OBSERVE: Blockchain-based Zero Trust Security Protocol for Connected and Autonomous Vehicles (CAVs) Data Using Simple Machine Learning

Bo Sullivan and Junaid Ahmed Khan
Western Washington University, WA, USA
sulliv81@wwu.edu, junaid.khan@wwu.edu

***Abstract*—Connected and Autonomous Vehicle (CAVs) can proactively share future trajectories with each other and coordinate for safe navigation and efficient route planning. However, for a vehicle to rely on a nearby vehicle's trajectory data for real-time navigation decisions, it needs to trust data shared by neighboring vehicles. There exist no such distributed method to do so, and therefore, we propose OBSERVE, a lightweight trust model for nearby vehicles to endorse each others' trajectories in real-time using blockchain. Vehicles self-organize to endorse each others trajectories through consensus, where, each vehicle verify nearby vehicles trajectory by predicting the corresponding trajectory, compare results with peers, and reach consensus on its truthfulness. We employ simple machine learning for vehicles to predict neighbors trajectories instead of the computationally heavy vision based prediction algorithms, thus leveraging timestamped coordinates towards an energy efficient endorsement. OBSERVE is validated on realistic data and has shown to achieve higher prediction accuracy with lower computation overhead.**

***Index Terms*—Machine Learning, Blockchain, Connected and Autonomous Vehicles**

I. Introduction

The increase in computing, caching and communication capabilities of CAVs enable them to send and receive trajectories along other perception data in real-time for safe navigation around each other. Vehicles can leverage the shared data from nearby vehicles for collective navigation decisions, where energy efficient routes can be defined based on nearby vehicles intended trajectories, while avoiding other vehicles and obstacles.

We envision an intersection traffic scenario in a world of CAVs with no signalling required since all vehicles crossing the intersection would proactively share the path they would take during the crossing of the intersection. To realize this, the data from each vehicle crossing the intersection need to be trusted, and therefore, there is a need to verify the truthfulness of the trajectories shared by vehicles.

Currently, there exist no solution to trust such data shared among vehicles in real-time and therefore, in this paper, we propose **OBSERVE**, **O**ptimized **B**lockchain to **S**tore and **E**ndorse **R**eliability of **V**ehicles **E**fficiently. A blockchain based zero trust protocol for real-time distributed vehicles trajectory data validation and endorsement through consensus. In OBSERVE, first, vehicles self-organize to share each others trajectories, each vehicle receiving the trajectory from a neighboring vehicle predicts the corresponding neighbor trajectory for a given amount of time. It endorses the vehicle's reliability if the predicted trajectories are similar to the actual GPS coordinates of the vehicle during that time. The endorsement information is then stored on a distributed ledger. Blockchain is a promising candidate for storing immutable records regarding trust on nodes, where multiple peers in the consortium based permissioned network mutually endorse each other's data using a consensus mechanism [1], where a majority of peers agreeing on the truthfullness of a given node's data enables it to build trust in the network.

Additionally, we employ simple machine learning algorithms on vehicles to reduce the computing requirements for performing the predictions, since complex algorithms would require more computing resources and therefore, providing an energy efficient *trust, but verify* solution considering future Electric Vehicles (EVs) or E-CAVs. We evaluated the proposed OBSERVE protocol using several machine learning models, including transformers [2] and vision based models [3], and the results show it to be accurately predicting trajectories, while consuming less computing resources on the vehicles.

The contribution of this paper are summarized as follows:

- We propose OBSERVE, a zero trust protocol for endorsing vehicles trajectories data in real-time using blockchain-based consensus.
- We employ several machine learning models to develop the predictive trust on nearby vehicles in using neighboring vehicles trajectory data.
- We show that the desired predictive trust can be achieved using simple machine instead of employing complex models, with the goal to reduce resource consumption on vehicles.
- We evaluated the proposed OBSERVE protocol using simulated connected vehicles trajectories from New York city, and have shown that it can efficiently develop trust among nearby vehicles in real-time.

