

Lightweight LSTM for CAN Signal Decoding

Paul Ngo

ngopaul@berkeley.edu

University of California, Berkeley
Berkeley, CA, USA

Jonathan Sprinkle

sprinkjm@arizona.edu

University of Arizona
Tucson, Arizona, USA

Abstract

This paper describes an approach to identify undecoded Controller Area Network (CAN) data from one vehicle, based on the data similarity to previously decoded CAN data from another vehicle. Modern vehicles communicate data and signals from on-board sensors and controllers through the CAN bus. Networked sensors contain information such as wheel speeds, fuel gauges, turn signals, and radar signals. In the effort to use this information and make cars safer through human-in-the-loop CPS, signals on the CAN bus such as wheel speed and radar can be used to support the driver. However, data from the CAN bus are encoded and in some cases compressed, and different car manufacturers use different encoding schemes to represent data on the CAN bus. With hundreds of messages and thousands of possible encoding schemes to consider, it is laborious to identify the unique bits and encoding schemes that represent signals on each vehicle. In this study, we propose a method for training a Long Short-Term Memory (LSTM) neural network on known radar signals from one vehicle manufacturer, a Toyota, and successfully apply the network to identify the encoding for radar signals on a different vehicle, a Honda. By augmenting the training dataset with varied encoding bit boundaries, a small and lightweight LSTM network can learn to recognize radar data across different encoding schemes. The results are an improvement on exhaustive-search algorithms and other methods previously used in the search for such signals.

CCS Concepts: • Computer systems organization → External interfaces for robotics; • Computing methodologies → Learning paradigms; Neural networks.

ACM Reference Format:

Paul Ngo and Jonathan Sprinkle. 2021. Lightweight LSTM for CAN Signal Decoding. In *The Workshop on Data-Driven and Intelligent Cyber-Physical Systems (DICPS '21)*, May 18, 2021, Nashville, TN, USA. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3459609.3460528>



This work is licensed under a Creative Commons Attribution International 4.0 License.

DICPS '21, May 18, 2021, Nashville, TN, USA

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8445-2/21/05.

<https://doi.org/10.1145/3459609.3460528>

1 Introduction

Modern vehicles integrate a variety of sensors which communicate through an on-board network, the CAN bus. The vehicle's sensors, such as encoders, radar sensors, and proximity sensors, communicate across the CAN bus by broadcasting a message with a unique ID, called a message ID, for each type of message. A single message on a CAN bus, which is generally a 64-bit signal, can contain within it multiple sensor measurements—signals. However, the specific encoding scheme for these signals on the CAN bus, and the identifying message ID for each message, are design choices that vary from vehicle to vehicle. Signals within a message are delineated by consistent bit locations—bit boundaries—and are interpreted with a specific endianness, signed-ness, scaling, and offset (see Figure 1).

CAN data from many modern vehicles can be read from an easily accessible OBD-II port. However, currently, in order to interpret the unlabeled stream of data from the CAN bus, proprietary software from the vehicle's manufacturer must be used. Alternatively, a CAN labelling file, called a DBC file, can be manually developed by searching through thousands of possible encoding schemes. Some labeling schemes have already been developed for specific vehicles through brute-force methods. However, these methods are time-intensive and yield results specific to each vehicle model.

The work in this paper reduces the time spent searching for the exact encoding used for the radar signals on the CAN bus by finding highly likely bit boundaries where signals are located on a CAN bus message. We used labeled radar data from a Toyota RAV4 to train an LSTM neural network to distinguish between radar signals and non-radar signals and be robust to incorrect encoding assumptions, thus making the search for radar signals on non-labeled CAN data much faster.

2 Problem Statement

CAN data is organized into messages labeled with a message ID, timestamp, bus number, and data 8 bytes (64 bits) in length. Each individual signal is identified by a unique message ID, a set of bit boundaries for where the signal should be interpreted within the 64 bits of the message, and an encoding scheme for the signal. CAN data was obtained from both vehicles (the Toyota and Honda) by using a black panda kit developed by comma.ai, which allows for interfacing to the OBD-II port. **Libpanda** was used to interpret the signals from the black panda kit.

