Bayesian Wireless Channel Prediction for Safety-Critical Connected Autonomous Vehicles

Mahmoud Elnaggar
University of Virginia
mae3tb@virginia.edu

Kamin Whitehouse
University of Virginia
whitehouse@virginia.edu

Cody H. Fleming
University of Virginia
fleming@virgina.edu

Abstract

We present a Bayesian deep learning approach to predict the quality of the wireless channel used by connected autonomous vehicles. The proposed approach captures the epistimic uncertainty of the wireless channel prediction model which is essential for decision making and motion planning algorithms in order to guarantee safe operation. We leverage a Bayesian Long Short-Term Memory (LSTM) network with Monte-Carlo (MC) dropout to estimate the posterior distribution of the packet delivery rate of non-stationary wireless channels. We train and validate the proposed approach on a dataset we generate in an indoor test-bed.

1 Introduction

The age of autonomous mobile systems has arrived. Self-driving cars have clocked over 10 million miles on public roads [6]. Enabling Wireless co-ordination between autonomous mobile vehicles can allow them reach high performance levels that would not otherwise be safe (e.g. achieving closer distances at higher speeds). A platoon of connected mobile vehicles or drones can move around blind corners at high speed by leveraging the sensing capabilities of the agents ahead of them [11].However, there is currently no methodology for providing safety guarantees for such high-speed states under *non-stationary* wireless channels. Wireless signals are transmitted through unguided media such as air or water and are therefore greatly affected by the surrounding environment. As mobile agents move through a physical environment, especially complex indoor environments but also outdoor and urban environments, changes in the environmental context produce *non-stationary* effects on the channel: future channel properties are not well predicted by those of the past. In this work, we focus on capturing the uncertainty in *non-stationary* wireless channel predictions which is essential to perform safe motion planning and control of connected autonomous vehicles. Traditional deep neural can quickly overfit the training data, especially LSTM networks [12], and/or provide over-confident predictions which may not be safe for safety-critical applications like autonomous driving. However, deep Bayesian neural networks have recently showed promise in capturing the epistimic uncertainty of their models which is important for safety-critical applications [3, 10, 2, 7, 9, 8]. We propose to use a Bayesian Recurrent Neural Network (RNN) that performs variational inference approximation via Monte Carlo (MC) dropout technique [4]. At each time step, the Bayesian RNN generates the predicted value of the packet delivery rate (PDR) of packets transmitted over the wireless channel from one vehicle to its connected neighbor over a future time horizon as well as the uncertainty of the predicted value.

1.1 Problem Formulation

Problem 1 *Given a platoon of N vehicles moving in a static environment. Assume that at each time $t = t_0$, each vehicle i has accurate estimates of its current state and the prediction of its future states over a time horizon $\boldsymbol{x}_i(t), \forall t \in [t_0, t_0 + T]$. Assume that every leading vehicle $i-1$ in the platoon sends its current and future state estimates periodically to the following vehicle i. The objective is,*

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

to estimate the posterior distribution of packet delivery rate over a future time horizon T, given the current observations $\hat{p}(PDR_i(t = t_0)|\mathcal{O}_i(t = t_0))$ at each follower vehicle i.

2 Methods

2.1 Bayesian LSTM Architecture

We propose to use a Bayesian Deep learning-based approach that comprises a Bayesian Long Short-Term Memory (LSTM) Network which estimates the posterior distribution of the output (PDR_i) over a finite time horizon T. The input to the Bayesian LSTM network is a sequence of length S_l the following observations $\mathcal{O}_i$ at each LSTM time step:

- The current position, orientation, the linear and angular velocity of the i^{th} vehicle $(x_i, y_i, \psi_i, v_i, \dot{\psi}_i)$.
- The current position, orientation and velocity of the leading vehicle $(x_{i-1}, y_{i-1}, \psi_{i_1}, v_{i-1}, \dot{\psi}_{i-1})$.
- The predicted future states of both vehicles over a time horizon T.
- The current measurements of the wireless channel (CSI) collected at the i^{th} vehicle.

The model architecture is a Bayesisn LSTM network. We approximate the variational inference by implementing MC dropout in LSTM network with the same network units dropped at each time step, randomly dropping inputs, outputs, and recurrent connections as described in [4].

The intuition behind using this approach is that the on-board sensors of autonomous vehicles are already being used to capture the static and dynamic features of the physical world, including the dynamic motion related to the vehicles themselves which have a great affect on the dynamics of the wireless channel. Our approach leverages these sensor data, the predicted motion paths of the vehicles, and the current state of the wireless channel to come up with not only the prediction of the future packet delivery rate of the wireless channel, but also with the uncertainty of that prediction. This approach can be generalized to include scenarios with moving objects in the environment by capturing and tracking these objects using the on-board sensors (e.g. LIDAR, Radar, Cameras) and input the tracked features to the Bayesian LSTM network to represent the current state of the physical world.

2.2 Dataset Generation

To the best of our knowledge, there is no publicly available data set that contains information about the motion states of connected vehicles as well as the variation of the wireless link quality between them as they move. In order to build our own dataset to train and validate our model, we use an indoor test-bed as shown in Figure 1. The test-bed is a 5m×4m×3m arena covered by a millimeter accurate motion capture system (OptiTrack [1]) and includes racing tracks for small sized robots/miniature

[1] `https://optitrack.com/products/prime-17w/`

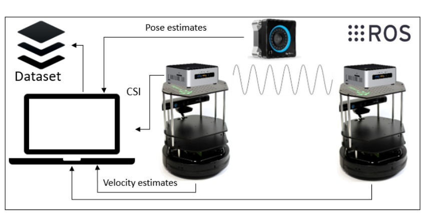


Figure 1: Data collection framework.

