

A New Data-Driven Approach For On-line Traffic Participant Behaviour Prediction at Intersections for Automated Driving

1st Mehran Zamani Abnili
Mechanical and Mechatronics Engineering
University of Waterloo
Waterloo, Canada
mzamania@uwaterloo.ca

2nd Nasser L. Azad
Systems Design Engineering
University of Waterloo
Waterloo, Canada
nlashgar@uwaterloo.ca

Abstract—With the developments in autonomous driving and the popularity, the subject has gained over the past few years in the scientific communities, especially considering that most of the traffic accidents are due to recognition errors and perception being neglected in the literature, this study aimed to develop a prediction method and employed it in intersection driving scenario. Being a data-driven approach, a simulation was set up in SUMO, and a combination of Dynamic Bayesian Network and Recurrent Neural Network were tasked to make predictions for the states of the ego vehicle and that of other traffic participants for 1, 2, 5, and 10-second horizons. A Kalman filter was used as a post-processing measure to ensure smooth transitions in the velocities especially in longer horizons. Results, disclosed for a random traffic participant pair in the data pool, exhibit valid predictions and competent accuracy.

Index Terms—Dynamic Bayesian Network, Recurrent Neural Network, Autonomous Driving, Perception

I. INTRODUCTION

Autonomous driving has become a point of interest among researchers in many different disciplines as the future of transportation lies heavily upon the discoveries and breakthroughs made today. Autonomous driving promises affordable long-range transportation to the public by replacing the human workforce and minimizing energy consumption; Provides individuals with disability a means for personal conveyance, minimizes harmful byproducts such as exhaust emissions in case of internal combustion engine-driven vehicles [1] and decrease power plant waste in case of their electric counterparts. Most importantly, autonomous vehicles are expected to minimize traffic accidents and thereupon the property, and tragically in some cases life and limb, associated with such occurrences. To emphasize the importance of the concern with traffic accidents, one must draw their attention to the statistics in this regard. According to a survey by U.S. National Highway Traffic Safety Administration (NHTSA) [2], in 2015, out of 2,189,000 accidents, $94\% \pm 2.2\%$ were caused by human error, and grievously, 35,092 were fatal [3].

The decision-making process for any autonomous mobile robotic system starts from the perception of the environment. These machines often use a battery of sensors to collect

data from the environment and need to process this data to be able to truly understand their surroundings. For humans, this process is often done semantically and automatically. A human driver does not bother with exact spacial measurements but always has a *somewhat* accurate sense of the situation. For machines, especially expert systems, lacking the human intellect, reason, and judgment, a framework needs to be introduced that encompasses the innumerable situations a human driver might find themselves in, which is practically quite unfeasible. The other approach is to let machines learn, and develop their judgment. The latter approach has seen a lot of interest, especially in recent years with the advancement of computer hardware, Although these techniques, theoretically, are not new.

The same document by NHTSA [2] recounts that most of the traffic accidents with human error element, were due to a recognition error ($41\% \pm 2.2\%$) followed by decision errors at $33\% \pm 3.7\%$ and execution errors at $11\% \pm 2.7\%$. Besides the fact that autonomous vehicles will have a gradual placement in the market and a transition phase is imminent and inevitable, autonomous vehicles must be able to learn and imitate the ideals for human drivers which humans fail to deliver. In terms of execution, machines have always been superior to humans in accuracy and efficiency. However, recognition and decision making, as a directly dependent variable, are the spotlight of research in this area. However, despite most of the traffic accidents being caused by a cognitive mishap, most of the research in this area is done either on decision making or controls exclusively, or the efforts for perception and control are done in conjunction and one module is expected to undertake both perception and controls. As for studies done on perception, *Zhang et al.* [4] studied “chaining neural networks” to make speed predictions considering V2X (Vehicle to Everything) communication, using VISSIM (Verkehr In Städten - SIMulationsmodell) data based on Wiedmann’s car-following model. *Moser et al.* [5] studied short-term vehicle velocity prediction considering V2X communication and using DBNs (Dynamic Bayesian Network). *Wang et al.* [6] developed a semantic method to classify driver behaviour based on their physical