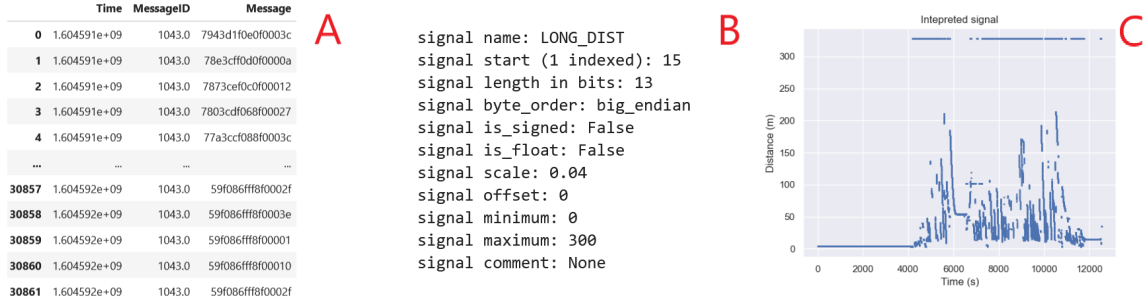


Figure 1. A long CAN message is interpreted with specific bit boundaries and an encoding, resulting in a signal output. A. A CAN message consists of a series of 64 bits which change each timestep. Bit boundaries delineate the start and end of a possible signal. **B.** A signal is interpreted using an endianness, signedness, scale, and offset. **C.** A final interpreted signal.

For the training data, we have a DBC file which labels the known bit boundaries and encoding scheme for radar data on the Toyota vehicle. This encoding scheme includes three main components: a) the endianness, b) the unsigned/signed nature of the signal, and c) the scale and offset of the signal.

We also have unlabeled CAN data from a Honda vehicle, which includes 106 different message IDs, with data 8 bytes in length. The goal is to find which message IDs contain radar data on the Honda vehicle, as well as possible bit boundaries within the messages which correspond to radar messages. There are likely multiple CAN message IDs which contain radar messages.

The search for radar signals is non-trivial. The hundreds of other messages on the Toyota RAV4 and Honda include many different types of signals. There are signals of fixed values of 0s or 1s, continuous signals (such as wheel speed and steering angle), indicator signals (such as for a turn signal or for a dashboard setting), counters (signal values which count upwards at a fixed rate), and checksums (signal values which verify the correctness of another signal). The challenge in the creation of a radar-finding algorithm is to find a signal which not only demonstrates continuous signal behavior—when the radar is tracking an object—but also indicator behavior—when a radar signal is unable to track an object, the signal often changes to a fixed value. Such an algorithm must be able to distinguish between a radar signal and other signals which may share some similar properties. Radar data is also highly discontinuous. While other signals on the CAN bus may be more easily recognized and classified with standard non-neural-network methods for signal classification, radar data does not track a continuous value. A radar signal may switch to track a different object suddenly—resulting in signal discontinuities—or lose line of sight with an object—resulting in a fixed unknown value.

3 Related Work

LSTMs, long-short-term-memory neural networks, are a type of neural network called a Recurrent Neural Network (RNN).

LSTMs demonstrate excellent performance at predicting time series data [1] and classifying time series data [2].

LSTMs have been demonstrated in the past to be effective at classifying ECG signals [3] and brain-wave data [4]. These signals are similar to CAN data in that an LSTM can learn to identify important features of the complex signal which cannot feasibly be determined manually. While a valid parameterization for CAN signal classification may be manually computed, the speed of using an LSTM can outperform classification by hand.

