Research Paper Presentation

Gautham Bellamkonda

Indian Institute of Technology, Hyderabad

May 10, 2021

Title and Authors

Title

Securing the Internet of Vehicles:
A Deep Learning based Classification Framework

Authors

- Tejasvi Alladi
- Varun Kohli
- Vinay Chamola, Senior Member, IEEE
- F. Richard YU, Fellow, IEEE

Glossary of terms

Internet of Vehicles (IoV) is a network of vehicles equipped with sensors, software, and the technologies that mediate between these with the aim of connecting and exchanging data over the Internet.
On Board Units (OBUs): Communication devices mounted on vehicles.
Road Side Units (RSUs): Communication devices mounted on along a road or a pedestrian passage way.
Deep Learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled.
Edge Computing is a distributed computing paradigm that brings computation and data storage closer to the location where it is needed to improve response times and save bandwidth.

Abstract

- Along with the various technological advancements, the next generation vehicular networks such as the Internet of Vehicles (IoV) also bring in various cybersecurity challenges.
- To effectively address these challenges, in addition to the existing authentication techniques, there is also a need for identification of the misbehaving entities in the network.
- This paper proposes a deep-learning based classification framework to identify potential misbehaving vehicles before the communication requests from OBUs can be entertained by the network infrastructure such as the RSUs.

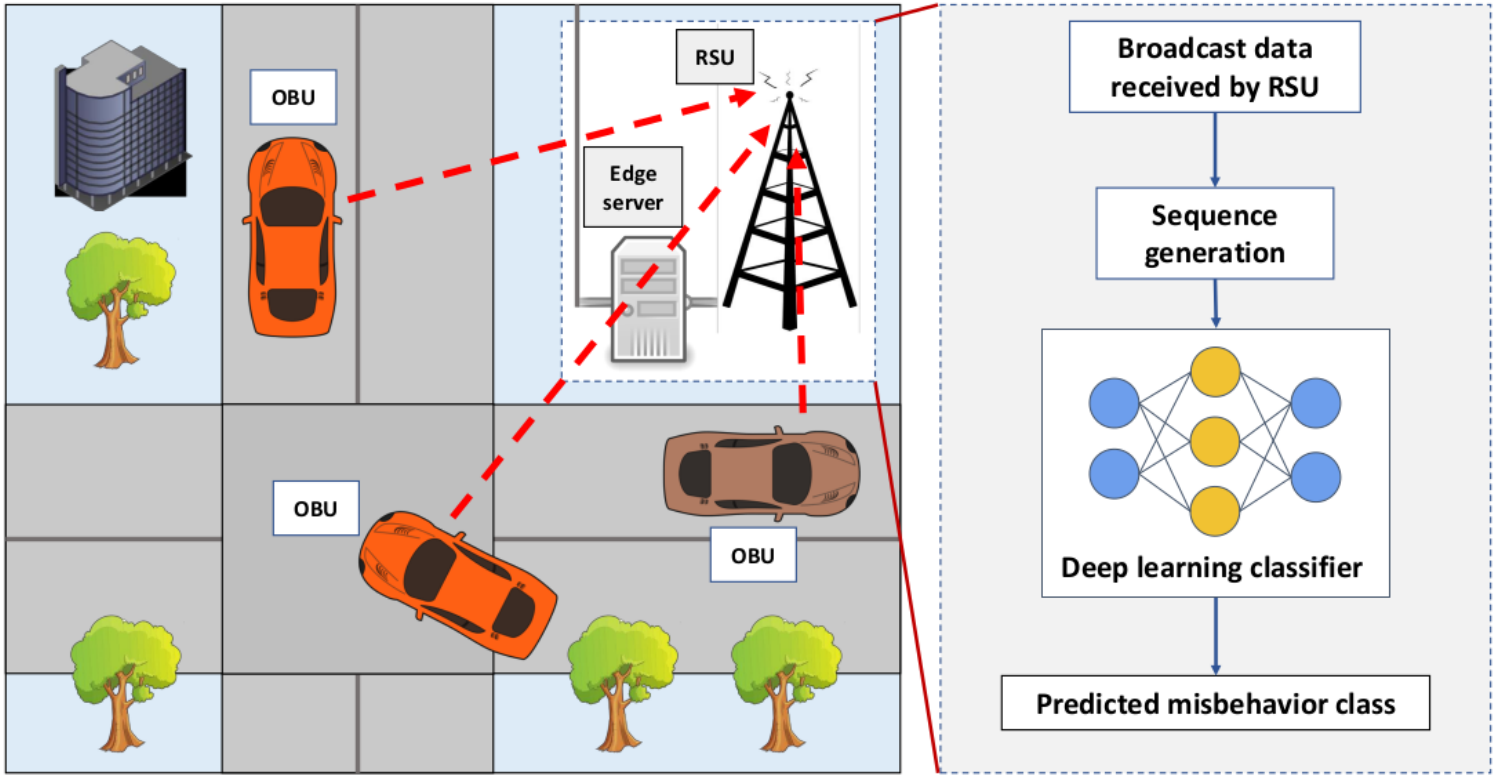
Why Deep Learning?