and psychological thresholds. *Gindele et al.* [7] experimented with DBNs to predict driver behaviour on intersections. *Sarkar et al.* [8] combined DBN with random forest to predict the trajectory of traffic agents in an urban intersection. Given the large portion of accidents being caused by recognition errors and the general setting of autonomous driving being altogether the same as the conventional street network, this study, expands on [9] and [10] and focuses on developing a perception strategy for intersection driving. As mentioned in previous studies, sophisticated prediction strategies can improve the performance of MPCs quite significantly. Most MPC controllers in the literature make simplifying assumptions for the development of the states that are not completely reflective of real-life scenarios. For instance, [11] uses least-square parameter estimators for velocity prediction, [12] considers predefined speed profiles which impair the generalization of the proposed controller, and [13] makes the assumption that the states of all other traffic participants are fully known for the NMPC-based multi-lane adaptive cruise controller. Universally, it is arguable that all of these endeavours would either produce better results or be at all practical, with a decent prediction strategy. Although statistically, most of the traffic accidents are due to recognition and perception errors, it seems that this crucial aspect of automated driving is being neglected in the literature. In this study, a novel data-driven prediction strategy is developed and evaluated for real-time behaviour prediction in an intersection environment which can also be a complement to MPCs. In this paper, background is discussed briefly in section II, followed by proposed methodology in III, results in IV and finally conclusion and future work in V.

II. BACKGROUND

A. Dynamic Bayesian Network

Bayesian Networks are probabilistic graphical models of dependencies between variables. Manifested in the form of acyclic directed graphs, Bayesian networks are solid tools that can be used to find the probability of any variable taking a specific value, given the conditions of variables which have direct impact on said variable taking a value (e.g. the probability of an accident given the driver’s age, psychological conditions, time of the day, weather circumstances, etc.). In the case of Dynamic Bayesian networks, some of the variable nodes in the digraph also relate to another time-slice, commonly to the slice in the future. As a result, DBNs are powerful tools to predict the state of a variable for an arbitrary horizon. A useful mechanism that makes DBNs suitable for driving applications and turns them into machine learning techniques is the training procedure commonly referred to as the EM-algorithm (*Expectation-Maximization*) in machine learning domain, which is a direct descendant of the forward-backward algorithm. In EM-algorithm it is expected that every variable is a mixture distribution and can be represented as a set of Gaussian normal curves. EM-algorithm then recursively identifies the best fit of these bell curves on the dataset. For a Bayesian network, Y representing a variable and θ being the setting of its parents, log-likelihood \mathcal{L} can be defined as:

$$\mathcal{L}(\theta) = \log P(Y|\theta) = \log \sum_X P(Y, X|\theta) \quad (1)$$

By defining distribution Q as a lower bound on the log-likelihood \mathcal{L}

$$\begin{aligned} \log \sum_X P(Y, X|\theta) &= \log \sum_X Q(X) \frac{P(Y, X|\theta)}{Q(X)} \\ &\geq \sum_X Q(X) \log \frac{P(Y, X|\theta)}{Q(X)} \\ &= \sum_X Q(X) \log P(Y, X|\theta) \\ &\quad - \sum_X Q(X) \log Q(X) \\ &= \mathcal{F}(Q, \theta) \end{aligned} \quad (2)$$

EM algorithm alternates between maximizing \mathcal{F} with respect to Q and θ respectively, holding the other fixed [14].

$$\mathbf{E \ Step:} \quad Q_{k+1} \leftarrow \operatorname{argmax}_Q \mathcal{F}(Q, \theta_k) \quad (3)$$

$$\mathbf{M \ Step:} \quad \theta_{k+1} \leftarrow \operatorname{argmax}_\theta \mathcal{F}(Q_{k+1}, \theta) \quad (4)$$

Once fully trained, the Bayesian Network can produce a likelihood for variable Y taking any value, given its parents. Naturally, the topology of the DBN has to follow a few basic rules:

- The source nodes (nodes with in-degree of 0) must be reserved for variables that can be directly measured from the environment. In other words, source nodes act as inputs.
- The complexity of the problem, for each node, scales exponentially with the number of parents and the number of states each parent can take. Therefore one node must not have too many parents.
- The child-parent dependency must be conserved. Each node should have either a direct or an indirect correlation with its neighbours.

While DBNs are grand instruments for mixture distributions, it is obvious that they fall short when dealing with continuous variables that are common in traffic datasets, especially dynamic variables such as velocity and acceleration. While they can incorporate such variables semantically as mentioned before, the exact value cannot be predicted without the help of an alternative solution.

