Accepted version

To appear in *Proc. 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring),* Florence, Italy, 2023.

Prediction of Communication Delays in Connected Vehicles and Platoons

Shahriar Hasan*, Joseba Gorospe†, Arrate Alonso Gómez†, Svetlana Girs*, Elisabeth Uhlemann*
*School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden
†Electronics and Computer Science Department, Mondragon Unibertsitatea, Arrasate, Spain
Corresponding author: S. Hasan (e-mail: shahriar.hasan@mdu.se)

***Abstract*—Automated vehicles connected through vehicle-to-vehicle (V2V) communications can use onboard sensor information from adjacent vehicles to provide higher traffic safety or passenger comfort. In particular, automated vehicles forming a platoon can enhance traffic safety by communicating before braking hard. It can also improve fuel efficiency by enabling reduced aerodynamic drag through short gaps. However, packet losses may increase the delay between periodic beacons, especially for the rear vehicles in a platoon. If the Lead Vehicle (LV) can forecast link quality, it can assign different platoon performance levels in terms of inter-vehicle distances and also facilitate the designing of safer braking strategies. This paper proposes a strategy for incorporating Machine Learning (ML) algorithms into, e.g., the LV of a platoon to enable online training and real-time prediction of communication delays incurred by connected vehicles during runtime. The prediction accuracy and its suitability for making safety-critical decisions during, e.g., emergency braking have been evaluated through rigorous simulations.**

I. Introduction

Automated driving in vehicle platoons can improve traffic safety, fuel efficiency, and traffic flow by enabling the Following Vehicles (FVs) to react to the speed changes of the Lead Vehicle (LV) through Vehicle-to-Vehicle (V2V) communications. In platooning, the LV periodically broadcasts its status and attributes, whereas event-driven messages are disseminated when a situation of common interest, e.g., a hazard, occurs. Automated vehicles can also use this platooning strategy in dense traffic situations. A vehicle in front can broadcast its speed periodically to the vehicles behind and also inform them about an intention to brake. In V2V, communication delays are time-varying and can be very high in dense data and road traffic scenarios [1]. In addition, since the rear vehicles are further away from the LV, they may experience more frequent outages and packet loss due to path loss, shadowing, and fading effects. One solution to this is to maintain short gaps between the vehicles, as this is also good for fuel efficiency. However, even though the likelihood reduces, packet losses may still occur and cause problems with safety since there is less time to react in case, e.g., emergency braking should be necessary. On the other hand, having longer inter-vehicle distances can result in losing contact with the LV and lead to reduced fuel efficiency. To this end, being aware of the experienced communication quality, e.g., the delay between periodic updates from the leading vehicle, is an important factor in order to make suitable control decisions that enables fuel efficiency while providing safety.

In this paper, we aim to find out whether or not such communication delays predicted using machine learning (ML) algorithms can be sufficient to keep a platoon safe also in case of, e.g., emergency braking. The type of communication delays we are considering in this paper is mainly affected by the likelihood of the successful delivery of messages between the transmitter and the receiver [2]. Being able to assess the likelihood of successfully receiving packets or characterizing the communication delay can facilitate assigning different platoon performance levels in terms of inter-vehicle distances. In addition, the forecast of link quality can help us design better braking strategies and thereby safer platoons. To this end, we employ ML algorithms in the LV of a platoon that collects data from the FVs, trains the algorithms online, and makes a real-time prediction of communication delays. The predicted delays are then used to make safety-critical decisions in platooning.

Aiming to predict the communication quality is not new. The authors in [3], [4] focus on Quality of Service (QoS) predictions in Cellular Vehicle-to-Everything (C-V2X) communications, which is important for applications such as teleoperated driving. Additionally, many previous works focus on defining a framework installed in an external service with information from a centralized network and ML algorithms trained offline with historical data to capture trends, e.g., [4], [5], and [6]. However, the network topology of a platoon may frequently vary due to, e.g., communication outages, vehicle joining, leaving, cut-in/cut-out, merge maneuvers, and more. In addition, the communications delays experienced in platooning vehicles may also vary due to the change in data and road traffic density. Hence, offline training of ML algorithms cannot facilitate the real-time prediction of communication delays required for adapting to the change in network topology and neighboring traffic density of a platoon. In contrast, online training of ML algorithms and real-time predictions can enable platooning vehicles to take appropriate measures for mitigating the effects of communication delays on string stability during cruising and safety during emergency braking.

