JOURNAL OF MULTIDISCIPLINARY & TRANSLATIONAL RESEARCH

Journal of Multidisciplinary and Translational Research (JMTR)

journal homepage: https://journals.kln.ac.lk/jmtr/

Link Lifetime Approximation in Vehicular Communication using a Comparative Movement and Vehicle Control Maneuver-driven Technique

Patikiri Arachchige Don Shehan Nilmantha Wijesekara[1]*

[1]*Department of Electrical and Information Engineering, Faculty of Engineering, University of Ruhuna, Galle 80000, Sri Lanka*

Abstract

High mobility and rapid topological changes are characteristics of vehicular networks. As a result, in contrast to other networks, the connections between the automobiles are erratic and only last a brief period of time. Therefore, understanding the connection lifespan is crucial for creating effective communication between vehicles without packet losses. Only sensor-based measurements have been taken into account in previous efforts in this field to forecast link lifetimes. However, as they lack knowledge about the vehicle's expected future behavior, we anticipate that estimating link lifetimes solely from sensor data may result in less accurate estimates. We suggest using throttle and steering angle driving action outputs in conjunction with brake components to improve sensor readings and provide more futuristic relative motion information in order to address this issue. In particular, we use throttle changes to calculate jerk and integrate and combine the sensor signals to calculate average acceleration and velocity values. We then calculate new motion components, taking into account the steering angle change when it varies from the prior timestep. We suggested modeling the connection lifetime prediction problem through optimization, taking into account the relative velocity of the automobiles, adding jerk, and modifying with driving outputs. But because of that method's high computational cost, we also suggest a suboptimal method based on deep neural networks (DNN) to lower the computational difficulty. The proposed model is simulated utilizing NS3 for vehicular data transmission and CARLA for self-driving by leveraging a pre-trained driving model. The findings demonstrate that, in comparison to current methods, the link lifetime forecasts of the suggested models are substantially closer to actual link durations; as a consequence, the suggested method may be applied to enhance vehicular communication.

Keywords: Link lifetime, optimization, vehicular communication, vehicle control maneuver.

Article info

Article history:
Received 30th April 2025
Received in revised form 25th May 2025
Accepted 15th June 2025
Available online 30th June 2025

ISSN (E-Copy): ISSN 3051-5262
ISSN (Hard copy): ISSN 3051-5602
Doi: https://doi.org/10.4038/jmtr.v10i1.24
ORCID iD: https://orcid.org/0000-0002-3045-1596
*Corresponding author:
E-mail address: nilmantha@eie.ruh.ac.lik
(P.A.D.S.N. Wijesekara)

Introduction

The driving decisions in an intelligent transportation system may be used to inform decisions in the vehicular network as autonomous driving predictions show how probable the automobile is to perform in near-future timesteps (Zhang & Lu, 2020). The output of autonomous driving in an intelligent vehicular network can be applied to prediction tasks in vehicular communication in addition to driving (Zhang et. al., 2018). As a result, it can connect vehicle communication with autonomous driving. Therefore, to enhance vehicle communication and driving performance, precise driving action estimation is necessary. Existing research studies relevant to link lifetime estimation do not examine future vehicle behavior and do not account for jerk in approximation of link lifetime; instead, they use current timestep sensor data to forecast lifetime (Wijesekara et. al., 2023; Sudheera et. al., 2019). In software-defined vehicular networks, where the centralized controller may make choices like routing improvements based on the network's link lifespan information, link lifetime estimate is very helpful (Cardona et. al., 2020). Furthermore, link lifetime estimations may be used to make distributed choices in vehicular ad hoc networks, allowing for the improvement of those networks through precise link lifetime projections (Yan and Olariu, 2011). As the future paths of the cars may alter as a result of changes in autonomous/manual driving outputs, we anticipate that this may make projections less accurate.

Assuming reasonably stable or statistically modeled link durations, Geng et al. (2024) and Almuseelem et al. (2025) concentrated on resource optimization and network performance with delay/energy tradeoffs. However, predictive link lifetime estimation based on vehicle dynamics isn't specifically highlighted in those works. Although they do not specifically focus on link lifetime prediction algorithms, researchers such as Nkenyereye et al. (2019) and Shah et al. (2020) have addressed problems in software-defined vehicular networks (SDVN), where efficient link lifetime estimation is crucial for different tasks such as routing and control decisions. Link lifetime estimates in these systems are frequently abstracted or simplified without taking jerk effects or sophisticated motion dynamics into account. Instead of predicting the lifetime of a communication link, Li et al. (2020) have concentrated on traffic flow stability. They did not incorporate autonomous driving outputs (steering, throttle) for link predictions. However, they did use empirical jerk for vehicle following.