B. Recurrent Neural Network

One alternative solution for the shortcoming of DBNs in dealing with continuous variables is using neural networks. Classically neural networks can be used to find a relationship between the inputs and outputs of an unknown function and act as function approximation tool. This task is undertaken by a network of “primitive functions” [15] that introduce non-linearity to the sum of their weighted inputs and associate them with a precisely defined output. Being a data-driven approach, similar to DBNs, makes them a decent match to go alongside DBNs.

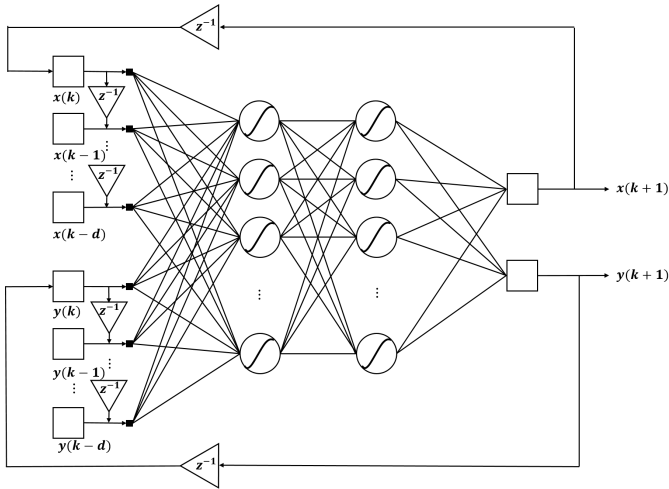


Fig. 1: Recurrent neural network schematics

Neural networks are plentiful in types, each specialized to a specific use case. To incorporate time-series and sequences of data necessary to extend prediction horizons to any desired value, recurrent neural networks are commonly chosen as the solution. The recurrent neural network can be fed a sequence of the history of a variable and be trained to predict the value of that variable for the next time-slice. After the training is completed, the neural network can recursively include its prediction as the ‘current’ state of its inputs and predict the values for the future time slices.

Fig. 1 illustrates the schematic of such network for two arbitrary variables x and y for a window of d history, predicting both variables in the next time slice ($k + 1$). The size of the history window (d), the length of each time-slice (sampling frequency), and the specifications of the network itself, such as the number of layers and number of nodes in each layer are design choices that need to be optimized for the application, but the trade-offs are clear.

III. METHODOLOGY

Driving in an intersection, especially in such a chaotic environment alongside other human drivers, pedestrians, cyclists and other traffic participants is a rigorous and critical task. One of the biggest differences between human drivers and hypothetical fully autonomous vehicles would be the cues humans can interpret from other traffic participants, coming to them in the form of hand signals, body language, high beam lights and occasional activation of hazard lights for gratitude. It is simply not feasible to teach all of that to a machine, but the same machine, blinded to all these cues, is expected to yield better performance than human drivers. This expectation can be realistic, relying on two basic principles.

- 1) The machine has more accurate measurements of the environment
- 2) The machine can execute control commands with more precision

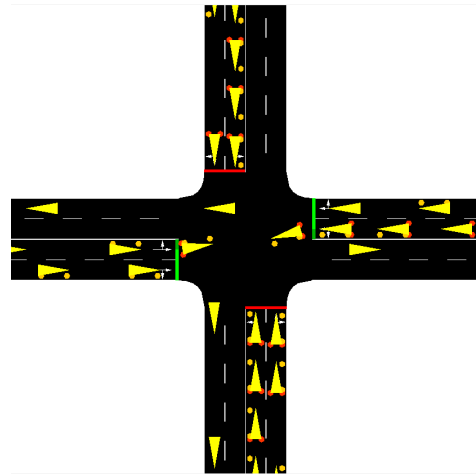


Fig. 2: Simulation environment

However, given these two assumptions, the perception for a vehicle is still a complex challenge and a humongous task. In this study a simulation was assembled in Simulation of Urban Mobility (SUMO) [16], to collect the data needed for the data-driven techniques mentioned in the background section. The simulation and the data derived from it were then used to train a combination of DBN and RNN, where the DBN handles discrete variables and mixture distributions, and the RNN takes care of continuous variables. Predictions can be made for each and every variable over an arbitrary horizon, however the DBN topology is a collocated follower to what is considered an input and an output. The forthcoming subsections discuss the properties of each component as well as their combination.