The rest of the paper is organized as follows. The following section highlights the related work. We describe the proposed OBSERVE protocol in Section III. Evaluation using different machine learning models on connected vehicles data is performed in IV along discussion on the results. In section V, we conclude the paper along providing insights on our future directions.

979-8-3503-7099-7/24/$31.00 ©2024 IEEE

II. Related Work

There has been numerous work done with the idea of realizing CAVs. With respect to our area of interest with reducing network overhead and decreasing latency, there has been work in deploying similar models aiming to leverage the vast amount of sensory data available.

The problems that we aim to address that these other works do not is the aggregated inputs of sensory features increase the computational overhead which is a vital constraint for vehicles with limited power supplies. Additionally, adding in more metrics and sensory inputs decreases the network latency which also can be detrimental in a real-time environment with a topologically dynamical environment.

Transformers, first introduced by Google Brain team in 2017 [4] changed the landscape with respect to how we view recurrent neural networks and ultimately steering away from these architectures for sequence related tasks. Their proliferation in the last few years has been widely profound and their use-cases continues to be expanded upon today as we learn about their potential for processing sequential inputs. Pursuant to our use case of transformers in CAVs, Qu et al. propose the validity of the application of transformers in robust CAVs models for privacy preservation through Federated Learning [5]. They surmised that transformer models could mitigate device forgetting which helps accelerate global model convergence when dealing with strong data heterogeneity. Much speculation and interest has been garnered with the recent advent of the power of transformer models. Their recent introduction to CAVs has seen a huge proliferation with promising results. Liu et. al made considerable gains in predictive motion trajectories through multimodal transformers [6]. Wang et al. propose a fusion of LiDAR and camera images to perform object detection with stacked transformers [7]. Prakash, Chitti, and Geiger also propose a multimodal transformer model leveraging LiDAR image representations as well as HD Camera inputs for end-to-end image prediction [8]. While these research efforts propose novel concepts and introduce increased metrics for object detection through transformers, none factor in the important resource-dependent constraints on CAVs. Our work with OBSERVE targets to measure more efficiently the resource constraints while trying to minimize network latency.

With the strong proliferation of blockchain technologies and ideas first conceptualized by Satoshi Nakamoto [9], an inundation of thought and ideas around the world of blockchain has exploded. Their use-cases have expanded beyond what S. Nakamoto originally thought and have shown strong demand in all areas of privacy (Health, Finance, Advertising, etc.). There resistance to tampering and non-repudiation makes them a strong candidate for a vehicular ad-hoc network. To date, Noh et al. gave a retrospective overview of how encryption schemes can further be applied to public ledgers in an vehicular ad-hoc network and also performed a rudimentary analysis on selected consensus algorithms and their effects as network size scales. [10]. X. Zhang and X. Chen gave a perspective on how VANETs can leverage a consortium blockchain can be used to offset immense storage constraints required for densely populated networks [11]. We adopt the notion of using a consortium blockchain in our network for non-repudiation, quick ledger access for vehicle trust, and storage concerns.

III. The Observe Protocol

In this section we describe the proposed zero trust protocol, we begin by first defining the scope of the system model below.

A. System Model

The system comprises of a set of nodes, such as vehicles $V = \{v_1, v_2, ...v_n\}$ connected through wireless communication technology such as Dedicated Short-Range Communications (DSCR) or Cellular (C-V2X) enabling each vehicle to share trajectories as well as predictions and endorsement information storage on blockchain.