To the best of our knowledge, this work is the first of its kind that attempts to use ML as an onboard prediction tool to facilitate online training of ML algorithms to predict communication delays in real-time. The motivation behind such real-time predictions is that information regarding expected delays can be used to, e.g., synchronize the actuation of vehicles to achieve string stability [7], synchronize the braking of vehicles to transition to a fail-safe state [8], and more. To achieve this, the FVs in a platoon can send information regarding experienced communication delays to the LV during platooning runtime. The LV uses these data to train the onboard ML algorithms and make predictions. The predicted information, e.g., expected maximum communication delay and relevant instructions, can be encapsulated in the next packet from the LV so that the FVs can use the predicted delay to follow the instructions of the LV. This work aims to understand whether or not the existing ML algorithms can predict the time-varying communication delays in real-time with sufficient accuracy required for making safety-critical decisions in platooning. To this end, we have conducted rigorous simulation studies under various neighboring traffic loads and beacon rates to evaluate the performance of two different ML algorithms in predicting communication delays. In order to demonstrate the benefits of communication delay predictions, we evaluate an emergency braking use case in which the predicted delays are used to perform delay-aware emergency braking.

The rest of the paper is organized as follows: In Section II, recent works employing ML algorithms in predicting different communication, network, and traffic parameters are described. Next, the system model is described in Section III. Two different ML algorithms used for prediction and their parameter selection are explained in Section IV. After this, the simulation settings and the evaluation results are presented in Sections V and VI, respectively. Finally, Section VII concludes the paper.

II. Related Works

Moreira *et al.* employ several ML algorithms to predict whether or not a packet of a specific size can be delivered to a vehicle from the base station within a specific time frame in C-V2X systems [5]. Torres-Figueroa *et al.* study whether or not a particular QoS, e.g., 50 ms or 100 ms delay, can be achieved when a previously trained ML model is employed in a vehicle and predictions are made based on certain Key Performance Indicators (KPIs) [3]. The authors conclude that when end-to-end delays are predicted at a vehicle without information from the base stations using ML algorithms, the prediction accuracy is insufficient for making decisions in safety-critical systems. The works in [3] and [5] consider the delay predictions in a C-V2X network but do not show how the ML models could perform in VANETs. Zhang *et al.* propose a latency prediction framework for V2X applications in [6] and show that Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) can predict latency with lower errors than other algorithms. Barmpounakis *et al.* in [4] propose a PreQoS approach to predict QoS metrics, e.g., data rate, packet error rate, and end-to-end delay, at the 5G core network that is aware of the traffic and data density. In [9], the authors analyze the performance of an LSTM-based Multivariate Multistep Autoencoder to predict the Uplink (UL) throughput in a teleoperated driving scenario. The algorithm is trained and tested offline with the collected data during the simulations, concluding that the LSTM model can capture most UL throughput fluctuations. The works in the literature proposing strategies for predicting communication and network parameters mostly rely on the base station of a centralized network, e.g., C-V2X, to collect data, or the collected data are trained offline to make predictions.

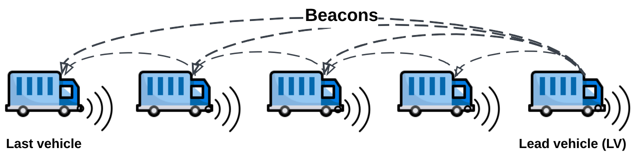
Khan *et al.* propose an ML-based prediction model in which a vehicle in a VANET learns the channel activity of the neighboring vehicles for some time and uses LSTM RNN to predict the neighboring vehicles' transmissions in the next time window [10]. The prediction of such channel activities can be used for scheduling the periodic messages of the learning vehicle. Sangare *et al.* in [11] use a Support Vector Machine (SVM) algorithm to predict the probability of successful reception of a transmission between a vehicle and a Roadside Unit (RSU), given the transmission rate and the distance between the vehicle and the RSU. For training the model, the authors generate data using an analytical model to mimic the transmissions in a vehicular network with IEEE 802.11p MAC protocol [12]. Alarcon-Aquino and Barria propose a Multiresolution Finite-impulse-response (FIR) NN algorithm to predict network traffic [13]. The authors use real-world ethernet traffic data to train the algorithm and predict future traffic. The results show that the proposed approach is more accurate in predicting time series compared to Multiresolution NN or FIR NN.