LSTMs also demonstrate excellent performance on classifying the modulation of signals, even with introduced noise [5]. Many vehicle signals have frequency-related or cyclical behavior based on the nature of the signal. In the context of our problem, the rate at which a radar signal loses track of an object which it is tracking or switches to track a different object can be roughly approximated with driving speeds.

Stacking multiple layers of RNNs, such as LSTMs, allow for the possibility of developing a latent space for higher order features of data, including different time scales and signal relationships in time [6]. Developing a model of higher order features may be useful for classifying signals even with incorrect bit boundary placements, which is demonstrated with our results in this paper.

LSTMs have been successfully applied to CAN bus data for a variety of applications. Qin et. al. applied an LSTM network to detect anomalies in CAN data [7], and Hossain et. al. used an LSTM network to detect malicious agents on a CAN bus [8]. Driver behavior and profiling was also demonstrated through the use of Convolutional Neural Networks (CNNs) working in combination with LSTMs [9, 10]. These examples demonstrate that LSTMs can be used to classify and describe the structure and nature of CAN bus signals.

Different CAN bus signals, such as wheel speed, have previously been found using LSTM networks. Huybrechts et. al. trained models to extract wheel speed data signals from the CAN bus and demonstrated that additional GPS data improved signal identification performance [11]. We

further develop this work by demonstrating that an LSTM can be applied to highly discontinuous signals such as radar signals, and that information about signal properties can be transferred across different vehicles.

Similar larger-scale neural machine learning methods have been applied to reverse engineer the encoding of signals on the CAN bus, such as the work from Verma et. al. on CAN-D [12]. Our work extends the possibilities of applying machine learning to signals which are highly discontinuous, unlike the identified signals described in Verma et. al.'s work which track continuous values.

4 Solution

A lightweight LSTM network was designed to distinguish between radar and non-radar signals, and be robust to incorrect bit boundaries. The network used two LSTM layers of 128 units each, followed by a fully connected ("dense") layer of 16 units (with linear activation), a dense layer of 128 units (with ReLU activation), and a dense layer of 2 units (softmax activation). Dropout with a rate of 0.5 was used between the second LSTM layer and the first dense layer.

An LSTM was chosen to distinguish radar signals because it performs well at classifying time series data. LSTMs improve upon standard RNNs with a modification to improve the ability for the network to learn long-term time dependencies between the input sequence. Figure 2 shows the general layout of an RNN. In the basic version of an RNN, the repeating modules' output and time series input is concatenated and passed into a single function, such as tanh. However, with an LSTM, the repeating modules are modified so that the network can selectively learn how strongly any repeating module's output should affect the next repeating module, or be passed along the network to the next layer.

The reason for using a relatively small network was to be able to efficiently run the network across multiple messages with several combinations of bit boundaries. Dropout was applied and a small layer size for the first dense layer was chosen to deter over-fitting. The LSTM was applied to identify radar signals in two steps: training and prediction.

For training, the labeled training data was specially modified in order to allow the LSTM layer to learn how to identify radar signals even if a bit boundary is incorrectly guessed during the prediction step.

Longitudinal radar data and non-radar signals were extracted from the Toyota CAN bus data using previously known bit boundaries that were developed as part of an open-source project [13] in an existing DBC file. These data were established as ground truth through several different methodologies. (1) The distance between fixed obstacles was recorded using measurement tape, and readouts were obtained from CAN data for these fixed obstacles while the vehicle moved; and (2) a coordinated drive was conducted between two vehicles, where velocity and GPS position of