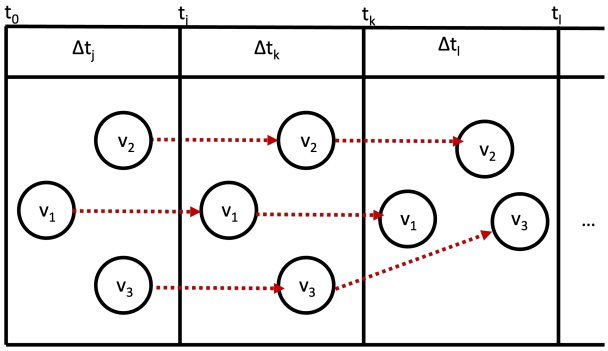
The trajectory of a vehicle v is defined as $C_v(l,t) =< (l_v^1, t_v^1), (l_v^2, t_v^2), ...(l_v^m, t_v^m) >$ with sequence of locations, $l_v^1, l_v^2, ..$ at times $t_v^1, t_v^2, ..$, respectively. The vehicle location information is further composed of the geographical latitude and longitude $l_v = (lat_v, lon_v)$ as its corresponding Global Positioning System (GPS) coordinate. We assume the vehicles are synchronized to a common clock with the time as $T = \{t_0, t_1, t_2, ...\}$ representing consecutive time instants t_0, t_1, t_2 and so on, while time can be further divided into regular slots $\Delta t_1 = t_1 - t_0, \Delta t_2 = t_2 - t_1$, and so on, defined at different granularity depending on the application.

The threat model we consider assumes the vehicle v's trajectory data $C_v(l,t)$ could be compromised, resulting in either generating fake or malicious trajectories that could be shared with nearby vehicles. To address this, we define a blockchain based zero trust protocol for nearby nodes to first verify trajectory data shared by nearby vehicles and then endorse each others trajectories at regular time intervals, where the frequency at which they are endorsed depending on the application and resource consumption requirements, however, we consider it to be less frequent in order to conserve the computing and communication resources on vehicles.

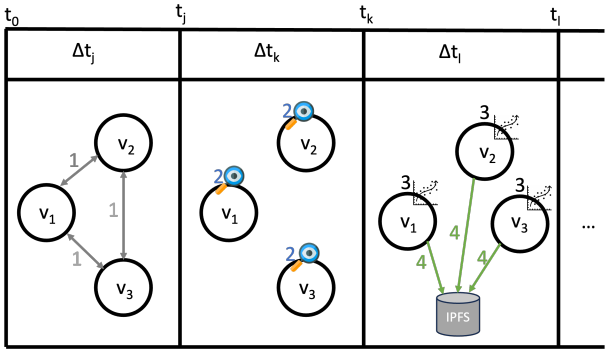
The blockchain model we consider is a permissioned blockchain where nodes are authenticated prior joining then network using state of the art public/private key based authentication or a similar encryption mechanism, therefore, developing such system is out of the scope of the paper. We employ the Inter Planetary File System (IPFS) [12], a distributed storage system for the vehicles to store the endorsement information on an immutable distributed ledger.

B. OBSERVE Protocol Workflow

The workflow of OBSERVE is described as follows. First, at time t_0, a set of CAVs, v_1, v_2, v_3 in each other's communication range exchange their respective trajectories $C_{v_1}, C_{v_2}, C_{v_3}$, comprising the path the nodes would take from time t_0 to t_l. Each node, upon receiving the neighboring vehicle trajectory splits it temporally into two parts. Assume it receives the trajectory data anytime between the initial time t_0 and t_j, it



(a) Nodes trajectories changing over time



(b) Observe protocol workflow over time

Figure 1: OBSERVE

provides the timestamped data from say, t_j to another time instant t_k (i.e. the interval Δt_k) to its machine learning model enabling in order to predict the corresponding neighboring vehicle's trajectories for the time slot Δt_l between t_k and the specified time instant t_l.