Previous studies show that statistical models such as Hidden Markov Models (HMMs) or ML algorithms such as Convolutional Neural Networks (CNN) fail to consider long-term dependencies for predicting sequential data [10]. LSTM circumvents this issue by considering long-term data for prediction, and it is widely used for predicting time sequences [3], [14]. To this end, LSTM RNN is used in this paper to predict communication delays in a platoon during runtime using only the previously experienced delays. In addition, the performance of Accurate Online Support Vector Regression (AOSVR) presented by Ma *et al.* [15] is also evaluated in predicting communication delays and compared with LSTM RNN. The main rationale behind using the AOSVR algorithm is that it can be trained online upon adding new data into the dataset without having to do all the regressions from scratch for each newly added sample, as done in traditional SVR.

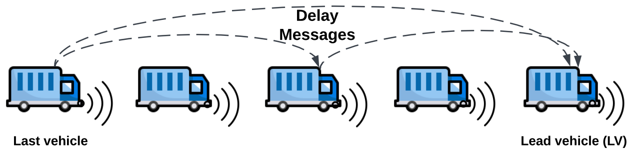
III. System Model

Without loss of generality, we assume that the LV broadcasts beacons using the Cooperative Adaptive Cruise Control (CACC) law proposed by Rajamani *et al.* in [16] as part of the California PATH program. The PATH CACC controller dictates that an FV receives control input both from the LV and the preceding vehicle, as shown in Figure 1a, thus facilitating lateral and longitudinal control.

Every time a vehicle receives a beacon from the LV through V2V communications, it can compute the communication delay by looking at the timestamps of the current beacon and the previously received beacon. However, if all vehicles were to inform about their experienced delays, we risk congesting the channel. To this end, we assume that only the last vehicle



(a) LV broadcasts beacons.



(b) One or more of the FV informs the LV of the experienced beacon age.

Fig. 1: Communication topology for delay prediction.

broadcasts its delay value intended to be received by the LV; see Figure 1b. The rationale behind this is that the last vehicle in a platoon is likely to experience the highest communication delay due to packet losses [1]. In addition, if emergency braking is required, the last vehicle must brake earlier [17], and thus adjusting the platoon actions to its experienced delay provides a better safety margin. However, if another type of safety application requires all delay values and additional information, such as the gap to the vehicle ahead from all the FVs, the communication model outlined in Figure 1b can be modified without loss of generality. Once the last vehicle computes its experienced delay, it is encapsulated in a message called *Delay Message* and broadcasted. As the *Delay Message* is triggered upon receiving a beacon from the LV, it is not a periodic beacon; instead, a *Delay Message* can be categorized as an event-driven message. The *Delay Message* is also relayed by the middle vehicle in the platoon to achieve better reliability. If the same *Delay Message* is received both from the last vehicle directly and the middle vehicle through relaying, the latest one is discarded by the LV. Every time the LV receives a *Delay Message*, an ML algorithm can be used to predict the maximum expected delay in the next beacon period. The predicted delay value can then be included in the next beacon so that all the vehicles are aware of the maximum expected delay within the platoon, i.e., the one experienced by the last vehicle. Note that the *Delay Message* may also be lost despite relaying, and there is no way of guaranteeing that it will always be successfully delivered to the LV.

IV. ML Algorithms for Real-Time Prediction

The LSTM RNN and AOSVR algorithms are evaluated for real-time prediction of communication delays in this paper.