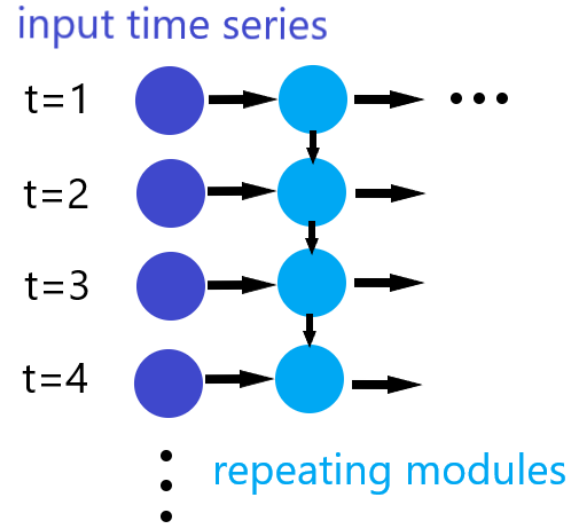


Figure 2. Depiction of the general layout of an RNN. Time series input is fed into several repeating modules. Each repeating module has output which contributes to the module associated with the next value in the time series.

each vehicle was recorded. We determined relative speed and velocity information from the CAN readout and compared these with data from the leading vehicle.

An additional perturbation to the data was performed to generate additional training samples. Additional signal data for both radar and non-radar signals were generated using randomized shifts to the bit boundaries around the signals. The randomized shifts, S , were drawn from a uniform distribution between 1 and 5 bits, inclusive. Seven (7) distinct perturbations were performed to generate additional training samples, resulting in an eight-fold increase in training data volume. The perturbations on the bit boundaries for each signal were:

- Shift both left and right bit boundary to the right by S
- Shift both left and right bit boundary to the left by S
- Shift right bit boundary to the right by S
- Shift right bit boundary to the left by S
- Shift left bit boundary to the right by S
- Shift left bit boundary to the left by S
- Shift left and right bit boundary to the left and to the right by S , respectively

These perturbations to the bit boundaries were added so that the neural network model would learn to identify radar signals even when a guessed bit boundary was incorrect. Therefore, not every bit boundary combination would need to be evaluated for a radar-like signal score during prediction. Close bit boundaries would still score highly, allowing rough bit boundaries to be found quickly.

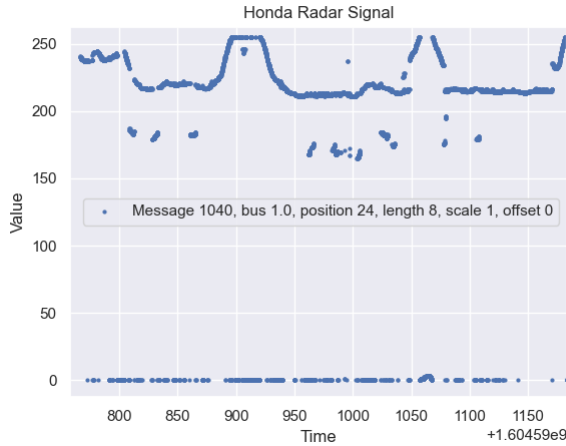


Figure 3. Graph of identified signal in Honda CAN data, based on bit boundaries predicted by the LSTM network.

All signal data were then re-sampled to 0.01s per time-step using the nearest point for the interpolation. This was important because some signals rely upon very specific changes to very high and very low values, and using a continuous interpolation, rather than a nearest-point interpolation, would modify the data signals unpredictably. The data were interpreted to be big-endian and unsigned. The data were individually scaled to be between 0 and 1, and then split into 1000 time-step lengths (100s) for the LSTM to train and test on. Finally, the LSTM was trained on shuffled data to predict the likelihood of two classes (radar and non-radar), using an 80:20 training and testing split, and achieved a test classification accuracy of 0.7125 over 9 epochs of training. Training took 35 minutes on an AMD Ryzen 5 2600 (no GPU).

For prediction, unlabeled signals were generated from each message ID from the Honda vehicle. We fixed bit boundaries to be of length 8 and iterated through all possible starting positions separated by 4 bits to generate possible signals. For the 106 message IDs, this generated 2552 signals. These signals were re-sampled to 0.01s, scaled to be between 0 and 1, and separated into 100s lengths, as before. The trained LSTM was used to generate predictions about whether each signal was a longitudinal radar signal. The output of the predictions were scores of each message ID and bit boundary combination. Prediction took about 4 hours on an AMD Ryzen 5 2600 (no GPU), running on a single core (without parallelization).