It then compares the predicted trajectories with the ones shared by the neighbor in order to verify the truthfullness of the corresponding neighbor's trajectory between time instants t_0 and t_l. In case the average difference between the neighbor shared trajectories and the ones predicted by the vehicle is less is than a predefined error threshold ϵ, the nodes trajectories between time instants t_0 and t_l are endorsed and this information is updated on the distributed ledger IPFS.

$$\frac{1}{|\Delta t|}\sum_{\Delta t\in[t_0,t_l]} C_v(l,t)[shared] - C_v(l,t)[predicted] \leq \epsilon$$

We consider here a generic Δt assuming variation in the number of slots the time between the two instants $[t_0, t_l]$ is divided. We consider different ϵ can be used depending on the accuracy of the trajectory sampled by each vehicle

Similarly, all neighboring nodes employ the above endorsement process and the final consensus on the truthfullness of the node's trajectory from time instant t_0 and t_l is updated on the blockchain in case a majority of nodes present in the vicinity of the vehicle between time instant t_0 and t_l successfully endorse its trajectory for the time duration $[t_0, t_l]$.

We consider the relation below to define the majority, however blockchain architectures typically considers 51% peers endorsing data as majority:

$$\gamma_u(C_v) = \begin{cases} 1, & \text{if} \frac{\sum\limits_{u_v\in V} u_v}{\sum\limits_{u(l,t)\in V} u(l,t)} > m \\ 0, & \text{otherwise} \end{cases}$$

where, $\gamma_u(C_v)$ is a binary variable representing a consensus among a set of vehicles in the sum $\sum\limits_{u_v\in V} u_v$ out of the total peers $\sum\limits_{u(l,t)\in V} u(l,t)$ in the location l and time t (i.e. present in the vicinity of the node v whose C_v, trajectory data is endorsed). In the above relation, considering $m = 0.5$ represents the 51% majority of the vehicle in a location and at a particular time reaching a consensus to validate the data shared by vehicle v.

Finally, the entry on the blockchain comprises of node IDs of the set of nodes for whom the data is endorsed through consensus. A vehicle's reliability or reputation for constantly getting its trajectory data endorsed can be evaluated from the number of times it appears in the ledger.

Figure 1 depicts a toy version of OBSERVE protocol. In the first figure, neighboring vehicles v_1, v_2 and v_3 trajectories over time are shown, where the time interval $[t_0, t_l]$ is divided into consecutive time slots $\Delta t_j, \Delta t_k$, and Δt_l. As an example we show only vehicle v_3 changing its trajectory in the time interval, for v_1 to leave space for it.

The second figure shows the trajectory exchange among all nodes happening during time slot Δt_j, each vehicle *observes* neighbouring vehicles trajectories during time slot Δt_k to predict the corresponding trajectories during time slot Δt_l. Each vehicle compare the trajectory shared and predicted during this time slot to decide whether to endorse neighboring vehicle trajectory. In case the threshold is within ϵ, as indicated above, all nodes update this endorsement on the distributed ledger at time instant t_l. Upon reaching a consensus where majority of the nodes indicated by m successfully endorse a node's data, an block is created on the distributed ledger comprising the node IDs of the node(s) for whom consensus is reached.

IV. Numerical Evaluation

We describe below the numerical evaluation process for OBSERVE using different machine learning algorithm along discussing the results.

A. Experiment Setup

The experiment setup to evaluate the machine learning models for trajectory prediction primarily considers factors such as FLOPs (floating point operations per second), CPU utilization, and model parameters. Furthermore, the investigation of evaluating OBSERVE explores various types of models for the prediction on time series trajectory data, specifically RNNs (Recurrent Neural Networks), GRUs (Gated Recurrent

Units), LSTMs (Long Short-Term Memory), and Transformers are employed, all trained solely on trajectory data.

The considered models, including RNNs, GRUs, LSTMs, and Transformers, are widely used in sequence modeling tasks due to their ability to capture strong spatial-temporal dependencies. However, the evaluation focuses on their efficiency and computational complexity rather than exploring hybrid approaches. This approach acknowledged the potential benefits of simpler models that could be trained and deployed with reduced computational resources, while relying solely on trajectory data compared to the computationally intensive visual data based trajectory prediction.

