construction technique, however, providing data privacy and security is indispensable, along with the assuredness that the user behavior is honest and so only valid and useful data is taken into account.

The system I am proposing, therefore, aims to fulfill the above-mentioned objectives and requirements by introducing several state-of-the-art technologies that work together to ensure them. The architecture of the system, on the physical level, has two main actor groups: the users, who are considering the use-case, vehicles, and the operators, edge RSUs, and the central servers in the cloud. These are, on a logical level, complemented by two consortium blockchain networks, responsible for reliability and user rewards. This conceptual architecture is presented in Figure 4.1. Here, the designed operation and protocols can also be examined.

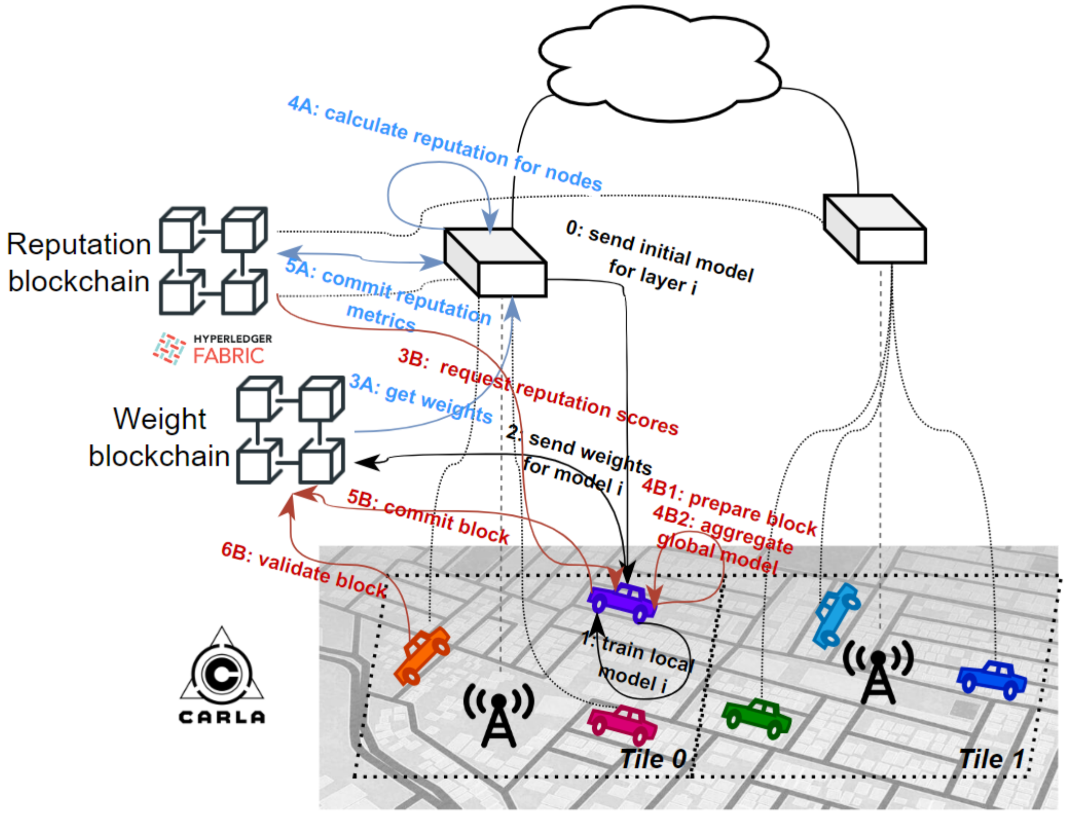


Figure 4.1: Logical system architecture

For producing an HD map tile in a given area, the actor groups perform a Federated Learning scheme with different frequencies for different HD map layers. The RSUs and the central server first distribute the initial models to the users present at the tile (0). The users then train the models on their local data (1) and upload the weights of these to the weights blockchain (2). The RSUs fetch these transactions (3A) to calculate the reputation scores of each participant (4A) and commit these to a separate reputation blockchain (5A). The users then fetch the scores (3B), and the user with the highest reputation gets to prepare (4B) and commit (5B) the block containing the weights transactions while earning a block reward. Naturally, the block committer must eliminate low-reputation transactions to keep good faith. After committing the block, the other users validate the block to keep the block committer in check, and to produce the aggregated global model for themselves, from the committed transactions.