5 Results

Predictions from the model estimated the probability of each CAN message ID and bit boundary combination containing a radar signal. We expected that a group of similar message IDs would all be scored with a high probability. This is because

car manufacturers commonly identify signals from similar sensors to have message IDs close to one another. When tabulating the results of the LSTM predictions, this was the case: of the 30 signals which were rated to be radar signals with a probability of 0.9 or higher, 25 of the signals were between message IDs 1040 and 1090, inclusive. This suggests that the radar signals on the Honda vehicle are all grouped together on these message IDs, as expected. Time series plots of the signals with high radar probability indicates that the signals are very likely to be radar signals.

A longitudinal radar signal is characterized by a semi-continuous set of values, broken by intermittent lower values and max values. The intermittent values are usually clustered together around the same lower value, or equal to a fixed value, such as 0. Figure 3 shows one of the extracted signals from the algorithm which demonstrates these characteristics.

Predicted radar signals were verified by generating signals using bit boundaries within messages highly scored by the LSTM network. The behavior of the predicted radar signals were compared alongside time-synced dashcam footage recorded on the Honda vehicle. We know that a leading vehicle that the Honda is following will be tracked by radar on the Honda vehicle as a part of Honda’s Collision Mitigation system. Therefore, we compare the predicted radar signal value to the visual distance to a leading vehicle in the same lane as the Honda.

Figure 4 verifies that a signal found by the LSTM network indeed is a radar signal, by showing frames from a dashcam video alongside the predicted radar signal. We were able to verify the relationship by matching the timing of the flat sections of the radar signal to the dashcam video showing the Honda and leading car at stoplights. The discontinuous behavior of the radar resumes after the Honda loses track of the leading vehicle.

6 Future Work

Further work needs to be performed in order to verify components of the radar messages, which may include additional radar signals. The signals can be verified by running a controlled test by comparing the radar data to known values for driving the Honda vehicle alongside objects placed at known locations, or validating from existing coordinated drives between the Honda and Toyota vehicles. In these coordinated drives, the known radar signals from the Toyota vehicle can be used to validate the predicted radar signals on the Honda vehicle. Additional radar-related signals on the CAN bus of other vehicles such as the Toyota include confidence scores, relative velocity, etc., and we would like to identify and parse those signals in order to further inform the properties of human driving when extracting data from the CAN bus.

In the future, a similar network architecture can be used to transfer known signals from one vehicle to another, to

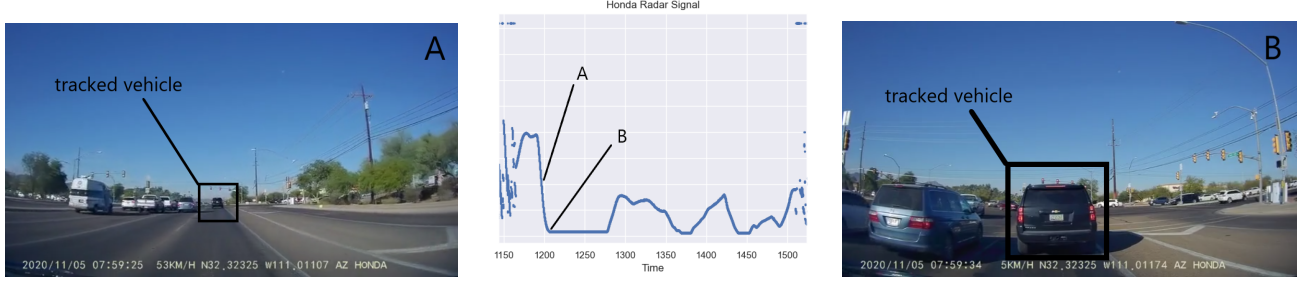


Figure 4. Verification of an identified radar signal. The Honda vehicle is following another leading vehicle in the same lane. At time **A**, the Honda decreases its distance to the lead vehicle as both come to a stop at a red light. At time **B**, the Honda comes to a stop behind the leading vehicle, matching with the flat part of the radar signal. The dashcam video matches with the radar signal behavior for a total of 346 seconds, at which point the Honda stops following the leading vehicle.