The main characteristic of RNNs is that each cell of the layer considers the previous output as a piece of extra information in the current step. This allows capturing the sequence or temporality of the data, as the cells consider historical information and thread it together with the output. LSTM networks are a particular type of RNN that can take long-term dependencies into account to predict time series. Figure 2 shows the schematic representation of the LSTM network used

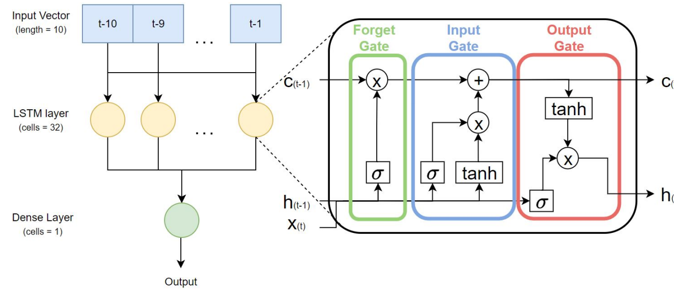


Fig. 2: Structure of the LSTM RNN.

in this work. This paper uses Keras[1] on top of Tensorflow[2] to develop the LSTM RNN. Ten previous communication delays are used as input for the LSTM RNN. The inputs are processed using a single LSTM layer composed of 32 cells, as depicted in Figure 2. The information obtained from the 32 cells is combined using a dense layer that gives a single output value. The training of the algorithm is done online at the LV of a platoon, adjusting the weights of all the cells each time a new delay message is received from the last vehicle and using the Adam optimization algorithm [18] with a learning rate of 0.001. The structure in Figure 2 has been kept simple to minimize the number of layers and cells, thereby reducing the inference time and ensuring real-time predictions.

Ma *et al.* propose the AOSVR algorithm in [15], which aims to optimize online data training. In the case of an SVR, it is necessary to calculate regressions from scratch every time the training set changes. AOSVR, on the other hand, allows updating SVR regressions each time a sample is added or removed from the training set. Ma *et al.* proposed an Incremental and a Decremental algorithm to adapt the regression function while the samples of the training set are changing. The Incremental algorithm updates the value of the previous samples based on the new one, and the Decremental (or "unlearning") algorithm removes samples from the training set and adjusts the regressions if necessary. In [19], Parrella studied the AOSVR proposed by Ma *et al.* and made an open-source implementation. With the AOSVR in [19], three parameters are required to be specified, e.g., ϵ, which defines an error margin in which predictions are not penalized, C_{SVR}, which is a regularisation parameter that determines the amount of misclassification that shall be avoided and $SizeLimit$, which limits the number of valuable samples that are taken into consideration for making a new prediction. In this paper, the AOSVR implemented by Parrella is employed at the LV in the platoon to facilitate online training and to make predictions of communication delays in real-time.

V. Simulation Settings and Evaluation Metrics

We implemented the proposed delay communication strategy and the ML algorithms in the PlatoonSAFE simulator[3]. PlatoonSAFE is an extension of the platooning simulator

[1]https://keras.io/
[2]https://www.tensorflow.org/
[3]https://github.com/shahriarHasan09/PlatoonSAFE

TABLE I: Configuration parameters for simulations and analysis.

Parameter	Value	Parameter	Value
Path loss model	Free space ($\alpha = 2$)	Fading model	Nakagami-m (m = 1.86)
PHY/MAC model	802.11p/1609.4	Frequency	5.89 GHz
Sensitivity	-94 dBm	Thermal noise	-95 dBm
Packet size	200 B	Tx power	100 mW
Platoon size	7	PATH CACC CDG	0.4 s
speed	100 kmh^{-1}	Deceleration rate	$-8\ ms^{-2}$
brakeAtTime	100 s	simulation time limit	110 s

TABLE II: Different configurations for varying the channel load.

Configuration no.	Neighbouring traffic				Platoon beacon frequency (Hz)	
	vehicles	vehicles/km	beacon frequency (Hz)	packets s^{-1}km^{-1}	periodic beacon	event-driven
Config1	500	95	50	4750	10	10
Config2	400	95	50	4750	10	10
Config3	300	65	20	1300	15	15
Config4	50	36	10	360	10	10

PLEXE [20], which inherits the implementation of several control algorithms, e.g., PATH CACC, realistic vehicle dynamics, engine models, and simulation of platoons under mixed-traffic scenarios from PLEXE. In addition, PLEXE extends the Veins simulator [21], which facilitates bi-directional coupling with the road traffic simulator SUMO [22] and provides several PHY layer channel models. We integrated the AOSVR algorithm, implemented by Parrella[4], into PlatoonSAFE. To incorporate the LSTM RNN, we established a User Datagram Protocol (UDP) communication between PLEXE and a Python module that utilizes Keras on top of TensorFlow to run the neural network.