A. Simulation in SUMO

In SUMO, a controlled intersection (i.e. intersection controlled with traffic lights) environment was set up with two lanes on each side of each arm to incorporate lane changes as well as every possible traffic scenario, while maintaining manageable data size and flow densities. A total of 3600 vehicles were deployed in the environment taking random trips while conserving traffic rules. The traffic light signal schedule was chosen to be symmetric and simple without advanced left (as advanced left signal does not have any learning benefits). Fig. 2 illustrates the SUMO environment.

The data collected from the simulation was then fused with the map of the environment to create meaningful variables for the DBN.

B. Dynamic Bayesian Network

Considering the rules mentioned in section II, DBN variables were established as declared in table I. To create these variables, the only data collected from the simulation were position and traffic light signals, to be fed directly to the DBN, as well as velocity which is a continuous variable and was aimed to be handled by the RNN. Fig. 3 illustrates the topology of the proposed DBN. In this topology, the status of the traffic light, the presence of vehicles in the vicinity, current

ID	Variable	Set of states
01	<i>StoppedAtLine</i>	{True, False}
02	<i>LeftTurnRequiresStop</i>	{True, False}
03	<i>PathIntersects</i>	{True, False}
04	<i>CarFront</i>	{True, False}
05	<i>CarSide</i>	{True, False}
06	<i>CarBehind</i>	{True, False}
07	<i>ApproachingIntersection</i>	{True, False}
08	<i>AccelerationFlag</i>	{True, False}
09	<i>Lane</i>	{1, 2, Intersection}
10	<i>LanePrime¹</i>	{1, 2, Intersection}
11	<i>TrafficLightSignal</i>	{Green, Yellow, Red}
12	<i>TrafficSignalPrime</i>	{Green, Yellow, Red}
13	<i>CurrentSection</i>	{West, North, South, East, Intersection}
14	<i>SectionPrime</i>	{West, North, South, East, Intersection}

TABLE I: DBN variables

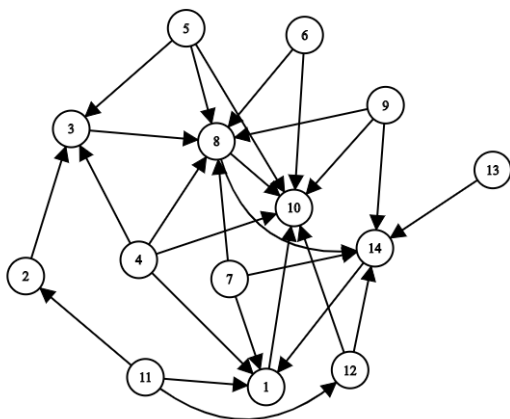


Fig. 3: DBN topology

lane, current section and whether the vehicle is approaching the intersection or departing from it are considered as inputs. Practically, these inputs can reliably be extracted from the environment, especially if V2X communications are considered.

To make predictions for continuous variables, the DBN outputs can be fed into the RNN, to complement the variable's history and add some context. However, was it not for the motion dynamics variables, the DBN would be sufficient to predict the states of any traffic participant, but because velocity is relative and depends on the states of the ego vehicle as well as the traffic participant, by some means, both of these value sets must be exposed to the RNN. To achieve that, an interface must have been designed for this connection. There are three general ways that this interface can be implemented. The trade-off between these configurations revolves around the difficulty to prepare the data for, training time and result reliability. Fig. 4 demonstrates simplified schematics of the architecture. In

¹“Prime” denotes a variable in next time slice

this study the leftmost configuration was selected where the data for the ego vehicle and traffic participants are processed with the same simple DBN topology in Fig. 3, due to shorter training times, easier pre-processing for the data and uncomplicated implementation.

In addition to the macro connection scheme, DBN variables need to be singled out and selected to meet the RNN, as brute-forcing everything into the RNN will cause jagged behaviour in the output. This jagged behaviour is because of the digital nature of the DBN outputs, especially when the distribution of states is not uniform across all the settings. The variables selected in this study are:

- *PathIntersection*
- *ApproachingIntersection*
- *AccelerationFlag*
- *TrafficLightSignalPrime*
- *SectionPrime*
- *LanePrime*