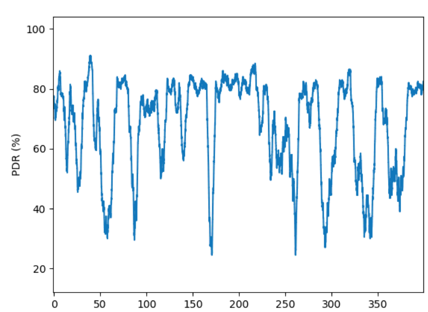
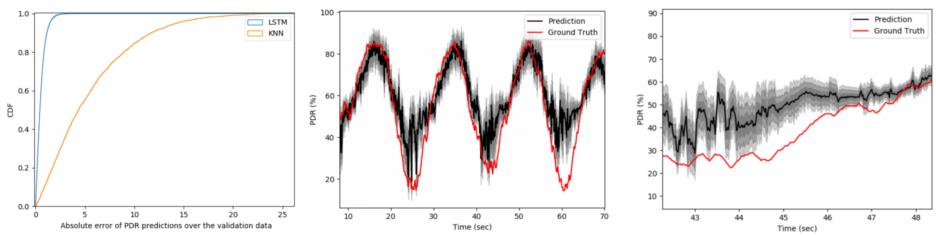


Figure 2: Example showing the significant variability in PDR due to a dynamic physical environment.



(a) (b) (c)

Figure 3: (a) The traditional LSTM network performs an order of magnitude better than KNN-based approaches. (b) A snapshot consecutive outputs of the Bayesian LSTM over time. PDR black line represents the mean of the mean and shaded grey zones represent 1,2 and 3 standard deviations of the estimated posterior.(c) A zoomed-in snapshot of the validation data showing the benefit of predicting uncertainty: the prediction uncertainty increases when the network's prediction is less accurate and vice-versa.

vehicles. We use two turtlebots, each equipped with an Intel NUC device that contains a commodity 5300 Wifi adapter used for communication between the two robots. We use the modified firmware [5] of the Wifi adapters in order to capture the Channel State Information (CSI) at the follower turtlebot. The OptiTrack system provides position and orientation information of the two turtlebots. We use the on-board wheel encoders on each turtlebot to provide linear and angular velocity information. All the data is synchronized and collected at a single computing device at the receiver. We use Robotic Operating System (ROS) framework to develop a data acquisition node that collects the data and build datasets to be used for training and validating the proposed Bayesian LSTM model. This method of dataset generation can be generalized for platoons of autonomous cars applications by using crowd sourcing techniques, recording features of static and moving objects extracted from each car's sensors and wireless adapters and uploading them to the cloud.

3 Experiments

We set up an environment in our test-bed arena that imposes fluctuations in the wireless channel quality values over time by removing the antennas from the Intel NUC devices and thus reducing the transmission power greatly relative to the area of the arena. Next, we collected data by moving one turtlebot in random motions around the arena with variable speeds and directions. We used a transmission frequency $f_t = 50Hz$. We recorded the PDR values over a time window $T = 4s$. Figure 2 shows that PDR values vary significantly between 20% and 90% as only one turtlebot moves randomly inside the arena. We used a Bayesian LSTM network with a sequence length $S_l = 20$, that contains one hidden layer of 128 LSTM cells. The dropout probabilities that generated best achieved performance were = 0.3 each. We used 50 Monte Carlo samples to generate the mean and variance of the estimated PDR posterior. We recorded a dataset of 24,000 data samples. We used 80% of the dataset for training and the rest for validation. First, we compared the performance of a traditional (non-Bayesian) LSTM network with a revised version of the KNN-based approach presented in [1] using the same features for prediction. Figure 3(a) shows that our trained LSTM network achieved a median prediction accuracy of +/-5% which is an order of magnitude better than performance of the KNN-based approach. Next, we trained the Bayesian LSTM network for 200 epochs. The network achieved an RMSE of 0.04 over training data and 0.17 over validation data. Figures 3 (b-c) show how the Bayesian LSTM outputs high uncertainty of less accurate PDR predictions and low uncertainty of more accurate ones.

4 Conclusion & Discussions

We presented a Bayesian deep learning approach to predict the quality of non-stationary wireless channels between moving vehicles along with uncertainty in prediction. Possible extensions of this work include training and testing the Bayesian LSTM model in increasingly complex environments, including outdoor environments which would require relying on more sensors in order to capture the dynamic environment (e.g. LIDAR, Radar, Cameras).

References

[1] R. C. Daniels and R. W. Heath. An online learning framework for link adaptation in wireless networks. In *Information Theory and Applications Workshop, 2009*, pages 138–140. IEEE, 2009.

[2] M. Fortunato, C. Blundell, and O. Vinyals. Bayesian recurrent neural networks. *arXiv preprint arXiv:1704.02798*, 2017.

[3] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[4] Y. Gal and Z. Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027, 2016.

[5] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. *ACM SIGCOMM Computer Communication Review*, 41(4):159–170, 2011.

[6] T. Higgins. Google's self-driving car program odometer reaches 2 million miles. Wall Street Journal, October 2016.

[7] A. Kendall, V. Badrinarayanan, and R. Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[8] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *2016 IEEE international conference on Robotics and Automation (ICRA)*, pages 4762–4769. IEEE, 2016.

[9] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017.

[10] N. Pawlowski, M. Rajchl, and B. Glocker. Implicit weight uncertainty in neural networks. *arXiv preprint arXiv:1711.01297*, 2017.

[11] V. Turri, B. Besselink, and K. H. Johansson. Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning. *IEEE Transactions on Control Systems Technology*, 25(1):12–28, 2017.

[12] W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.