Operating in the described fashion ensures data privacy by employing Federated Learning so that the local data is never shared or exposed, only the weights of the local model are broadcast. Other decisive requirements, such as data security and system usefulness are guaranteed by the fact that the system only allows transactions from reputable nodes to be added to the ledger. This way the updates are carrying the results of the actually performed computations so these are strongly expected to be beneficial for the system. Finally, the most neglected aspect in other works, the economic feasibility of the system, is solved by incentivizing users to contribute. Namely, that the system encourages users to achieve a high reputation, i.e., to put in more work, so they can receive rewards from block commits.

4.2 Theoretical model

Every system that expects user activity must provide some advantage of a certain nature, e.g.: entertainment, information, easement of some process, or even financial, to reward or compensate for their efforts. This notion is most understated if the user "suffers" high costs for participating or operating the system. Therefore, the compensation should be manifold to balance these out and encourage honest and symbiotic utilization of such a system.

Hence the proposed system demands image recognition model training for local models, there is a high expected cost from participants. These mostly materialize in higher energy needs to power CPUs or GPUs for training, so there is a decreased fuel efficiency. The benefits, henceforth, are required to outweigh these costs making the system effective and efficient for both actor groups. In other words, modeling the economic feasibility is indispensable to acquire a sense of applicability in reality.

To model the utilities of each actor group we define these as the difference of the incomes and the costs, as can be seen in Equations 4.1 and 4.2. It is important to emphasize that the model considers a single iteration on a single tile, thus it measures a single local iteration. Moreover, in our model, we factor in the energy used to achieve a given reputation instead of the concrete reputation value, due to the more materialistic nature of the energy that is useful in economic evaluations. These two metrics, however, are expected to be closely correlated so by spending more energy on local calculations the achieved reputation grows higher.

Table 4.1: Mathematical notations used in the system model.

Notation	Description
n	Number of clients
s	Subscription fee
r	Block reward
ν	Vehicle density coefficient
$\mathbf{E}[X]$	Expected value of required energy for reputation (E_{cmp})
m	Maximum value of E_{cmp}
ξ	Effective capacitance coefficient
c_n	Number of CPU cycles using a single data sample by $user_n$
d_n	Size of local data samples for $user_n$
f_n	CPU frequency of $user_n$
α	Energy cost coefficient

In the case of the clients, the materialistic income, the first term, consists of the expected reward, considering all vehicles in the tile. It is calculated, as seen in Equation 4.5, by taking the probability of a vehicle having spent the maximum energy on achieving a high reputation among its peers and multiplying it with the attained block reward. The next two terms are the suffered costs, in terms of computational energy, detailed in Equation 4.4 [22] and subscription fee. The last term is an abstract income, referred to as perceived value, that tries to capture the beneficial character of the system from the users' perspective. Specifically, it captures a logarithmic growth in relation to the total spent computational energy by the users. The logarithmic function is a simpler choice to replicate the notion that having a growing number of spent computational energy is analogous to having a curve with a gradually deteriorating, but positive gradient.

$$U_c = f_Y(y)r - \alpha\mathbf{E}[X] - s + \alpha\ln(n\mathbf{E}[X]) \tag{4.1}$$

Compared to clients, the system has a more straightforward model. Here, the incomes are the subscription fees from all participating users in the given tile, while the cost is simply the block reward that the most reputable vehicle receives. The number of users, Equation 4.3, is central to both functions, it depends on the vehicle density and the current subscription fee. The function has a radical hyperbolic curve characteristic in terms of the subscription fee, representing the lower willingness to participate as the fee increases. This feature is analogous to the demand curve in economics.

$$U_s = ns - r \tag{4.2}$$

$$n = \lfloor \frac{\nu}{\sqrt{s}} \rfloor \tag{4.3}$$

$$E_{cmp} = \xi c_n d_n f_n^2 \tag{4.4}$$

$$\begin{gathered}
X_1 \ldots X_n - E_{cmp} \text{ of each vehicle on given tile } X \sim U[0,m] \\
Y - \max\{X_1 \ldots X_n\}; \quad Y \sim U[0,m] \\
P(Y < y) = P(X_1 < y, \ldots, X_n < y) = \Pi_{i=1}^{n} P(X_i < y) = (\tfrac{y}{m})^n \\
f_Y(y) = \tfrac{\mathrm{d}}{\mathrm{dy}}(\tfrac{y}{m})^n = \tfrac{n}{m^n} y^{n-1}
\end{gathered} \tag{4.5}$$