- Internet of Vehicles is expected to usher in an era of connected vehicles, which use their On Board Units (OBUs) to communicate with each other and with the road-side infrastructure called Road Side Units (RSUs).
- The growing number of communication links in IoV also increases the potential attack surfaces. Thus there is a greater need for developing security solutions for IoV networks.
- The classical intrusion detection approaches based on statistics or even machine learning may not be sufficient to address such high data applications. Hence, there is a need for deep learning-based classification approaches to identify potential attack scenarios.

Edge Computing

- Another aspect of consideration is regarding the deployment of the proposed classification approach. The architecture proposed by Loukas *et al*. is deployed on the cloud servers, however, cloud deployment has been shown to be both cost and computation-intensive.
- Instead edge computing as an alternative to cloud computing has been widely discussed. Even in vehicular networks, task offloading to edge computing has been shown to be quite promising.

Network Model



Network Model

- Considered IoV network model is a network of interconnected vehicles that communicate with each other (V2V communication) and with the RSUs (V2I communication) which are deployed at major road intersections.
- Deep learning classifiers are deployed on the edge servers that are co-located with these RSUs. The data broadcasted by the vehicles is received by the nearest RSU, which then passes it on to the edge server to check for possible intrusion in the network.
- The broadcasted messages received by the edge server from each vehicle is converted into time sequences and then passed through the deep learning classifiers to predict it as one of the classes defined.

Classification Approaches

Coarse grained classification method (CGCM)

This approach is a two-class coarse-grained classification method (CGCM) that distinguishes normal vehicle data from misbehavior data. In this approach, the faults and attacks are grouped into a single misbehavior (Faults + Attacks) class.

Fine grained classification method (FGCM)

The second ap- proach is a fine-grained classification method (FGCM) with three predicted classes based on normal vehicle behavior, faulty behavior, or attack behavior. Thus, compared to the first approach this is a more fine-grained classification approach.

Deep Learning Models

Two types of deep neural networks called Long Short-term Memory (LSTM) and Convolutional Neural Networks (CNN) are considered in this work.

Long Short-term Memory

LSTMs are found to be quite efficient in classifying time series data due to the feedback loops present in their architecture that can remember temporal data. Thus they find their applications in temporal sequence classification problems such as intrusion detection and speech recognition.

Convolutional Neural Networks

CNNs are a type of deep neural networks which work best on visual images, using an architecture of sliding filters and convolutional input layers.

Deep Learning Models

Two deep-learning architectures namely stacked LSTM and CNN-LSTM are used. Stacked LSTMs are created using multiple layers of LSTMs stacked one after the other, while CNN-LSTMs are intelligent combinations of CNN and LSTM layers. Four different models based on these archi- tectures are used here

- **Model 1** : Two 1D CNN layers with 1024 and 512 filters each followed by a max-pooling layer, an LSTM layer of 512 units and an output dense layer of 1 unit.
- **Model 2** : 3-LSTM model with 3 stacked LSTM layers
- **Model 3** : 4-LSTM model with 4 stacked LSTM layers
- **Model 4** : 5-LSTM model with 5 stacked LSTM layers

Each of the LSTM layers in the stacked LSTM models consists of 256 units.