Current link lifetime prediction techniques ignore future autonomous driving system vehicle behavior in favor of only using current sensor data. Vehicle jerk dynamics and driving action outputs have never been integrated for communication prediction in any previous work. To the best of our knowledge, this study is the only study that utilizes autonomous driving outputs to forecast link lifetimes.

Consequently, the driving output is utilized in this work to enhance the connection lifespan forecasts for vehicle communication. In particular, the outputs of the autonomous driving model are used to precisely update the vehicle dynamics sensor readings, which are then fed into the link lifespan prediction model that accounts for jerk to provide precise forecasts. To lower the computing complexity, we provide a simpler suboptimal method albeit with lower accuracy utilizing deep learning and treat the prediction problem as a non-linear optimization task. In particular, more precise and future relative motion optimized by driving outputs is incorporated to determine the time to complete remaining relative displacement to the maximum transmission distance.

The following succinctly describes the novelty of this study. First, this study integrated sensor data with autonomous driving outputs (steering, throttle) for link prediction. To increase prediction accuracy, a new jerk-aware motion model was developed. Finally, two link lifespan estimation models with variable accuracy and computational efficiency are suggested using optimization and DNN.

This study crosses several fields, like computer science by designing and optimizing DNN; telecommunications by studying vehicular communication aspects and link prediction; automotive engineering by combining jerk modeling and vehicle dynamics; and transportation engineering by combining autonomous driving elements. Thus, the work addresses complex issues in intelligent vehicular networks in a comprehensive, multidisciplinary manner (Zhang et al., 2020; Tan et al., 2022).

Methodology

Link lifetime estimation

Figure 1 visually represent the variables influencing link lifetime.

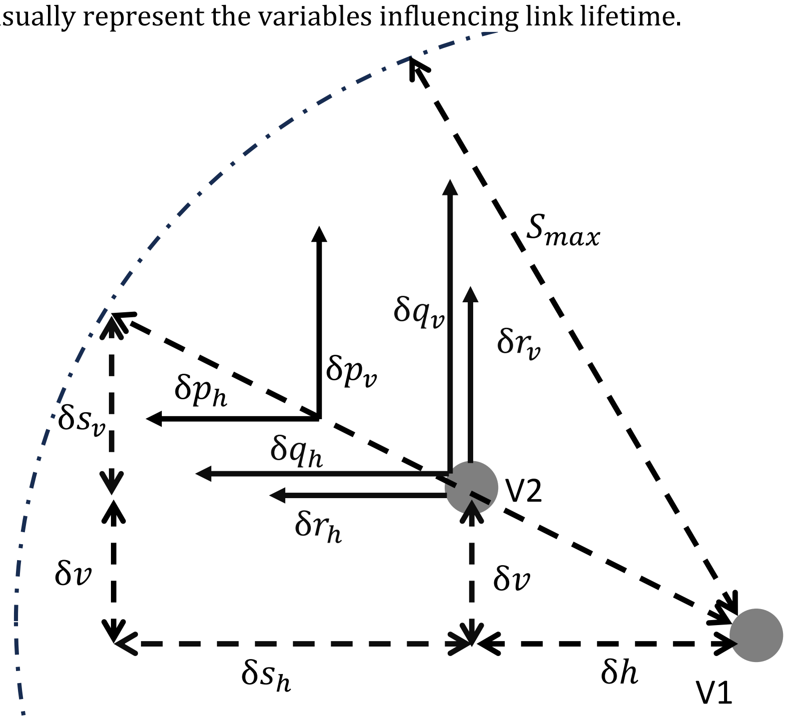

Figure 1. Link lifetime computation concept.