To make the system applicable there are certain constraints, C_{1-4} that are required. Firstly, both of the actor groups must have a positive utility, otherwise, it is economically damaging to participate in or operate the system. Secondly, in strong correlation to the previous constraints, the number of users is required to be a natural positive number. Lastly, the CPU cycles on a single data sample is maximum the CPU frequency enables under a given time period.

$$\begin{aligned}
&(C_1) : U_c > 0 \\
&(C_2) : U_s > 0 \\
&(C_3) : n \in \mathbb{N}^+ \\
&(C_4) : c_n \leq 1f \ for \ 1 \ sec \ timeframe
\end{aligned} \tag{4.6}$$

4.3 Optimization

To ensure model usefulness it is essential to gain the optimal setting under which the system is materialistically profitable for the system operator and at the same time overall beneficial to the clients. The first criterion stems from the fact that the system needs to make a profit in order to ensure investments for continuous operation, maintenance, and further development. The sensible clients, on the other hand, are anticipated to participate in a commonly beneficial system that proves to be also rewarding individualistically. Therefore, the system utility function is maximized as long as the client utility function is non-negative.

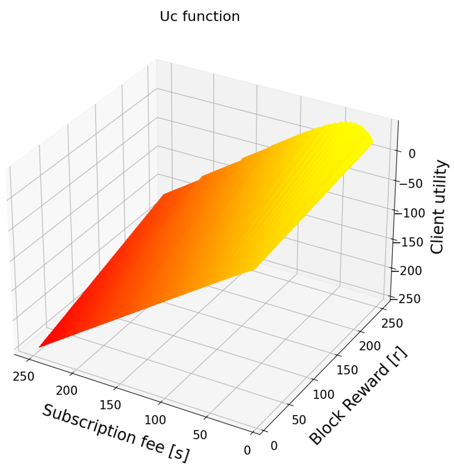
Plotting the functions while using common parameters for vehicle density ($\sim$ 100 $vehicles/km^2$ [18]) and energy consumption cost (taking the cost of Tesla's AMD Ryzen APU as reference [23]) shows the surfaces of Figure 4.2. It is easy to observe, given the derivatives, which taken from the smoothed surface (without the lower integer part function), in Equation 4.7, that the client utility is growing as $\lim_{r\to\infty} U_c$, but diminishing as $\lim_{s\to\infty} U_c$, given the C_{1-4} constraints. On the other hand, the system utility derivatives, seen in the Equations 4.8, have a somewhat inverse face. The system utility grows as $\lim_{s\to\infty} U_s$, but decreases with a constant degree when $\lim_{s\to\infty} U_c$. Ultimately, the derivatives of the defined functions only take zero when some input parameters are zero, conclusively the functions reach their extremes at the edge of the defined domain.

$$\begin{gathered} \frac{\partial}{\partial s}U_c = \frac{\nu r m^{\frac{-\nu}{\sqrt{s}}} x^{\frac{\nu}{\sqrt{s}}-1}(\nu \log m - \nu \log x - \sqrt{s}) - s(2s+\alpha)}{2s^2} \\ \frac{\partial}{\partial r}U_c = \frac{\nu m^{\frac{-\nu}{\sqrt{s}}} x^{\frac{\nu}{\sqrt{s}}-1}}{\sqrt{s}} \end{gathered} \tag{4.7}$$

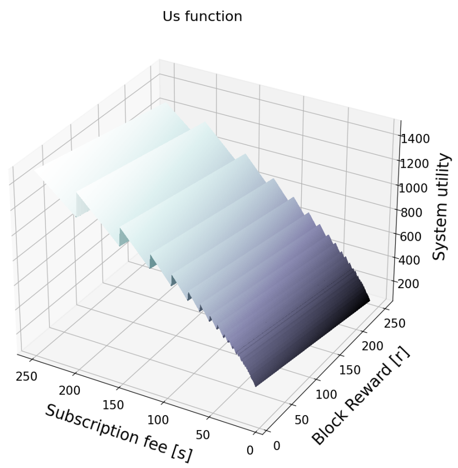
$$\begin{aligned} \frac{\partial}{\partial s}U_s &= \frac{\nu}{2\sqrt{s}} \\ \frac{\partial}{\partial r}U_s &= -1 \end{aligned} \tag{4.8}$$