C. Recurrent Neural Network

In this study, the RNN is tasked with finding the velocity of a random traffic participant. While the DBN can quickly learn from very large datasets (in this study the DBN was trained with 480,000 sample points in a matter of seconds), the RNN takes much longer to train with the back-propagation algorithm, therefore only a small part of the data was chosen for training. This selection was based on data size and events (such as maximum and minimum velocities and maximum and minimum accelerations) as well as random entries. The network structure is conical with 4 layers in total, with the number of nodes computed in each layer using (5), where N is a constant for its subscript and n is a counter. The history window (d) is chosen to be 6 seconds and sampled at 1Hz. The activation functions are sigmoid.

$$N_{nodes \text{ in layer}} = (N_{ins} + N_{outs}) * (N_{hidden \text{ layers}} - n_{current \text{ layer}} + 1) \quad (5)$$

D. Kalman Smoothing

To minimize the jagged behaviour in the output from the digital nature of the DBN, a Kalman filter was introduced to smooth out the velocity predictions. Based on the basic discrete kinematic model (6), the state-space model (7) was created.

$$v(k + \Delta k) = v(k) + a(k)\Delta k \quad (6)$$

$$\begin{bmatrix} v(k + \Delta k) \\ a(k + \Delta k) \end{bmatrix} = \begin{bmatrix} 1 & \Delta k \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v(k) \\ a(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_v(k) \\ u_a(k) \end{bmatrix} \quad (7a)$$

$$Y(k) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v(k) \\ a(k) \end{bmatrix} \quad (7b)$$

To find $u_a(k)$ or the acceleration input, (8) was used, where Δk is the prediction horizon.

$$u_a(k) = \frac{v(\text{horizon} + k + \Delta k) - v(\text{horizon} + k)}{\Delta k} \quad (8)$$

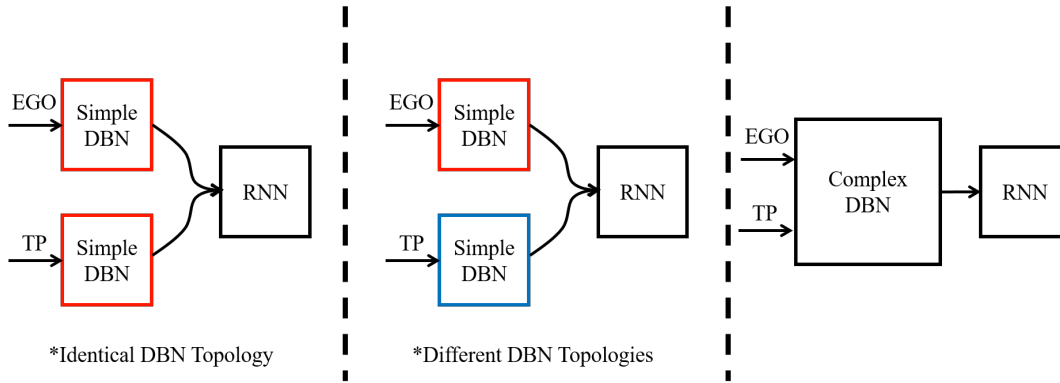


Fig. 4: DBN-RNN interface

Given that, the sensitivity of the Kalman filter to new inputs can be adjusted with the Q matrix. As the coefficient of the Q matrix increases, new inputs are trusted less and are not registered as valid right away. This gives Kalman filter the means to introduce inertia to the velocity changes which are physically in place but the neural network misses in the process. In this study the Q matrix coefficients were chosen as:

$$Q = \begin{cases} 0.02 & \text{for 1s prediction} \\ 0.10 & \text{for 2s prediction} \\ 0.60 & \text{for 5s prediction} \\ 4.00 & \text{for 10s prediction} \end{cases} \times I_2$$

IV. RESULTS

Velocity predictions for random vehicle pairs were established for 1, 2, 5, and 10-second horizons. A sample result is illustrated in Fig. 5 without and in Fig. 6 with Kalman filtering engaged. This specific drive-cycle is for a vehicle pair that coexisted in the simulation for a long period. This extended coincidence, alongside the multitude of full stops, signifies high traffic flow and multiple stops behind the traffic light. It is apparent that the prediction horizon has an inverse relationship with the prediction accuracy, and as the horizon extends, the errors become larger too. Fig. 5 also exhibits error spikes in larger horizons which are legacy of the digital DBN inputs, these spikes are recognized to be mitigated by the Kalman filter in Fig. 6. Results in Fig. 6 appear to be following the trend of speed variations very well, however at some points, especially in the sudden peaks in the actual speed profile, such as the peak at 300 seconds, the scaling seems to be not quite exact, which can be attributed to the nature of neural networks. Compared to the results previously declared in [10], for highway merging where the velocities in the drive cycles are more gradually changing and have fewer stops, these results do not exhibit the same level of accuracy, however, this does not point to a futile fate for this approach in this application either. It is critical to keep in mind that intersection driving is by far and undoubtedly more complex than highway merging, and besides that, there are multiple amends for this issue. For instance, replacing the RNN with a