The primary objective of this approach is to identify machine learning models that exhibit lower compute intensity while maintaining satisfactory performance in trajectory prediction tasks when comparing metrics such as Mean Absolute Error (MAE) and Precision in correct neighbor trajectory prediction. By analyzing FLOPs, CPU utilization, and model parameters, the study aims to gain insights into the computational efficiency of different model architectures together with OBSERVE, in particular exploring shallow network models with often just one or two hidden layers. This analysis provides a quantitative understanding of the resource requirements of each model used with OBSERVE, enabling a more informed comparison of the computational requirements.

Furthermore, we use FLOPs, CPU utilization, and model parameters to identify models that strike a balance between computational efficiency and trajectory prediction accuracy. This analysis can inform the selection of models that are suitable for deployment in CAVs networks, particularly in scenarios where compute resources may be limited or where cost-effective solutions are desired. These traits are highly sought after in a resource-constrained environment such as E-CAVs where minimizing resource consumption leads to a greater range of travel and stronger selling point for vehicle manufacturers.

B. Data and pre-processing

The training data for the sequencing models is composed of a 3 hour 20 minute synthetic trajectory trace of Brooklyn, New York city neighborhood. The trace is comprised of over $20,000$ unique vehicles with sensory data including GPS coordinates, acceleration, angular measurement (with respect to steering wheel), and speed. The Brooklyn neighborhood has four lanes of bi-directional traffic with crosswalks and stoplights at each intersection. The synthetic data is meant to be as representative as possible of a real urban environment where random events (lane changes, pedestrians, stop lights) could occur to keep a sequencing model as robust as possible and widely applicable for such a task.

In order to prepare the data for sequencing predictions, the data is further processed to segregate unique vehicles that maintained a spatial-temporal relationship of 300 feet for 60 seconds at a time, thus, defining neighboring vehicles to be within 300 feet spatially, and temporally within 60 seconds to be able to complete OBSERVE workflow. We selected 300 feet as a typical short range wireless communication range, where 60 seconds is chosen to compare predictions in the one minute time interval i.e. $[t_0, t_l]$ along keeping optimal model size and training times in perspective for short time series forecasting predictions.

1) Experiment Hardware: All data preparation and model runs are conducted on an Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz Processor with 16 GB of installed RAM. The training and test data as well as model are optimized to train on a single NVIDIA 1670 GPU with 8 GB of VRAM made possible with PyTorch's CUDA toolkit version 11.8.

2) Experiment Software: All model runs are conducted with a Windows 10 operating system utilizing Python version 3.6. The machine learning framework used is PyTorch 1.13.1 with CUDA version 11.6.

C. Results

We propose to utilize light-weight machine learning models for trajectory prediction in CAVs, aiming to achieve high accuracy while minimizing the resource requirements. By harnessing the potential of these light-weight models, we demonstrate the feasibility of reducing resource consumption without compromising prediction performance. These efforts are particularly useful in demonstrating the effectiveness of CAVs in blockchain networks with the absence of computer vision, which is often resource intensive compared to the sequencing networks considered.

Through our investigation of sequencing networks and finding optimal neighborhood size we found that each model had varying degrees of applicability given the prediction length (3 seconds up to 24 seconds) and neighborhood sizes (1 to 150 nodes). The Gated Recurrent Unit (GRU) model performed optimally given certain constraints like training time and Mean Absolute Error (MAE). The Informer transformer was able to achieve near-perfect prediction accuracy based on MAE, but the training time for the short time series forecast (STSF) prediction was not applicable given the immediate need for quick prediction and low resource consumption.

First and foremost, we analyzed the Mean Absolute Error (MAE) (Table 1) as a fundamental metric to assess the accuracy of our trajectory predictions with OBSERVE. By comparing the predicted trajectories with the ground truth data, we observed consistently low MAE values across various models. This indicates that our light-weight models are fully capable of generating highly accurate predictions, thus instilling confidence in their reliability for CAVs trajectory prediction tasks.