Relative displacement (δh, δv), relative velocity ($\delta p_h, \delta p_v$), relative acceleration ($\delta q_h, \delta q_v$), relative jerk ($\delta r_h, \delta r_v$), and highest transmission range (S_{max}) are the factors that impact link lifetime, as illustrated graphically in Figure 1.

$$(\delta h)^2 + (\delta v)^2 + 2(\delta h\delta p_h + \delta v\delta p_v)t + (\delta h\delta q_h + \delta v\delta q_v + (\delta p_h)^2 + (\delta p_v)^2)t^2 + \Big(\delta p_h\delta q_h + \delta p_v\delta q_v + \tfrac{1}{3}(\delta r_h\delta h + \delta r_v\delta v)\Big)t^3 + \Big(\tfrac{1}{4}((\delta q_h)^2 + (\delta q_v)^2) + \tfrac{1}{3}(\delta r_h\delta p_h + \delta r_v\delta p_v)\Big)t^4 + \tfrac{1}{6}(\delta r_h\delta q_h + \delta r_v\delta q_v)t^5 + \tfrac{1}{36}((\delta r_h)^2 + (\delta r_v)^2)t^6 \le S_{max}{}^2 \quad (1)$$

Given the restriction in inequality (1), the link lifetime calculation can thus be represented as an optimization assignment to maximize *t*. To replace the optimization problem, the DNN shown in Figure 2 was developed and built. DNN was used to reduce computational complexity, as the inequality in (1) is of order 6 and is thus computationally inefficient.

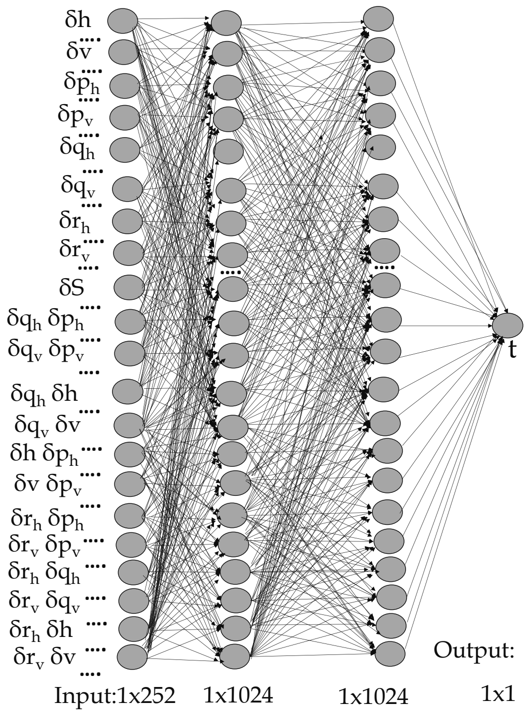

Figure 2. Proposed DNN for link lifetime estimation. The input layer contains 252 (21x12) neurons. Connections between layers are indicated with standard arrows and layer names are labeled.

DNN has two hidden layers of size 1024 (Figure 2). The solution is to learn to predict link lifetime as a regression challenge. When the link lifespan forecasting paradigm is centralized, as in the case of an SDN, batch predictions can be used to make multiple predictions with a short inference time. Considering the maximum value of inequality (1), it is a sixth-order polynomial. Because of the multiplication of vector magnitudes, the neural network needs 1–12th order values of the vectors to obtain the best approximation of the link lifetime. As a result, the input layer contains 252 (21x12) neurons.

Impact of throttle

The vehicle's force is in direct relation to the throttle value (F) (Pourciau, 2006). Equation (2) thus provides the link between the throttle and jerk (r).

$$\lambda(\Delta F) = Mr \qquad (2)$$

M is the vehicle's mass, ΔF is the change in throttle value, and λ is a constant in Equation (2). Velocity, jerk, and acceleration are all collinear while the steering angle is constant; centripetal acceleration may be disregarded because it has no bearing on link lifespan prediction. Let β be the angle that a vector makes with the horizontal axis, and let A be such a generic vector. Then, $r_h = rcos(\beta)$ and $r_v = rsin(\beta)$.

Let ΔT be the constant jerk time. Equation (3) provides the average anticipated acceleration (q) when q_{sen} is the acceleration that the vehicle senses.

$$|q| = |q_{sen}| + \frac{|r|\Delta T}{2} \qquad (3)$$

Likewise, the average velocity value (p) may be expressed using Equation (4).

$$|p| = |p_{sen}| + \frac{|q|\Delta T}{2} + \frac{|r|\Delta T^2}{4} \qquad (4)$$

Impact of steering angle

If the throttle and steering angle are both altered, the throttle-related vectors should be calculated first, followed by the steering-angle-based calculations.