The joint utility function shows the optimal setting with regards to block reward and subscription fee, as seen in Figure 4.2c. Here, the darker surface shows the system utility, which is being maximized, while the yellow one is the client utility where it is above zero, making joining sensible. The remaining part of the system utility function is cropped due to being economically unviable. As it is evidently seen, with a fixed number of potential clients the plot exhibits direct proportionality, so when s grows r likewise needs to grow to attain max utility. If that were not the case, the users would be discouraged from joining the system, as equation 4.3 shows. Keeping this constraint shows that the system has a maximum utility when r and accordingly s is the biggest.

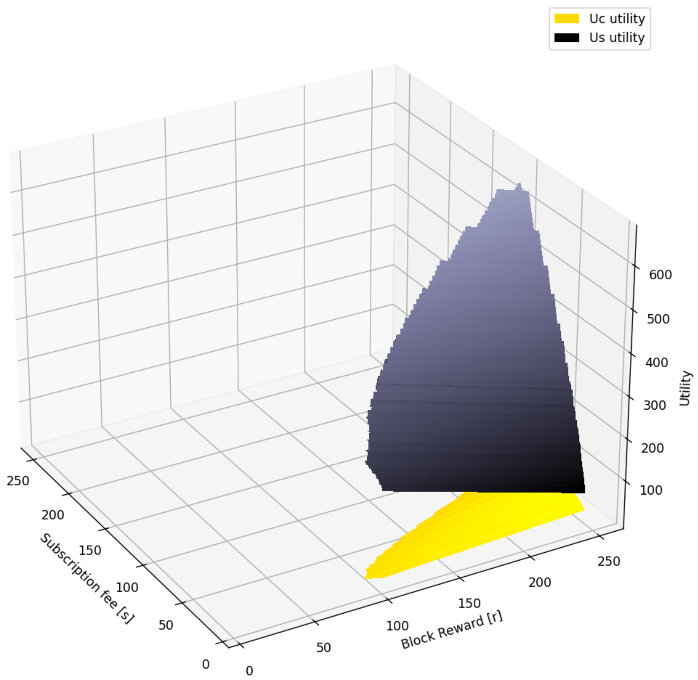
It is important to note that the vehicle density can be highly varying in different areas, therefore both economic quantities need to be set dynamically. Making operators implement a dynamic pricing model based on actual characteristics. Pricing strategy moreover, is not solely decided based on the mentioned metrics, but there are further socioeconomic considerations. In other words, even if the theoretical model shows that the highest utility is achieved as long as the block reward and the subscription fee are mutually expanded, customers have price sensitivity. Conclusively, this may make having a high subsription fee unfeasible, even if the client utility function is positive.



(a) Client utility function



(b) System utility function



(c) Joint utility function

Figure 4.2: Utility functions with common parameters

Chapter 5

Numerical simulation

To gather evidence on the actual system proceedings, performance, and ultimately on the real-world applicability of the proposed system, thorough simulations were conducted. Here, the theoretical variables are substituted with some values taken from a range that could occur in reality, and the utility values are evaluated against these. Finally, there are clear showings of how the system would perform under various settings.

5.1 Setup

The simulation framework that is used to numerically evaluate the designed system, is multi-tiered, as its structure can be observed in Figure 4.1. At the core, the industry-standard tool, CARLA is utilized for holistically simulating every component of a vehicular environment. This encompasses vehicles, their sensors, and other actors, such as pedestrians and micro-mobility users. To control these possibly large-scale multi-actor scenarios, CARLA provides an interface and an API, so simulations can be controlled externally. In this fashion, the controller application can gather information regarding the simulation, including the position and status of the vehicles, sensor data, and infrastructure details, at every step. These data produce the local data for each vehicle to be learned via FL.

To carry out the basic processes of an FL scheme, the custom framework utilizes the CarlaFLCAV project [24]. This can provide, among other features, out-of-the-box dataset generation and labeling, and FL using the YOLOv5 object-detection model. The latter was taken as the starter, but it needed to be adapted to perform the proposed protocol. This required changing the implementation of the cloud aggregation from simple averaging to the reputation-calculating Fool's Gold algorithm and extending the FL to store its data, both weights and reputation on the corresponding distributed ledgers. The rewarding procedure was also added, so at each iteration, the maintained balance of the participants changed according to their current reputation. Other than these adjustments the FL process was unmodified.