Moreover, we delved into the computational efficiency of OBSERVE using different models by examining the floating point operations per second (FLOPs) metric (Table 2). FLOPs serve as a measure of the number of floating-point operations required to perform the prediction task. Our findings revealed that the light-weight models employed in OBSERVE exhibited significantly reduced FLOPs compared to their more complex counterparts. This reduction in computational complexity translates to improved efficiency and reduced resource

Model	MAE	Accuracy
LSTM	6.3	93.91
RNN	7.2	92.29
GRU	6.1	95.37
Transformer	0.7	99.53
OAD	6.2	99.75

Table I: Illustrates the Mean Absolute Error comparing different sequencing models and combining their accuracy for neighboring node trajectory predictions.

Model	Sequence Length	Params	FLOPs	CPU Util. %
LSTM	480	335.49K	43.34	57.7
RNN	480	83.97K	10.78	6.4
GRU	480	251.65K	32.54	50.8
Informer	96	32.3M	59.12	92.2
OAD	480	7.5M	49.40	78.79

Table II: Model parameters for each sequencing model tests. These metrics serve as a baseline for shallow model complexity to minimize resource restraints.

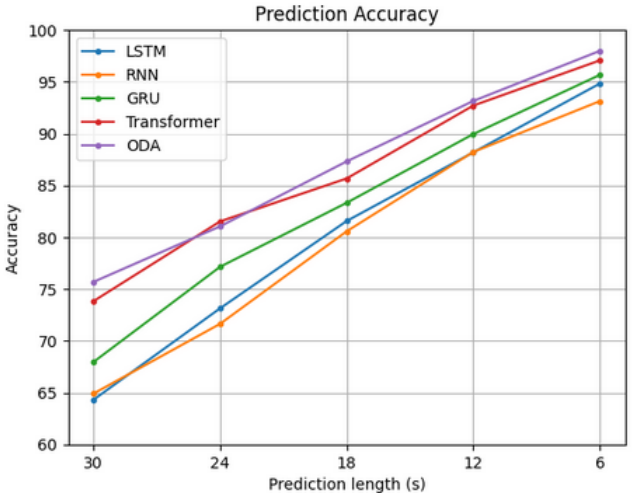


Figure 2: Advantages of using object detection assisted trajectory predictions but at what expense denoted in Table 2. Accuracy increases as the time prediction decreases towards 6 seconds.

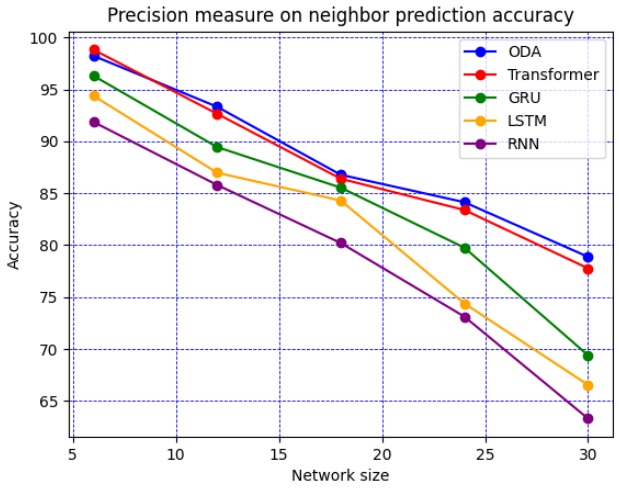


Figure 3: Comparing the precision for different models at predicting their neighbors based off of coordinate data.

consumption, which is a crucial consideration for real-time trajectory prediction in resource-constrained CAVs where E-CAVs are emerging as clear winners in this domain with limited battery constraints.

To further investigate the resource utilization of OBSERVE, we assessed the CPU utilization (Table 2) during the prediction process. Remarkably, our light-weight models demonstrated efficient utilization of CPU resources, resulting in lower computational demands. This implies that the models are well-suited for deployment on hardware with limited processing capabilities. The ability to achieve accurate trajectory prediction while utilizing fewer CPU resources highlights the practicality and scalability of our approach.