Dataset

The popular VeReMi Extension dataset is used, as it covers several misbehaviour types including both faulty transmissions (faults) and cybersecurity attacks (attacks). There is data corresponding to eight such faults in this dataset namely constant position/velocity, constant position/velocity offset, random position/velocity, and random position/velocity offset. The dataset also consists of the following nine attacks namely Denial of Service (DoS), data replay, disruptive, DoS random, DoS disruptive, data replay sybil, traffic congestion sybil, DoS random sybil, and DoS disruptive sybil.
The data available in this dataset is in a rudimentary form, comprising of individual messages of every single simulated vehicle. The messages from each vehicle were used to create our dataset comprising of time sequences for each vehicle.

Dataset

Each of these generated sequences is 20x7 in size containing 20 messages per sequence and seven data fields, namely X, Y position coordinates of the vehicle, X, Y velocity coordinates of the vehicle, timestamp at which the message was broadcast, the pseudo-identity of the vehicle, and label for the vehicle's class type. The road map used in the simulation environment is plotted in Fig. 2 where the original X and Y position coordinates (in meters) are shown to be scaled down by a factor of 1000.

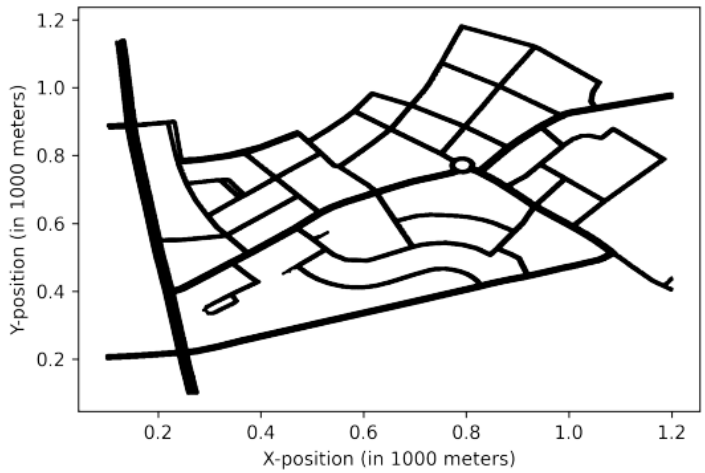


Figure: Roadmap

Dataset

1. Dataset for the CGCM classifier: has a total sample size of 85000 sequences: 42500 each for normal behavior and misbehavior, with 2500 sequences for each of the 17 misbehavior types con- sidered. The validation and test sets comprise 13600 sequences (6800 each for normal and misbehavior) and 6800 sequences (3400 each for normal and misbehavior) respectively.
2. Dataset for FGCM classifier: consists of 59998 input sequences, nearly 20000 for each class. There are 20000 normal behavior se- quences, 2500 sequences for each of the 8 fault types to obtain 20000 sequences of faults, and 2222 sequences each for the 9 attack types, making a total of 19998 attack sequences. This division is done to have an equal amount of data for each class. The validation set and test set are also created similarly with a total size of 9595 sequences (3200, 3200, 3195 sequences for normal, faults and attacks) and 4793 sequences (1600, 1600, 1593 sequences for normal, faults and attacks) each.

Training and Testing

Keras, a Python library for deep learning was used for the development of the models, and all the experimental training was carried out on the Google Colaboratory environment. We used Adam optimizer with a learning rate of 0.0003. The four models discussed (Model 1-4), were trained on mean absolute error (MAE) loss which is defined in the following equation where x is the input value, x_p is the predicted value and n is the total number of data-points over which the MAE is calculated. Testing was conducted on Intel i5 7 th Generation workstation using the Jupyter Notebook environment.

$$\mathcal{MAE} = \frac{1}{n}\sum_{i=1}^{n}|x - x_p| \tag{1}$$

Evaluation

We evaluate both the classifiers in terms of the four popular evaluation metrics after passing the test dataset through these classifiers. The four metrics accuracy, precision, recall, and F1-score denoted by $\mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{F}1$ respectively are defined below. Here $\mathcal{TP}, \mathcal{TN}, \mathcal{FP}, \mathcal{FN}$ refer to True Positive, True Negative, False Positive and False Negative respectively.