identify different CAN bus signals, especially highly discontinuous signals. Such a network architecture could be used in an auto-encoder to represent different signal types in a latent vector space and allow for efficient and accurate classification of CAN bus signals across multiple vehicle makes and models.

7 Conclusion

Labeling and decoding signals on a vehicle's CAN bus, even with many discontinuities, can be greatly assisted by training a small LSTM network on previously labelled data on a different vehicle. Such an LSTM network can be trained to be robust to bit boundary perturbation of signals through the manipulation of training set data, which allows for much faster identification of rough bit boundaries for unlabeled data. Furthermore, the LSTM architecture of a dropout connection from an LSTM layer to a low-neuron-count dense layer with linear activation proved successful in transferring generalized information about the structure of highly discontinuous radar data from one vehicle to another. Given the complex structure of a radar signal, it is likely that applications of a similar model to categorize a variety of different vehicle signals would be successful.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) award number CID DE-EE0008872. The views expressed herein do not necessarily represent the views of the U.S. Department of Energy or the United States.

References

- [1] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, "A comparison of arima and lstm in forecasting time series," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 1394–1401.
- [2] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "Lstm fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2018.
- [3] S. Saadatnejad, M. Oveisi, and M. Hashemi, "Lstm-based ecg classification for continuous monitoring on personal wearable devices," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 2, pp. 515–523, 2020.
- [4] S. Kumar, A. Sharma, and T. Tsunoda, "Brain wave classification using long short-term memory network based OPTICAL predictor," *Scientific Reports*, vol. 9, no. 1, p. 9153, Jun. 2019. [Online]. Available: <https://doi.org/10.1038/s41598-019-45605-1>
- [5] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433–445, 2018.
- [6] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," 2014.
- [7] H. Qin, M. Yan, and H. Ji, "Application of controller area network (can) bus anomaly detection based on time series prediction," *Vehicular Communications*, vol. 27, p. 100291, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2214209620300620>
- [8] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "Lstm-based intrusion detection system for in-vehicle can bus communications," *IEEE Access*, vol. 8, pp. 185 489–185 502, 2020.
- [9] A. Cura, H. Küçük, E. Ergen, and I. B. Öksüzöğlu, "Driver profiling using long short term memory (lstm) and convolutional neural network (cnn) methods," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2020.
- [10] J. Zhang, Z. Wu, F. Li, C. Xie, T. Ren, J. Chen, and L. Liu, "A deep learning framework for driving behavior identification on in-vehicle can-bus sensor data," *Sensors*, vol. 19, no. 6, 2019. [Online]. Available: <https://www.mdpi.com/1424-8220/19/6/1356>
- [11] T. Huybrechts, Y. Vanommeslaeghe, D. Blontrock, G. Van Barel, and P. Hellinckx, "Automatic reverse engineering of can bus data using machine learning techniques," in *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*, F. Xhafa, S. Caballé, and L. Barolli, Eds. Cham: Springer International Publishing, 2018, pp. 751–761.
- [12] M. E. Verma, R. A. Bridges, J. J. Sosnowski, S. C. Hollifield, and M. D. Iannacone, "CAN-D: A Modular Four-Step Pipeline for Comprehensively Decoding Controller Area Network Data," *arXiv e-prints*, p. arXiv:2006.05993, Jun. 2020.
- [13] Comma.ai, "opendbc: The project to democratize access to the decoder ring of your car," 2021. [Online]. Available: <https://github.com/commaai/opendbc>