The vector alignment before (grey color) and after (black color) the steering angle ($\Delta\Omega$) modification is depicted in Figure 3.

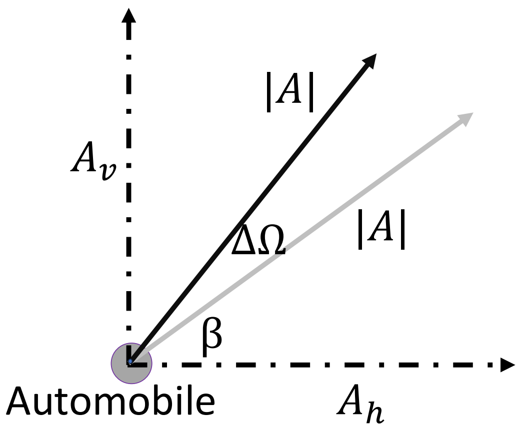

Figure 3. Steering angle impact. A is a vector that is an output vector computed for throttle adjustments.

Let $A_{h,old} = |A|\cos(\beta)$ and $A_{v,old} = |A|\sin(\beta)$ be the h and v elements of the vector A by not considering the impact of steering angle rotation. Then, $A_{h,new} = \cos(\beta + \Delta\Omega)$ and $A_{v,new} = |A|\sin(\beta + \Delta\Omega)$ are the forthcoming vectors examining the impact of steering angle rotation. Thus, $A_{h,new}$ and $A_{v,new}$ can be deduced as shown in Equations (5) and (6).

$$A_{v,new} = |A|\sin(\beta + \Delta\Omega) = A_{v,old}\cos(\Delta\Omega) + A_{h,old}\sin(\Delta\Omega) \quad (5)$$

$$A_{h,new} = |A|\cos(\beta + \Delta\Omega) = A_{h,old}\cos(\Delta\Omega) - A_{v,old}\sin(\Delta\Omega) \quad (6)$$

The vectors acceleration, jerk, and velocity are all part of A. Only the vectors specified above have an impact on the vehicle's future trajectory because the relative displacement has already been calculated for the current time stamp.

Results

Self-driving scenarios were conducted in client-server mode and pedestrians and traffic were simulated using CARLA (Wijesekara, 2022). In CARLA, different towns which included highways, urban, and sub-urban traffic conditions were selected to simulate different traffic scenarios. The cars' driving action output (throttle and steering) and mobility data were then extracted and sent to the NS3 (Riley and Henderson, 2010) for real-time communication. Traffic flow and driver behavior parameters were derived from recent autonomous driving studies to validate ground dataset fidelity in verified vehicle movement patterns (Terapaptommakol et al., 2022). NS3 merely simulates automobiles because there are only linkages between them. GurobiPy commercial program was utilized for optimization.

Performance analysis metrics

Link lifetime Mean Absolute Error (MAE), which was calculated using Equation (7), as the measure to assess the performance of the link lifetime prediction model was utilized first.

$$MAE = \frac{1}{B}\sum_{i=1}^{B}\frac{1}{T}\sum_{t=1}^{T}\frac{1}{J}\sum_{a=1}^{J}\frac{1}{J}\sum_{b=1}^{J}\left|T_{pitab} - T_{gitab}\right| \quad (7)$$

T is the link lifespan, a and b are indices of nodes, i is the test scenario index, p is the projected value, and g is the ground truth in Equation (7). Additionally, J is the total amount of nodes, B denotes the total number of experiments, T notates the aggregate count of temporal steps, and t is the temporal step. Thus, for vehicle pairs a and b at time t for test scenario i, T_{pitab} and T_{gitab} stand for the expected and ground truth link lifespan values, respectively, in Equation (7).

Then, the performance of a link lifetime-driven routing technique was evaluated (Wijesekara and Gunawardena, 2023) using the packet delivery ratio (PDR), as given in Equation (8).

$$PDR = \frac{1}{D}\sum_{i=1}^{D}\frac{1}{F}\sum_{f=1}^{F}\frac{1}{N_{if}}\sum_{k=1}^{N_{if}}\lambda_{ifk} \qquad (8)$$

In Equation (8), index i represents the i^{th} routing cycle, index f represents the f^{th} flow, D is the aggregate count of routing cycles, F is the overall count of flows, λ_{ifk} is the event that the k^{th} packet of the f^{th} flow in the i^{th} routing cycle is delivered at the destination, and N_{if} is the aggregate quantity of packets of the f^{th} flow.