The distributed ledgers are managed by the Hyperledger Fabric [25], which provides a consortium blockchain network. Essentially here the network members are trusted, identified parties who can access, modify, and commit to the ledger. In the proposed system these parties are operator companies or different national or governmental organizations. In the simulation framework, a basic network was running with two organizations, maintaining two different channels (these can be considered as separate ledgers in Hyperledger). One of these is the weights channel, which receives the weight transactions from the vehicles, while the other is the reputation one, which keeps track of the balance of each vehicle as well as its reputation.

The simulations were run on a machine with a dedicated NVIDIA A100 GPU and an AMD Epyc-Rome 16-core CPU in the cloud of HUN-REN Research Network.

5.2 Results

To corroborate the theoretical optimization, simulations were carried out. Most importantly the energy-related costs were researched, by measuring the CPU and GPU usage on the server, in 1-minute intervals, during the simulations to answer how these directly affect them. Meanwhile, the FL-based object-detection simulations, as previously thoroughly introduced, were performed by the custom simulation framework under different configurations. These can be browsed in Table 5.1.

In the test cases value sets were chosen with careful consideration. As the number of clients, and with a strong correlation, the number of FL iterations have a tremendous impact on the simulation time, they were kept within reasonable bounds. Specifically, when 5 clients were simulated the iteration count was set to 10, but during the other simulation, the ratio was inverted. The subscription fee and block reward, conversely, traversed a wide logarithmic scale to get a correct measurement of how the actual resource usage compared to them. Finally, there were some taken to see how the energy price coefficient influences the utility functions, if it deviates from the current average.

The plots contain the balance at each point in the linear space defined by the set of values of the block reward and subscription fee. These data points are the sum of Equation 4.1, but now with simulated results. The first component is the

Table 5.1: Variable settings for test cases

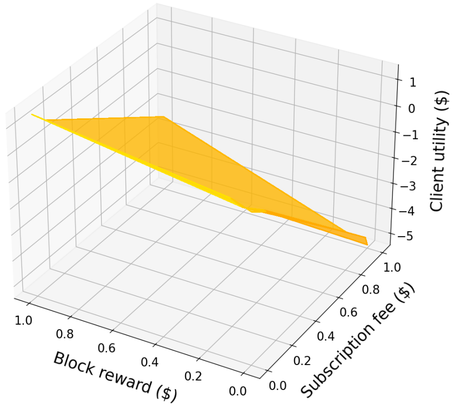
Notation	Description	Set of values
n	Number of clients	(5, 10)
s	Subscription fee [$]	(0.01, 0.1, 1)
r	Block reward [$]	(0, 0.01, 0.1, 1)
α	Energy coefficient $[\frac{kWh}{\$}]$	(0.16)

vehicle balances throughout the iterations, $r - s$, for a given setting of r and s. The second component is the average of the energy cost of physical server usage (CPU and GPU), while the third, perceived value, is the logarithm of the second term times the number of vehicles.

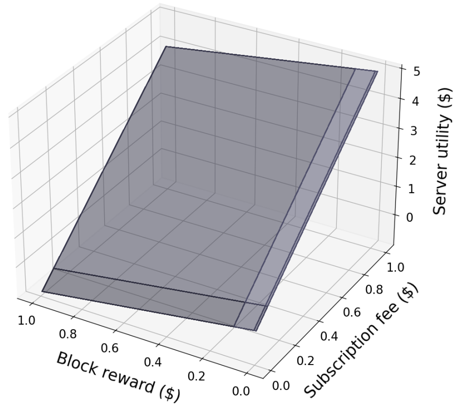
The simulation results for the lower client count are depicted in Figure 5.1. Both the client and server utilities show the expected behavior. The prior grows as the block reward increases, but heavily decreases as the subscription fee is upped. It is easily noticeable how quickly the utility deteriorates as s grows, making the usage of the system excessively unprofitable. Hence, having block reward is indispensable. Moving to the latter, its behavior is inverse, growing in s and receding in r. It is conspicuous, although, that here the growth of s is linear, not radical as seen before in Figure 4.2b, because the number of vehicles in the simulations was not expressed as a function of s, rather it was fixed.