We simulated a platoon of seven vehicles cruising at 100 kmh^{-1} using the PATH CACC controller with 5 m gaps. During cruising, either AOSVR or LSTM RNN is used to predict the communication delays at the LV upon receiving a packet from the last vehicle containing its experienced delay value. In order to estimate the values of the AOSVR algorithm parameters, we have conducted rigorous simulations by varying the traffic and data density of the neighboring vehicles in a platoon. The AOSVR algorithm is then tested by varying the ϵ, C_{SVR}, and $SizeLimit$ parameters and using the obtained simulation results to understand which combination of the parameter values demonstrates better prediction accuracy. Our results suggest ϵ, C_{SVR}, and $SizeLimit$ values 0.0001, 0.03, and 5.0, respectively. The PHY and MAC layer parameters used for simulation analysis are listed in Table I.

In order to model the neighboring traffic, we consider four simulation configurations as depicted in Table II. Configs 1 and 2 are intended to generate high data and road traffic in simulations. Configs 1 and 2 only differ by the number of neighboring vehicles to understand their effects on communication delays. Config 3 is intended to generate a moderate level of delay, and it has a significantly lower traffic density (vehicles/km) and packet density (s^{-1}km^{-1}) compared to Configs 1 and 2. Finally, Config 4 represents a sparse traffic scenario. We evaluated the ML algorithms for each configuration in Table II, and ten simulation runs were carried out for each combination. In the following list of metrics, the first two metrics are used to evaluate the performance of the ML algorithms in predicting communication delays. Moreover, metrics three and four evaluate a braking scenario in which the predicted delays are used for emergency braking.

[4]https://github.com/fp2556/onlinesvr

TABLE III: RMSE of the last vehicle's predicted delay with respect to the LV. The results of four different simulation configurations are presented.

	Root Mean Square Error (RMSE)			
Algorithms	Config 1	Config 2	Config 3	Config 4
LSTM RNN	0.477	0.430	0.072	0.026
AOSVR	0.325	0.322	0.0850	0.027
Avg. delay	0.552	0.496	0.075	0.026

1) *Root Mean Square Error (RMSE) of predicted delays*: The RMSE is calculated as $\sqrt{\sum_{i=1}^{n}(\frac{(y_i-\hat{y}_i)^2}{n}}$, where y_i is the actual delay and $\hat{y}_i$ is the predicted delay of the last vehicle with respect to the LV using ML algorithms.
2) *RMSE of average delays*: To provide a baseline for comparison, we include a benchmark case where the RMSE is calculated between the actual communication delay and the average of previously experienced actual delays by the last vehicle.
3) *Inter-vehicle collisions*: The inter-vehicle gaps between any two vehicles in a platoon at a complete standstill must be greater than zero to avoid collisions.
4) *Stopping distance of the LV* (m): The stopping distance of the LV is calculated as the time between hazard detection until the LV reaches zero speed.

Note that metrics three and four represent the fail-safe conditions, which dictate that a platoon must avoid collisions and minimize the stopping distance of the LV to avoid the hazard that caused the emergency braking [1].

VI. Evaluation of Prediction Errors and Emergency Braking Using Predicted Delays

In this section, we first evaluate the ML algorithms in terms of RMSE. Then we evaluate the emergency braking use case to understand the extent to which the predicted delays can be used to make safety-critical decisions.

A. Evaluation of LSTM RNN and AOSVR in Predicting Communication Delays

Table III presents the RMSE for different configurations using the ML algorithms. The results demonstrate that Configs 1 and 2, representing dense data and road traffic scenarios, exhibit higher RMSE than Configs 3 and 4. The reason is that the experienced communication delays vary more frequently in dense scenarios as the vehicles contend to gain access to the same channel, and a vehicle refrains from transmitting for a random amount of time if the channel is found busy. In more sparse scenarios, e.g., Configs 3 and 4, the LV can predict the delays with much higher accuracy using the ML algorithms. Another possible reason for higher RMSE with Configs 1 and 2 is that the LV does not successfully receive all the *Delay Messages* sent by the last vehicle or relayed by the middle vehicle due to packet losses.