Performance analysis

A 673, 000-item dataset was created by optimizing the differential motion variables and driving motions for the purpose of training the DNN for link lifetime estimation (Wijesekara, 2025). When data is normalized within a range of 100 s, the training mean square error (MSE) of the link lifetime prediction DNN model was as small as 10^{-7}, or 33 ms. Prior to entering the data into the DNN, the values were normalized in the scale [-1, 1] after computing terms up to the 12^{th} order during preprocessing. The data were normalized by using 100 s, because the average link lifetime was around 11 s with a 95% confidence interval upper limit way below 100 s. By examining the training curve, the number of hidden tiers and neurons in each of them was determined experimentally. Adam (Ogundokun, 2022) was employed as the optimizer, MSE (Jin and Montúfar, 2023) as the loss function, and ReLU (Lin & Shen, 2018) was employed as the inner and output layer activation function.

The Adam optimizer was used to train the DNN with momentum parameters β_1 = 0.9 and β_2 = 0.999 and a learning rate of 0.001. Training was conducted using a batch size of 256 over a maximum of 200 epochs using the MSE loss function. To avoid overfitting, early stopping was applied with a 20-epoch patience. The dataset was divided into three parts for validation as 10% for testing, 10% for validation, and 80% for training. To make sure that the performance of the model was robust, a 5-fold cross-validation was also carried out. First, the performance of the model was assessed on the held-out test set by using the proper metrics. Then, a grid search was used for hyperparameter tuning in some variables like hidden layer sizes, which had values of 256, 512, and 1024; dropout rates, which had values of 0.1, 0.2, and 0.3; and learning rates, for which values of 0.0001, 0.001, and 0.01 were tested. Finally, the model was implemented using TensorFlow, and the DNN model could achieve an average inference time of 8 milliseconds per prediction, with a total training time of roughly 110 minutes on an NVIDIA RTX 3080 GPU, which was used to train and test the model.

The performance of the suggested DNN-driven link lifetime estimation (proposed DNN) and the suggested optimization-driven link lifetime estimation (proposed optimization) was contrasted with the most advanced sensor-driven link lifetime estimations (sensor-driven machine learning (Wijesekara, 2023) and sensor-driven optimization (Sudheera et. al., 2019)). The 95% confidence intervals shown by the error bars in Figures 4 and 5 are calculated as mean ± 1.96 × SEM, where SEM is the standard error of the mean over total simulation runs using various random seeds.

Exact link lifetime analysis

Using the driving behaviors that the self-driving model predicts, the sensor values were modified in this experiment. Predicted lives are then documented when the predictions are acquired at various timesteps during the experiment. Additionally, the timestamps were recorded when a link expires and when a new link is made. As it may happen in a real-world vehicle network, transmission power in this experiment was set to a fixed value to ensure that the maximum transmission distance is maintained as a constant. The number of vehicle nodes were changed from two to ten to do 100 experimental runs. Lastly, the MAE was computed, and Figure 4 displays the outcomes. The mean link lifetime for different numbers of vehicles ranged from 5.4 s to 11.3 s, with their standard deviations ranging from 1.7 s to 3.8 s. The 95% confidence interval maximum value is thus quite below 100 s. So, 100 s was used to normalize the link lifetimes by considering 100 s as the upper limit in a worst-case scenario.