In terms of the joint plotting of the utility functions it is observable that the slopes for U_c in s and U_s in c are different, making the optimal U_s a trapezoid where U_c is non-negative. The optimal utility could be found at the edge of the block reward domain along the upper edge of the trapezoid, with an assigned subscription fee, that is marked by the $\frac{\partial}{\partial r}U_c$ sloped line. This result strongly correlates to the findings in Section 4.3. It is important to notice, however, that here the α parameter was kept at the base setting, i.e., US average price of 1 kWh for EVs [26], and this was within the same order of magnitude as the block reward and subscription fee.

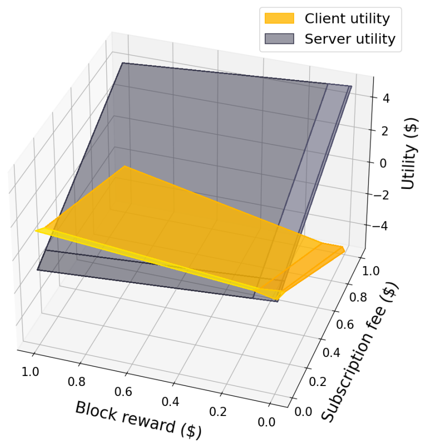
To visualize the effect of change in the number of clients, the utility functions were plotted alongside each other in Figure 5.2. In the first subfigure, showing the client utility, the effect of the only non-materialistic value contained in the equations, the perceived value, can evidently be inspected, as this quantity depends most directly on the number of participants. The purple surface denoting the U_c of the 10 client setting is much flatter. Additionally, even at the most unprofitable setting, where r is zero, it has a much higher utility value, although also negative, than the 5-client setting. Given the flatness and overall higher values, a larger portion of the utility function becomes non-negative upon client increase, and this



(a) Client utility function for 5 clients



(b) System utility function for 5 clients



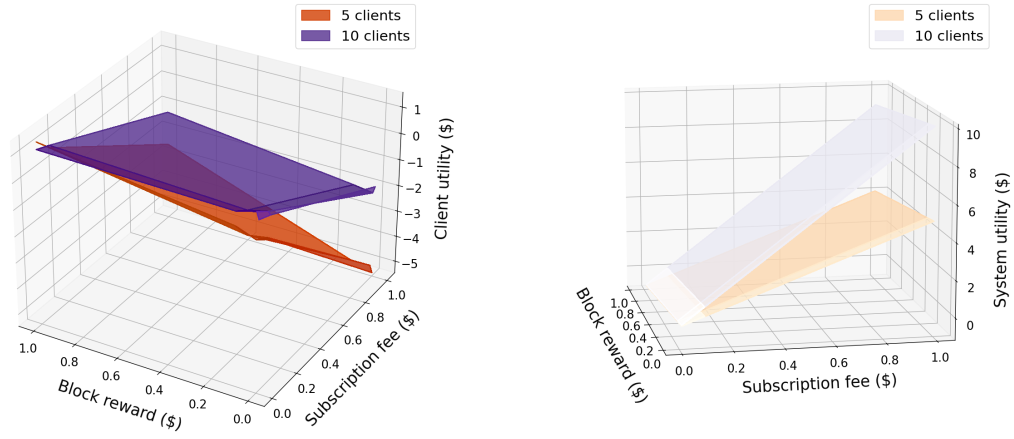
(c) Joint utility function

Figure 5.1: Utility functions for 5 clients

implies that the system also gets more optimization opportunities to maximize profit, making a client number dependent dynamic pricing approach possible.

As for the second subfigure, the impact is even more obvious as the utility linearly depends on the ns product. The plot clearly shows this increment in slope with regard to the subscription fee. Therefore, the utilities for the same $r-s$ settings show the same proportionality as the number of clients.

For large-scale simulations, some selected test cases were run with a high, 50 participating client number, to evaluate the growth of perceived value. The results are illustrated in Figure 5.3. The trend is clearly visible that as the number of clients



(a) Client utility **(b)** System utility

Figure 5.2: Comparing utility functions between 5-client and 10-client settings

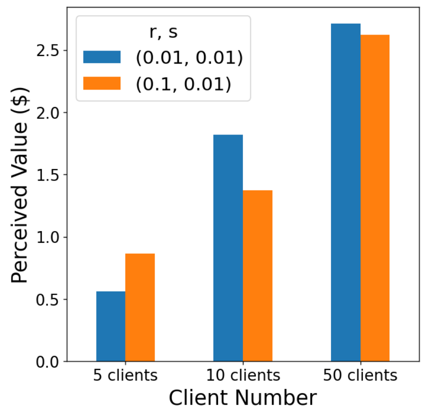


Figure 5.3: The growth of perceived value with regards to client number

increases the perceived value is moving accordingly, although the rate is different. The deviation between the test cases is likely due to some system utilization anomaly, as it is directly not dependent on r.