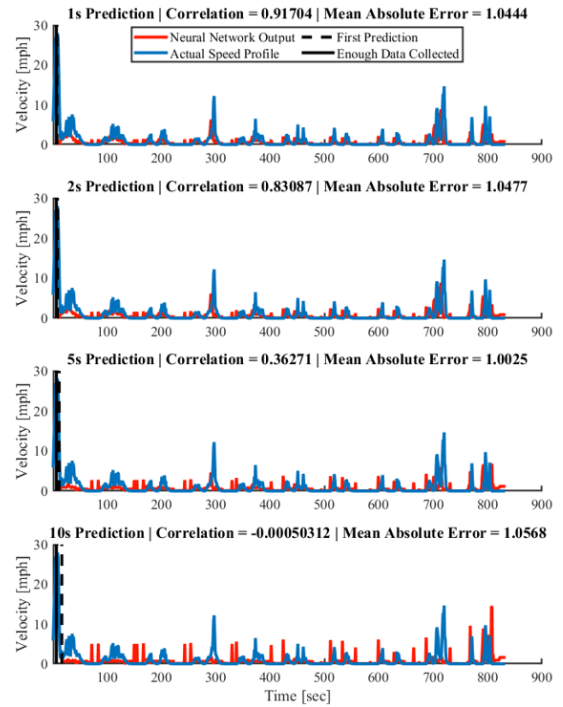


Fig. 5: Prediction results for random vehicle pair

more sophisticated method such as Long Short-Term Memory networks (LSTMs) could be a proper adjustment to this approach. Furthermore, Model Predictive Controllers (MPC), which are the prime target end-users for these predictions, tend to rely on 2-3 second prediction horizons which are adequately accurate.

V. CONCLUSION AND FUTURE WORK

In this endeavour, DBN was studied as a method of real-time microscopic traffic participant motion prediction in an urban intersection environment. It is understood that DBN is not sufficient for continuous variable processing, therefore a recurrent neural network was utilized to compensate for this shortcoming of the DBN. Moreover, a Kalman filter was used