$$\mathcal{A} = \frac{\mathcal{TP} + \mathcal{TN}}{\mathcal{TP} + \mathcal{TN} + \mathcal{FP} + \mathcal{FN}} \tag{2}$$

$$\mathcal{P} = \frac{\mathcal{TP}}{\mathcal{TP} + \mathcal{FP}} \tag{3}$$

$$\mathcal{R} = \frac{\mathcal{TP}}{\mathcal{TP} + \mathcal{FN}} \tag{4}$$

$$F1 = \frac{2\mathcal{PR}}{\mathcal{P} + \mathcal{R}} \tag{5}$$

Evaluation

	Model 1				Model 2				Model 3				Model 4			
Class	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}1$	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}1$	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}1$	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}1$
Normal	0.933	0.893	0.984	0.936	0.95	0.926	0.977	0.951	0.973	0.951	0.998	0.974	0.965	0.936	0.999	0.966
Faults + Attacks	0.933	0.982	0.882	0.93	0.95	0.976	0.922	0.948	0.973	0.998	0.948	0.974	0.965	0.999	0.932	0.964
Average	0.933	0.9375	0.933	0.933	0.95	0.951	0.9495	0.9495	**0.973**	**0.9745**	**0.973**	**0.974**	0.965	0.9675	0.9655	0.965

TABLE I: Evaluation metrics for the course-grained classification method (CGCM)

	Model 1				Model 2				Model 3				Model 4			
Class	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}1$	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}1$	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}1$	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}1$
Normal	0.924	0.829	0.973	0.895	0.939	0.861	0.977	0.915	0.982	0.958	0.989	0.973	0.934	0.852	0.971	0.908
Faults	0.926	0.976	0.799	0.879	0.938	0.975	0.839	0.902	0.981	0.987	0.957	0.972	0.935	0.969	0.834	0.896
Attacks	0.992	0.985	0.992	0.988	0.991	0.984	0.989	0.987	0.994	0.992	0.991	0.991	0.994	0.991	0.991	0.991
Average	0.947	0.93	0.921	0.921	0.956	0.94	0.935	0.935	**0.986**	**0.979**	**0.979**	**0.979**	0.954	0.937	0.932	0.932

TABLE II: Evaluation metrics for the fine-grained classification method (FGCM)

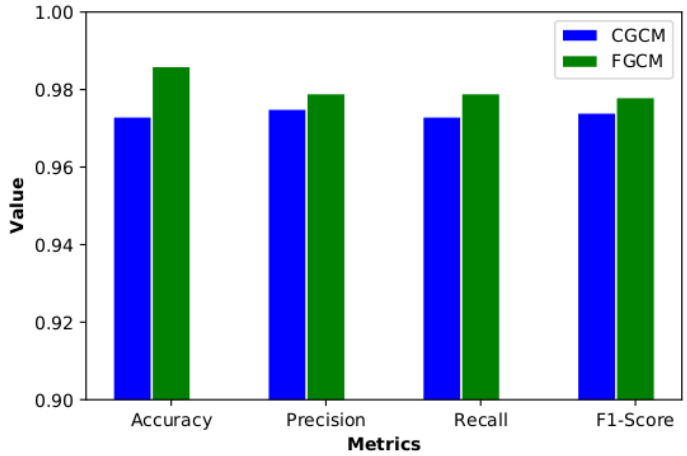


Figure: Comparison of the best average evaluation metrics for both the classification approaches

Conclusion

This letter proposed deep learning-based classification ap- proaches for identifying and classifying misbehaving vehicles in the IoV networks. Two classification approaches were con- sidered where one is a coarse-grained classification of normal vehicle data and all possible misbehavior types, the other is a more fine-grained classification to classify the misbehavior types into faults and attacks. Experimental results show that the fine-grained classification performs better across all the metrics and can be a better classification approach to different possible faults and attacks.

Thank You!