Ultimately the acquired results reassure the initial conceptual assumptions and expectations made during the architectural and theoretical modeling when running the system on an industry-standard simulator with the physical resource usage taken into account.

Chapter 6

Conclusion

In this work, I present a reliable and economical information aggregation and sharing system for vehicular use-cases. Specifically, the construction of HD maps that provide real-time information regarding various aspects of the environment of transportation, from road geometry to dynamic traffic data. The proposed system uses a reliable federated learning scheme, aided by blockchain, to aggregate information from trusted reputable sources only. To complete the system and encourage honest user participation, an incentivization protocol is developed boosting system usefulness for all parties.

Apart from the comprehensive modeling of the system, a thorough economic analysis is also provided including the optimization of both system and client utilities. The theoretical claims are corroborated by numerical simulations based on industry-standard tools, to ensure theoretical, physical, and economic feasibility. The conclusion is that, as long as the utility of the clients is positive, the system is highly profitable by giving out large rewards as the clients are compelled to join and earn both material and immaterial benefits.

List of Figures

2.1 Representation of HD map consisting of different layers with static and dynamic information. 4

3.1 HD map generation steps 6

3.2 Mobile Mapping System used by Baidu 7

4.1 Logical system architecture 13

4.2 Utility functions with common parameters 18

5.1 Utility functions for 5 clients 22

5.2 Comparing utility functions between 5-client and 10-client settings . . 23

5.3 The growth of perceived value with regards to client number 23

List of Tables

4.1 Mathematical notations used in the system model. 15

5.1 Variable settings for test cases 21

Bibliography

[1] Autopilot review. Tesla autopilot using hd maps?, 2022. URL https://www.autopilotreview.com/tesla-autopilot-hd-maps/.

[2] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning, 2021.

[3] Rong Liu, Jinling Wang, and Bingqi Zhang. High definition map for automated driving: Overview and analysis. *Journal of Navigation*, 73(2):324–341, 2020. DOI: 10.1017/S0373463319000638.

[4] Ericsson. https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/5g-nr-evolution, 2020. Accessed: 2022-10-21.

[5] Zhibin Bao, Sabir Hossain, Haoxiang Lang, and Xianke Lin. A review of high-definition map creation methods for autonomous driving. 122:106125. ISSN 0952-1976. DOI: 10.1016/j.engappai.2023.106125. URL https://www.sciencedirect.com/science/article/pii/S0952197623003093.

[6] Hao WEI. How baidu apollo builds HD (high-definition) maps for autonomous vehicles. URL https://towardsdatascience.com/how-baidu-apollo-builds-hd-high-definition-maps-for-autonomous-vehicles-16

[7] Kai-Wei Chiang, Jhih-Cing Zeng, Meng-Lun Tsai, Hatem Darweesh, Pin-Xu Chen, and Chi-Kuei Wang. Bending the curve of HD maps production for autonomous vehicle applications in taiwan. 15:8346–8359, . ISSN 2151-1535. DOI: 10.1109/JSTARS.2022.3204306.

[8] Kai-Wei Chiang, Chi-Kuei Wang, Jung-Hong Hong, Pei-Ling Li, Chin-Sung Yang, Meng-Lun Tsai, Jeffrey Lee, and Sean Lin. Verification and validation procedure for high-definition maps in taiwan. 1(1):18, . ISSN 2731-6963. DOI: 10.1007/s44212-022-00014-0. URL https://doi.org/10.1007/s44212-022-00014-0.

[9] László Toka, Márk Konrád, István Pelle, Balázs Sonkoly, Marcell Szabó, Bhavishya Sharma, Shashwat Kumar, Madhuri Annavazzala, Sree Teja Deekshitula, and A Antony Franklin. Federated learning for vehicular coordination use cases. In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6, 2023. DOI: 10.1109/NOMS56928.2023.10154405.