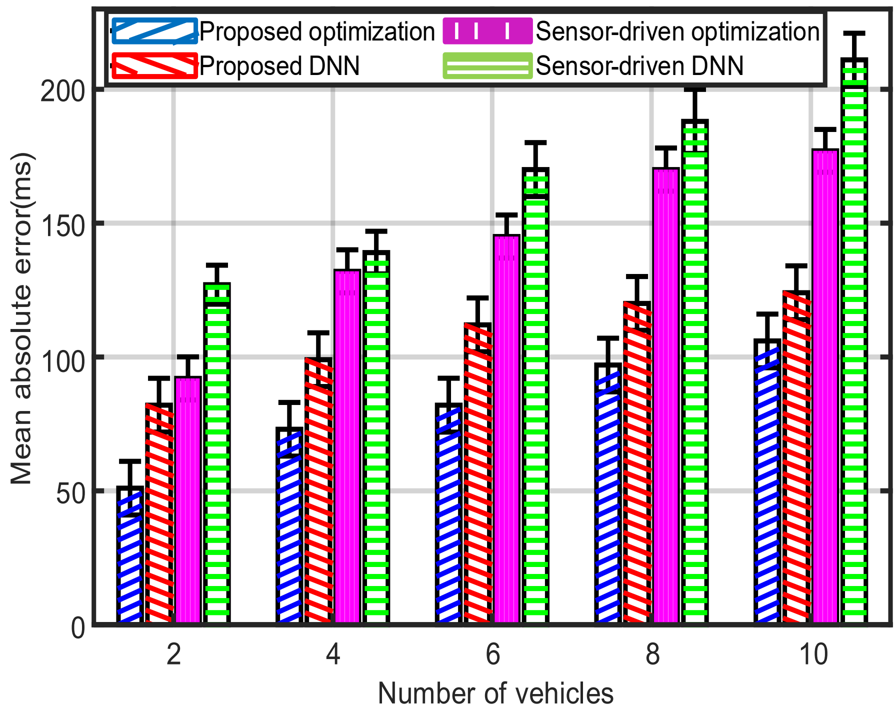

Figure 4. Link lifetime estimation performance assessment.

Under any number of automobile nodes, the MAE of both suggested models is lower than that of state-of-the-art methods, as shown in Figure 4. Estimations are more correct than those based on present sensor readings because the suggested models account for jerk and modify differential sensor readings based on autonomous driving actions. However, all the prediction abilities of the model decline as vehicle density increases.

This is because, as the number of automobiles rises, the relative motion becomes more dynamic, causing the motion of the vehicles to vary more often than it would in a scenario with fewer people. This outcome demonstrates the superiority of the proposed link lifetime estimation method over current ones.

Link-lifetime-driven routing evaluation

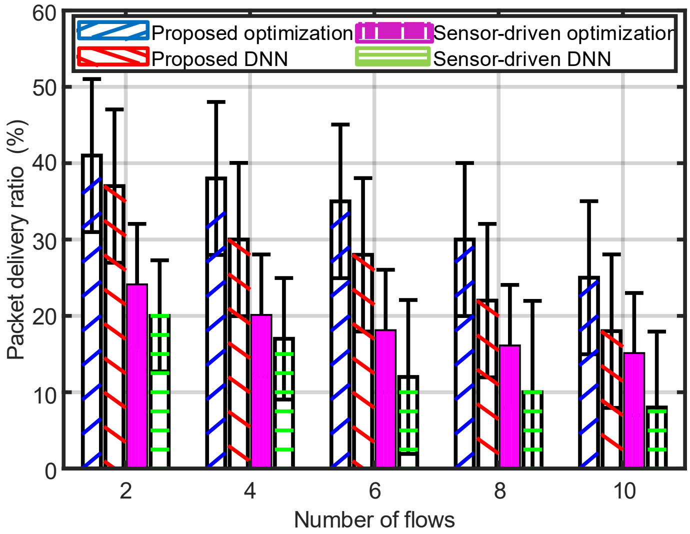

Figure 5. Link lifetime-driven routing performance assessment.

In this experiment, different numbers of packets from random sources were sent to random destinations in each routing cycle, where the number of flows is changed in each routing experiment. The number of nodes in this experiment were fixed at 10. For the link lifetime-driven routing (Wijesekara and Gunawardena, 2023), each of the precedingly used frameworks were separately used for link lifetime estimation. The results of this experiment are shown in Figure 5.

As demonstrated in Figure 5, under any number of flows, the PDR of the proposed optimization-driven link lifetime-based routing is higher than when any other technique is used for link lifetime estimation. The reason for the preceding behavior is that due to accurate link lifetime estimations, the PDR of routing is enhanced. The performance is then followed by proposed DNN-driven link lifetime-based routing, sensor-driven optimization-based routing, and sensor-driven DNN-based routing. Moreover, with the increment of the number of flows, the PDR of all techniques decreases due to the increment of contention in the network due to the existence of multiple parallel flows. However, the previously mentioned trend is still prevalent irrespective of the number of flows. Thus, it can be concluded that the proposed link lifetime estimation model improves link lifetime-driven routing applications.