Furthermore, the results in Table III indicate that the RMSE is lower when using ML compared to the benchmark case of using average delays, validating that ML provides added benefits in approximating the communication delays. In particular, the RMSE with average delay is significantly higher than that with AOSVR in dense data traffic scenarios. However, in sparse scenarios, such as Config 4, the RMSE of predicted delays and average delays are similar, as the variation in experienced communication delays is lower in this case. In addition, the LSTM RNN algorithm performs slightly better than AOSVR in sparse scenarios, e.g., Configs 3 and 4. However, the prediction accuracy with AOSVR is better than LSTM RNN in dense scenarios, e.g., Configs 1 and 2. The rationale behind this performance difference in different traffic densities is mainly due to the process of online training with LSTM RNN and AOSVR algorithms. When a new sample, i.e., delay value, arrives at the LV, AOSVR removes the oldest sample from its list of valuable samples to include the new one and adapts the regressions. Recall that the *SizeLimit* value used in our simulations is five, i.e., the predictions with AOSVR are made based on the latest five delay values. As a result, AOSVR focuses on a more short-term prediction, which is useful when the communication delays vary more frequently, e.g., in dense data and road traffic scenarios. On the other hand, LSTM RNN slightly adjusts its weights for every new sample and considers long-term dependencies. As a result, LSTM RNN is a more conservative algorithm that shows better performance when the variation in communication delays is less frequent or less arbitrary, e.g., in sparse data and road traffic scenarios such as Configs 3 and 4.

B. Evaluation of an Emergency Braking Use Case using Predicted Communication Delays

In order to evaluate whether or not the prediction accuracy is adequate for transitioning a platoon to a fail-safe state through emergency braking upon encountering a road hazard, we consider the braking strategy proposed in [8] called the Synchronized Braking (SB). In SB, the braking actions of all the platooning vehicles are delayed for a period τ_{wait} before the entire platoon performs emergency braking synchronously. The rationale is that if, e.g., the last vehicle in a platoon is experiencing a higher delay than its predecessor, the platoon vehicles start braking at different times upon receiving a beacon, which can lead to inter-vehicle collisions when braking with a strong deceleration rate [8]. The τ_{wait} period in [8] is calculated by averaging the previously experienced delays, which can be problematic if the experienced communication delay during braking is significantly higher than the average value. To this end, we instead propose that the adopted τ_{wait} is obtained by predicting the delay using the delay prediction approach proposed in this paper. The emergency braking maneuver is evaluated in terms of collision avoidance and the stopping distance of the LV, which are considered conditions for transitioning a platoon to a fail-safe state [1].

The same simulation settings as in Table III are used to evaluate the emergency braking using the predicted delays. In this case also, the platoon cruises using the PATH CACC controller with 5 m CDG at 100 kmh^{-1}. The LV starts broadcasting event-driven messages at 100 s into the simulation time upon encountering an imaginary road hazard. The platoon performs braking using the SB strategy or the Normal Braking (NB) strategy; the deceleration rate is –8 ms^{-2}. The NB strategy implies that the platooning vehicles start braking as soon as they receive an event-driven message, i.e., no waiting before braking as in the SB strategy. In SB, the LV computes the τ_{wait} period that all the vehicles should pursue before braking by using the delay predicted by either LSTM RNN or AOSVR algorithm. Furthermore, for the sake of comparing, we evaluate the case in which the LV computes τ_{wait} by taking the average of previously experienced communication delays.

TABLE IV: No. of collisions out of 10 simulation runs.

	No. of collisions out of 10 runs.			
Algorithms	Config 1	Config 2	Config 3	Config 4
SB-LSTM	0	0	0	0
SB-AOSVR	1	0	0	0
SB-AvgDelay	0	1	0	0
NB	7	4	0	0

TABLE V: Average stopping distance of the LV (m)

	Stopping distance of the LV (m).			
Algorithms	Config 1	Config 2	Config 3	Config 4
SB-LSTM	77.536	72.535	63.869	64.009
SB-AOSVR	72.341	70.009	63.510	63.702
SB-AvgDelay	77.612	75.536	63.814	64.037
NB	60.81	60.81	60.81	60.81