Finally, we evaluated the overall prediction accuracy achieved by OBSERVE with light-weight models compared to a YOLOv5 object assisted detection (OAD) model (Figures 2 and 3). Through extensive testing and validation, we observe that these models consistently achieved high accuracy rates, keeping close to the performance of more resource-intensive model such as YOLOv5. This is a significant finding as it demonstrates that it is possible to strike a balance between accuracy and resource consumption, favoring the adoption of light-weight models for trajectory prediction in OBSERVE.

Further results were compiled for short time series forecasting predictions at the 30 second, 24 second, 18 seconds, 12 second, and 6 second intervals ahead. Results presented above are from the standard 80/20 train/test split and as such 12 second predictions into a vehicle's future trajectory.

With limited training data given 60 second training times, the goal of Table 1 was to look at the input sequence length comparing how their input size affected CPU utilization with respect to parameters. Too few parameters runs the risk of underfitting and too many, the opposite as well as a larger memory footprint. The goal was to minimize model complexity while still being able to accurately predict to the test data. The floating point operations per second (FLOPs) also serves as a benchmark to look at how the computer hardware is being used efficiently with respect to CPU utilization which correlates with the model's complexity.

In summary, our results emphasize the effectiveness of light-weight machine learning models leveraging blockchain technology in achieving accurate trajectory prediction in autonomous vehicles while minimizing resource requirements. By attaining low mean absolute error, reduced FLOPs, efficient CPU utilization, and high overall prediction accuracy, our study highlights the potential of leveraging light-weight models for trajectory prediction tasks. This research contributes to the advancement of efficient and practical solutions for CAVs.

V. Conclusion

The trend towards Electric Connected and Autonomous Vehicles (E-CAVs) brings the need for energy efficient trust models. Vehicles in each other's vicinity need to verify the truthfulness of trajectories and other data shared by neighboring vehicles, in real-time, to avoid potentially malicious or false data usage for navigation and path planning decisions. We proposed the first zero trust protocol, OBSERVE as a *trust, but verify* mechanism for nearby vehicles to predict the trajectories shared by neighboring vehicles, and endorse their truthfulness in case the predicted and shared trajectories are similar. Information regarding a successful endorsement of a vehcile trajectory from a majority of neighboring vehicles is then stored on a distributed ledger (IPFS) as the blockchain based solution to maintain an immutable record of the trust a vehicle builds over time. We employed lightweight machine learning algorithms for the prediction in OBSERVE and have achieved higher accuracy and lower resource utilization.

Future directions includes scaling, both spatially to consider longer and more trajectories, as well as considering longer time duration, for a large number of vehicles. Additionally, we are considering to implement OBSERVE as a federated learning approach to further preserve privacy and improve efficiency of the proposed protocol in a distributed environment.

References

[1] J. A. Khan and K. Ozbay, "Affirm: Privacy-by-design blockchain for mobility data in web3 using information centric fog networks with collaborative learning," in *2023 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2023, pp. 456–462.

[2] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[3] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[5] L. Qu, Y. Zhou, P. P. Liang, Y. Xia, F. Wang, L. Fei-Fei, E. Adeli, and D. Rubin, "Rethinking architecture design for tackling data heterogeneity in federated learning," *arXiv preprint arXiv:2106.06047*, 2021.

[6] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal motion prediction with stacked transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7577–7586.

[7] Z. Wang, X. Zhang, S. Wang, T. Xin, H. Zhang, and J. Lu, "Multi-scale spatial transformer network for lidar-camera 3d object detection," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[8] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7077–7087.

[9] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[10] J. Noh, S. Jeon, and S. Cho, "Distributed blockchain-based message authentication scheme for connected vehicles," *Electronics*, vol. 9, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/1/74

[11] X. Zhang and X. Chen, "Data security sharing and storage based on a consortium blockchain in a vehicular ad-hoc network," *IEEE Access*, vol. 7, pp. 58 241–58 254, 2019.

[12] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.