Discussion

Optimization problem of the inequality (1) is the solution of a sixth-order polynomial inequality, complexity of which scales as $O(n^6)$, where n is the number of vehicles. On the other hand, regardless of the number of vehicles, the suggested DNN inference entails a fixed number of matrix multiplications with complexity $O(k)$, where k is the number of total neurons. According to the benchmarks used in this study, the average DNN inference latency for the tested scenario (up to 10 vehicles) is roughly 8 ms, whereas each instance of numerical optimization takes 45 to 120 ms. This validates the computational efficiency of the suboptimal DNN approach by achieving roughly 5.6 times faster execution with only approximately a 31.2% relative increase in MAE (2.55% normalized to a 1000 ms maximum error). Also, the computational complexity of DNN is lower due to its batch predictions, while that of the optimization approach is higher with the increment of the number of nodes, as it does not have such capability.

Ablation studies were performed comparing model accuracy with and without jerk consideration to validate the significance of including jerk in link lifetime prediction. The results showed approximately a 23% error reduction compared to sensor-driven optimization when jerk is included. Empirical results are reflected in the jerk ranges used in CARLA simulations: normal driving jerk values range from 0.5 to 3.2 m/s^3, while emergency maneuvers can reach up to 8.5 m/s^3 (Li et al., 2020). To guarantee realistic throttle and brake profiles, simulation parameters were adjusted in comparison to actual driving datasets (Khan et al., 2019). As a result, jerk incorporation captures significant dynamic vehicle behavior that influences link stability.

The suggested optimization method consistently outperforms competing techniques across vehicle counts of 2 to 10 nodes, according to the link lifetime estimation results. On average, the suggested optimization reduces the MAE by about 25.9% when compared to the suggested DNN model, 43.1% when compared to sensor-based optimization, and 51.5% when compared to sensor-based DNN techniques. These improvements are statistically significant ($p < 0.001$), according to paired t-test analyses for the five node count settings, with magnitudes of t-values exceeding 9 across comparisons. These results confirm that combining jerk-aware optimization with autonomous driving outputs can significantly increase the accuracy of link lifetime prediction in dynamic vehicular scenarios.

Conclusion

This article introduced a unique method for link lifetime estimate in automotive networks by enhancing sensor-driven motion readings with the addition of drive outputs towards futuristic motion instead of sensor readings in the current timestep. This study presents a low-complexity, suboptimal deep neural network to replace the optimization to simulate the problem, including relative jerk. By employing autonomous driving outputs to alter differential motion parameters in future (incorporating how the motion of the vehicle may change in the near future using throttle and steering angle changes), the accuracy of the link lifespan forecast was demonstrated to be greater than that of conventional sensor-based predictions. Specifically, the suggested optimization technique shows statistically significant performance improvements over both deep learning and optimization baselines and reduces link lifetime prediction errors by up to 51.5% in comparison to traditional sensor-based techniques, indicating its potential to improve vehicular communication reliability. Consequently, the suggested model may be used to estimate link lifetimes and offer precise information on link lifetime to enhance communication in smart vehicular networks.

Acknowledgement

The authors would like to acknowledge the authors of optimization-based link lifetime prediction for kindly providing their model.

Conflict of interest statement

The authors declare no conflict of interest.

References

Almuseelem, W. (2025). Deep reinforcement learning-enabled computation offloading: A novel framework to energy optimization and security-aware in vehicular edge-cloud computing networks. *Sensors, 25*(7), 2039. https://doi.org/10.3390/s25072039

Cardona, N., Coronado, E., Latré, S., Riggio, R., & Marquez-Barja, J. M. (2020). Software-defined vehicular networking: Opportunities and challenges. IEEE Access, 8, 219971–219995. https://doi.org/10.1109/ACCESS.2020.3043039

Geng, J., & Jin, S. (2025). Computation offloading in MEC-assisted vehicular networks with task migration and result feedback. Ad Hoc Networks, 158, 103936. https://doi.org/10.1016/j.adhoc.2025.103936

Jin, H., & Montúfar, G. (2023). Implicit bias of gradient descent for mean squared error regression with two-layer wide neural networks. Journal of Machine Learning Research, 24(137), 1–97.