[10] Zijian Zhang, Shuai Wang, Yuncong Hong, Liangkai Zhou, and Qi Hao. Distributed dynamic map fusion via federated learning for intelligent networked vehicles. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 953–959. IEEE. ISBN 9781728190778. DOI: 10.1109/ICRA48506.2021.9561612. URL https://ieeexplore.ieee.org/document/9561612/.

[11] Shuai Wang, Chengyang Li, Derrick Wing Kwan Ng, Yonina C. Eldar, H. Vincent Poor, Qi Hao, and Chengzhong Xu. Federated deep learning meets autonomous vehicle perception: Design and verification. URL http://arxiv.org/abs/2206.01748.

[12] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[13] Jinliang Xie, Jie Tang, and Shaoshan Liu. An energy-efficient high definition map data distribution mechanism for autonomous driving. *arXiv preprint arXiv:2010.05233*, 2020.

[14] Xinze Wu, Xiangming Wen, Luhan Wang, Wei Zheng, Zhaoming Lu, and Luning Liu. A cooperated approach between v2i and v2v for high definition map dissemination in automated driving. In *2020 XXXIIIrd General Assembly and Scientific Symposium of the International Union of Radio Science*, pages 1–4. IEEE, 2020.

[15] Xianzhe Xu, Shuai Gao, and Meixia Tao. Distributed online caching for high-definition maps in autonomous driving systems. *IEEE Wireless Communications Letters*, 10(7):1390–1394, 2021.

[16] Haixia Peng, Qiang Ye, and Xuemin Sherman Shen. Sdn-based resource management for autonomous vehicular networks: A multi-access edge computing approach. *IEEE Wireless Communications*, 26(4):156–162, 2019.

[17] Rongbo Zhang and Kaiyu Cai. The application of edge computing in high-definition maps distribution. In *Proceedings of the 2020 The 2nd World Symposium on Software Engineering*, pages 116–121, 2020.

[18] László Toka, Márk Konrad, István Pelle, Balázs Sonkoly, Marcell Szabó, Bhavishya Sharma, Shashwat Kumar, Madhuri Annavazzala, Sree Teja Deekshitula, and A. Antony Franklin. 5g on the roads: Latency-optimized federated analytics in the vehicular edge. *IEEE Access*, 11:81737–81752, 2023. DOI: `10.1109/ACCESS.2023.3301330`.

[19] Jiawen Kang, Zehui Xiong, Dusit Niyato, Yuze Zou, Yang Zhang, and Mohsen Guizani. Reliable federated learning for mobile networks. 27(2):72–80. ISSN 1536-1284, 1558-0687. DOI: `10.1109/MWC.001.1900119`. URL `https://ieeexplore.ieee.org/document/8994206/`.

[20] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. The Limitations of Federated Learning in Sybil Settings. In *Symposium on Research in Attacks, Intrusion, and Defenses*, RAID, 2020.

[21] Yang Zhao, Jun Zhao, Linshan Jiang, Rui Tan, Dusit Niyato, Zengxiang Li, Lingjuan Lyu, and Yingbo Liu. Privacy-preserving blockchain-based federated learning for IoT devices. 8(3):1817–1829. ISSN 2327-4662, 2372-2541. DOI: `10.1109/JIOT.2020.3017377`. URL `https://ieeexplore.ieee.org/document/9170559/`.

[22] Jiawen Kang, Zehui Xiong, Dusit Niyato, Han Yu, Ying-Chang Liang, and Dong In Kim. Incentive design for efficient federated learning in mobile networks: A contract theory approach. In *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, pages 1–5, 2019. DOI: `10.1109/VTS-APWCS.2019.8851649`.

[23] Evannex. Tesla's new AMD ryzen APU vs old intel atom GPU: How do they compare? URL `https://insideevs.com/news/588007/tesla-showdown-old-intel-gpu-vs-new-amd-apu/`.

[24] Shuai Wang, Chengyang Li, Derrick Wing Kwan Ng, Yonina C. Eldar, H. Vincent Poor, Qi Hao, and Chengzhong Xu. Federated deep learning meets autonomous vehicle perception: Design and verification. *IEEE Network*, 2022. DOI: 10.1109/MNET.104.2100403.

[25] Hyperledger Fabric. Introduction, 2023. URL https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html.

[26] US Department of Energy. The cost to charge an electric vehicle explained. URL https://www.energy.gov/energysaver/cost-charge-electric-vehicle-explained.