Table IV depicts the number of collisions out of ten simulation runs using SB and NB strategies. There are no collision cases when prediction is done using LSTM RNN for any of the configurations. However, there is one collision case with the AOSVR algorithm and Config 1. In order to understand the reason for collisions, let us look at the corresponding stopping distances of the LV with Configs 1 and 2 and SB-LSTM and SB-AOSVR in Table V. The stopping distances with the SB-LSTM case are higher than the SB-AOSVR case because the predicted delay, i.e., τ_{wait}, with the LSTM RNN algorithm is higher than the AOSVR algorithm. This is also the reason why LSTM RNN shows higher RMSEs than AOSVR in Table III with Configs 1 and 2. The LSTM RNN algorithm overestimates the τ_{wait} period by a small margin, which helps avoid collisions during emergency braking but causes the LV to traverse a longer distance. On the other hand, SB with the AOSVR algorithm minimizes the stopping distance of the LV but causes a collision in one out of 40 simulation runs, which is due to underestimating the τ_{wait} period. Looking at the average delay scenario, where τ_{wait} is calculated by taking the average of previously experienced delays, there is one collision case with Config 2. Moreover, the average stopping distance of the LV is higher than the SB-LSTM and SB-AOSVR cases. Finally, Table IV shows that the platoon undergoes collisions for seven and four simulation runs out of ten with Configs 1 and 2, respectively, when NB is used. Although the stopping distance is shorter with the NB strategy due to no waiting before emergency braking, it is unsuitable for transitioning a platoon to a fail-safe state because of inter-vehicle collisions. Note that the platoon would require to decelerate much slower to avoid collisions with the NB strategy, which would lead to a higher stopping distance of the LV compared to the SB strategy [8]. However, in sparse data and road traffic scenarios, e.g., Configs 3 and 4, the NB strategy performs better than the SB strategy in terms of the stopping distance of the LV.

In summary, the simulation results show that AOSVR outperforms LSTM RNN and average delay in terms of lower RMSEs in dense data and road traffic scenarios. However, LSTM RNN performs better in emergency braking scenarios by avoiding all collisions due to its more conservative delay estimations. While both algorithms have their strengths and weaknesses, LSTM RNN is better suited for emergency braking scenarios as it prioritizes safety over accuracy.

VII. Conclusions

In this paper, we propose a strategy of incorporating Machine Learning (ML) algorithms into the Lead Vehicle (LV) of a platoon to enable online training and real-time prediction of communication delays. The simulation results show that the proposed communication approach allows the LV to collect data during runtime and use it to train ML algorithms online. Our simulations also demonstrate that the prediction of communication delays using ML algorithms in dense traffic scenarios exhibits significantly lower errors compared to the benchmark case of averaging the previously experienced delays. Furthermore, collisions are avoided for all simulation runs conducted in this paper when communication delays are predicted using the LSTM RNN algorithm and used in an emergency braking scenario. In contrast, a collision occurs when emergency braking is performed using average communication delays. Moreover, collisions occur in most cases when a platoon performs emergency braking in a dense traffic scenario with the normal braking strategy, and no measures are taken to mitigate the effects of communication delays. The findings of this paper highlight the importance of considering communication delays in platooning safety, particularly in dense traffic scenarios. The results show that averaging the previous delays or using normal braking is insufficient to assure platoon safety. However, a platoon exhibits safe behavior during emergency braking by predicting communication delays using ML and the communication strategy proposed in this paper. Therefore, we can conclude that ML-based real-time predictions of communication, network, and traffic parameters hold great potential for facilitating safety-critical decisions in platooning.

VIII. Acknowledgement

This work was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska - Curie grant agreement No 764951, the Basque Government's Research Group Program under grant No. IT1451-22, the AUTOEV@L (ref KK-2021/00123), and the Swedish Foundation for Strategic Research through the Serendipity project.

References

[1] S. Hasan, S. Girs, and E. Uhlemann, "Characterization of transient communication outages into states to enable autonomous fault tolerance in vehicle platooning," *IEEE OJ-ITS*, vol. 4, pp. 101–129, 2023.

[2] A. Brown, E. Cullen, J. Wu, M. Brackstone, D. Gunton, and M. McDonald, "Vehicle to vehicle communication outage and its impact on convoy driving," in *Proc. IEEE IV 2000*, pp. 528–533, Dearborn, MI, USA, 2000.

[3] L. Torres-Figueroa, H. F. Schepker, and J. Jiru, "Qos evaluation and prediction for c-v2x communication in commercially-deployed lte and mobile edge networks," in *Proc. IEEE VTC2020-Spring*, pp. 1–7, Antwerp, Belgium, 2020.