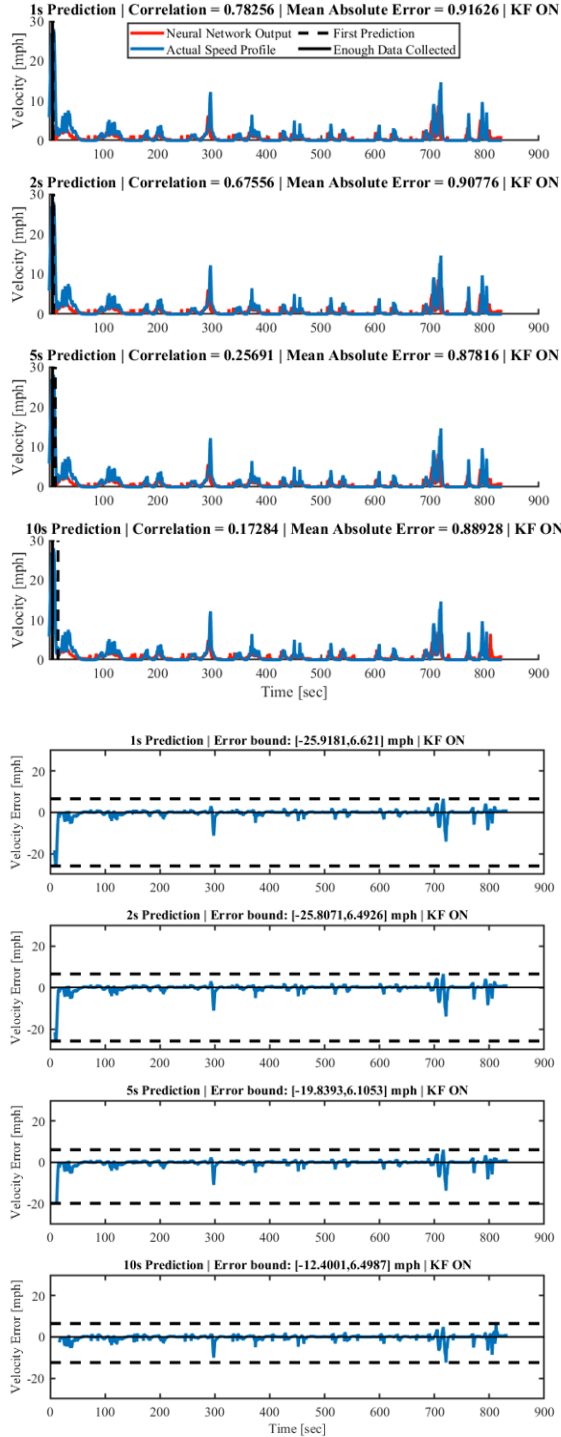


Fig. 6: Prediction results for random vehicle pair with Kalman Filtering

as a post-processing measure to ensure the velocity predictions do not suffer from radical and sudden peaks. It was argued that these predictions can help MPCs yield better results. The combination of the proposed prediction strategy with MPCs or other controllers that can make use of them should be an interesting investigation for the future. Moreover, other DBN network topologies and other methods for continuous variable prediction can be investigated and compared with the results of this study. Altogether, the envisioned future of transportation relies on the progress of methods and techniques such as this and the benefits of autonomous driving make it worthwhile.

ACKNOWLEDGMENT

The authors would like to thank Toyota, NSERC, and OCE for their generous support sponsoring this project.

REFERENCES

- [1] D. Watzenig and M. Horn, *Automated Driving Safer and More Efficient Future Driving*. Springer International Publishing, 2018.
- [2] *Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey*. Feb 2015.
- [3] *2015 Motor Vehicle Crashes: Overview*. NHTSAs National Center for Statistics and Analysis, 2016.
- [4] F. Zhang, J. Xi, and R. Langari, "Real-time energy management strategy based on velocity forecasts using v2v and v2i communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 2, p. 416430, 2017.
- [5] D. Moser, H. Waschl, R. Schmied, H. Efendic, and L. D. Re, "Short term prediction of a vehicles velocity trajectory using its," *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, vol. 8, no. 2, p. 364370, 2015.
- [6] W. Wang, J. Xi, and D. Zhao, "Driving style analysis using primitive driving patterns with bayesian nonparametric approaches," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, p. 29862998, 2019.
- [7] T. Gindele, S. Brechtel, and R. Dillmann, "Learning driver behavior models from traffic observations for decision making and planning," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, p. 6979, 2015.
- [8] A. Sarkar, K. Czarnecki, M. Angus, C. Li, and S. Waslander, "Trajectory prediction of traffic agents at urban intersections through learned interactions," *2017 IEEE 20th International Conference on Intelligent Systems (ITSC)*, 2017.
- [9] M. Zamani Abnili and N. L. Azad, "Short term predictions of preceding vehicle speeds for connected and automated vehicles," *Controls, Dynamic Systems and Robotics (CDSR)*, 2019.
- [10] M. Zamani Abnili and N. L. Azad, "Short-term traffic participants behavior prediction using a novel hybrid method in highway merging for automated vehicles," *IET Intelligent Transportation Systems (Under Review)*.
- [11] B. Sakhdari and N. L. Azad, "Adaptive tube-based nonlinear mpc for economic autonomous cruise control of plug-in hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, p. 1139011401, 2018.
- [12] H. Chen, L. Guo, H. Ding, Y. Li, and B. Gao, "Real-time predictive cruise control for eco-driving taking into account traffic constraints," *IEEE Transactions on Intelligent Transportation Systems*, p. 111, 2018.
- [13] S. Tajeddin, S. Ekhtiari, M. Faieghi, and N. L. Azad, "Ecological adaptive cruise control with optimal lane selection in connected vehicle environments," *IEEE Transactions on Intelligent Transportation Systems*, p. 112, 2019.
- [14] Z. Ghahramani, "Learning dynamic bayesian networks," in *International School on Neural Networks, Initiated by IIASS and EMFCSC*, pp. 168–197, Springer, 1997.
- [15] R. Raul, *Neural networks: a systematic introduction*. Springer, 1996.
- [16] 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 *The 21st IEEE International Conference on Intelligent Transportation Systems*, IEEE, 2018.