Khan, M. I., Aubet, F. X., Pahl, M. O., & Härri, J. (2019). Deep learning-aided application scheduler for vehicular safety communication. arXiv. https://arxiv.org/abs/1901.08872

Li, T., Hui, F., Liu, C., Zhao, X., & Khattak, A. J. (2020). Analysis of V2V messages for car-following behavior with the traffic jerk effect. Journal of Advanced Transportation, 2020, 9181836. https://doi.org/10.1155/2020/9181836

Lin, G., & Shen, W. (2018). Research on convolutional neural network based on improved ReLU piecewise activation function. Procedia Computer Science, 131, 977–984. https://doi.org/10.1016/j.procs.2018.04.291

Nkenyereye, L., Islam, S. R., Choi, Y. H., Bilal, M., & Jang, J. W. (2019). Software-defined network-based vehicular networks: A position paper on their modeling and implementation. Sensors, 19(17), 3788. https://doi.org/10.3390/s19173788

Ogundokun, R. O., Maskeliunas, R., Misra, S., & Damaševičius, R. (2022). Improved CNN based on batch normalization and Adam optimizer. In O. Gervasi et al. (Eds.), Computational science and its applications – ICCSA 2022 (pp. 593–604). Springer. https://doi.org/10.1007/978-3-031-10547-5_43

Pourciau, B. (2006). Newton's interpretation of Newton's second law. Archive for History of Exact Sciences, 60(2), 157–207. https://doi.org/10.1007/s00407-005-0101-4

Riley, G. F., & Henderson, T. R. (2010). The ns-3 network simulator. In W. Wei & S. M. Fdida (Eds.), Modeling and tools for network simulation (pp. 15–34). Springer. https://doi.org/10.1007/978-3-642-12331-3_2

Shah, S. D. A., Gregory, M. A., & Li, S. (2020). Software-defined vehicular networks: A cooperative approach for computational offloading. In 2020 30th International Telecommunication Networks and Applications Conference (ITNAC) (pp. 1–3). IEEE. https://doi.org/10.1109/ITNAC50341.2020.9314995

Sudheera, K. L. K., Ma, M., & Chong, P. H. J. (2019). Link stability-based optimized routing framework for software-defined vehicular networks. *IEEE Transactions on Vehicular Technology, 68*(3), 2934–2945. https://doi.org/10.1109/TVT.2019.2890839

Terapaptommakol, W., Phaoharuhansa, D., Koowattanasuchat, P., & Rajruangrabin, J. (2022). Design of obstacle avoidance for autonomous vehicle using deep Q-network and CARLA simulator. World Electric Vehicle Journal, 13(12), 239. https://doi.org/10.3390/wevj13120239

Wijesekara, P. A. D. S. N. (2022). Deep 3D dynamic object detection towards successful and safe navigation for full autonomous driving. The Open Transportation Journal, 16(1), e2208191. https://doi.org/10.2174/18744478-v16-e2208191

Wijesekara, P. A. D. S. N., Sudheera, K. L. K., Sandamali, G. G. N., & Chong, P. H. J. (2023). Machine learning based link stability prediction for routing in software-defined vehicular networks. In *20th Academic Sessions* (p. 60).

Wijesekara, P. A. D. S. N., & Gunawardena, S. (2023). A machine learning-aided network contention-aware link lifetime-and delay-based hybrid routing framework for software-defined vehicular networks. Telecom, 4(3), 393–458. https://doi.org/10.3390/telecom4030022

Wijesekara, P. A. D. S. N. (2025). A relative motion and driving action-based link lifetime estimation model for vehicular networks. In 22nd Academic Sessions (p. 33).

Yan, G., & Olariu, S. (2011). A probabilistic analysis of link duration in vehicular ad hoc networks. IEEE Transactions on Intelligent Transportation Systems, 12(4), 1227–1236. https://doi.org/10.1109/TITS.2011.2158594

Zhang, H., & Lu, X. (2020). Vehicle communication network in intelligent transportation system based on Internet of Things. Computer Communications, 160, 799–806. https://doi.org/10.1016/j.comcom.2020.06.022

Zhang, S., Chen, J., Lyu, F., Cheng, N., Shi, W., & Shen, X. (2018). Vehicular communication networks in the automated driving era. IEEE Communications Magazine, 56(9), 26–32. https://doi.org/10.1109/MCOM.2018.1800071

Zheng, Y., Cui, L., Tso, F. P., Li, Z., & Jia, W. (2024). DNN acceleration in vehicle edge computing with mobility-awareness: A synergistic vehicle–edge and edge–edge framework. Computer Networks, 251, 110607. https://doi.org/10.1016/j.comnet.2022.110607