[4] S. Barmpounakis, N. Maroulis, N. Koursioumpas, A. Kousaridas, A. Kalamari, P. Kontopoulos, and N. Alonistioti, "AI-driven, QoS prediction for V2X communications in beyond 5G systems," *Comp. Net.*, vol. 217, p. 109341, 2022.

[5] D. C. Moreira, I. M. Guerreiro, W. Sun, C. C. Cavalcante, and D. A. Sousa, "QoS Predictability in V2X Communication with Machine Learning," in *Proc. IEEE VTC2020-Spring*, pp. 1–5, Antwerp, Belgium, 2020.

[6] W. Zhang, M. Feng, M. Krunz, and H. Volos, "Latency Prediction for Delay-sensitive V2X Applications in Mobile Cloud/Edge Computing Systems," in *Proc. IEEE GLOBECOM 2020*, pp. 1–6, Taipei, Taiwan, 2020.

[7] P. Fernandes and U. Nunes, "Platooning With IVC-Enabled Autonomous Vehicles: Strategies to Mitigate Communication Delays, Improve Safety and Traffic Flow," *IEEE T-ITS*, vol. 13, no. 1, pp. 91–106, 2012.

[8] S. Hasan, A. Balador, S. Girs, and E. Uhlemann, "Towards emergency braking as a fail-safe state in platooning: A simulative approach," in *Proc. IEEE VTC2019-Fall*, pp. 1–5, Honolulu, HI, USA, 2019.

[9] S. Barmpounakis, L. Magoula, N. Koursioumpas, R. Khalili, J. M. Perdomo, and R. P. Manjunath, "Lstm-based qos prediction for 5g-enabled connected and automated mobility applications," in *Proc. IEEE 5GWF 2021*, pp. 436–440, Montreal, QC, Canada, 2021.

[10] M. I. Khan, F.-X. Aubet, M.-O. Pahl, and J. Härri, "Deep learning-aided application scheduler for vehicular safety communication," *arXiv preprint arXiv:1901.08872*, 2019.

[11] M. Sangare, S. Banerjee, P. Muhlethaler, and S. Bouzefrane, "Predicting transmission success with support vector machine in vanets," in *Proc. IFIP/IEEE PEMWN 2018*, pp. 1–6, 2018.

[12] K. Bilstrup, E. Uhlemann, E. Ström, and U. Bilstrup, "On the ability of the 802.11 p mac method and stdma to support real-time vehicle-to-vehicle communication," *Eurasip J. Wirel. Commun. Netw.*, vol. 2009, no. 1, p. 902414, 2009.

[13] V. Alarcon-Aquino and J. A. Barria, "Multiresolution fir neural-network-based learning algorithm applied to network traffic prediction," *IEEE Trans Syst Man Cybern C Appl Rev*, vol. 36, no. 2, pp. 208–220, 2006.

[14] F. Tang, B. Mao, N. Kato, and G. Gui, "Comprehensive survey on machine learning in vehicular network: technology, applications and challenges," *IEEE Commun. Surv. Tutor.*, 2021.

[15] J. Ma, J. Theiler, and S. Perkins, "Accurate on-line support vector regression," *Neural computation*, vol. 15, pp. 2683–703, 11 2003.

[16] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, "Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons," *IEEE TCST*, vol. 8, no. 4, pp. 695–708, 2000.

[17] C. Bergenhem, K. Meinke, and F. Ström, "Quantitative safety analysis of a coordinated emergency brake protocol for vehicle platoons," in *Proc. ISoLA 2018*, pp. 386–404, Springer, 2018.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR 2015*, San Diego, CA, USA, 2015.

[19] F. Parrella, "Online support vector machines for regression," *Thesis for the degree of Info. Science, Dept. of Info. Science, University of Genoa, Italy*, 2007.

[20] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. Lo Cigno, "PLEXE: A Platooning Extension for Veins," in *Proc. IEEE VNC'2014*, pp. 53–60, Paderborn, Germany, 2014.

[21] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE TMC*, vol. 10, pp. 3–15, Jan 2011.

[22] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using sumo," in *Proc. IEEE ITSC'2018*, pp. 2575–2582, Maui, HI, USA, 2018.