2-Step Prediction for Detecting Attacker in Vehicle to Vehicle Communication

1st Nur Cahyono Kushardianto
IEMN-DOAE, Polytechnic University of the Hauts de France (UPHF)
59313,Valenciennes, France
nurcahyono.kushardianto@etu.uphf.fr

2nd Yassin El Hillali
IEMN-DOAE, Polytechnic University of the Hauts de France (UPHF)
59313, Valenciennes, France
yassin.elhillali@uphf.fr

3rd Charles Tatkeu
COSYS-LEOST, Univ Gustave Eiffel, IFSTTAR
Univ Lille, F-59650 Villeneuve d'Ascq, France
charles.tatkeu@univ-eiffel.fr

2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall) | 978-1-6654-1368-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/VTC2021-FALL52928.2021.9625513

***Abstract*—Smart vehicles can be more adaptive to the road condition by exchange information between the vehicle. They can avoid traffic congestion, dangerous obstacles, even traffic accident earlier. This technology is closely related to the safety of the driver, therefore it must receive special attention. V2V communication has the potential to threaten interference and even attacks. There have been many studies that have focused on finding solutions to deal with these disorders. The first step is to strengthen the system's ability to detect attacks on V2V. On the other hand, the development of Machine Learning (ML) looks very promising to support this goal. In the proposed scheme, a 2-Step Prediction for detecting attackers is used. This system is using two classifiers ML from two modified training datasets. We show that the proposed scheme can improve the attack detection performance compared to one detection step.**

***Index Terms*—V2V, attack, detection, 2-Step**

I. Introduction

Intelligent Transportation System (ITS) are expected to provide safer travel, adaptive to road condition, less traffic congestion, and various entertainment service to the user. In order to make it happen, ITS system will exchange data between different ITS entities, roadside units, and traffic management. Absolutely when it's happened, security and privacy will be much important. A system that is closely related to human safety must receive special attention. Some cyber-attack can be a threat to ITS such as DoS attack which floods the network with bogus messages until the legitimate user cannot be connected to the system, fake users can falsify information about traffic condition, malicious users will steal personal data from legitimate users, and so on [1].

Cyber-attacks on *Vehicle to Vehicle* (V2V) are growing with more and more types of attacks. To detect and differentiate each type of attack is not easy and becomes a problem in itself. There are several methods of IDS such as support vector machine, decision tree, genetic algorithm, data mining, Artificial Neural Networks (ANN) and so on. Among the ANN is widely used for IDS but this method has a weakness in the training time [2]. Another method for IDS implementation is ML, Its method is widely used in the domain where it shows its perfection over classic rule-based algorithms. The ML method is being unified in cyber detection system in order to replace the first level of security analysts [3].

This work was supported by Ministry of Education and Culture of Republic of Indonesia, Université Polytechnique Hauts-de-France, Institut d'Electronique, de Microélectronique et de Nanotechnologie/Département Opto Acousto Electronique (IEMN/DOAE).

We created an attack detection scheme on V2V using the latest *Vehicular Reference Misbehavior (VeReMi)* Dataset [4]. In this scheme, we use a 2-Step Prediction technique to detect all types of attacks at one time. We apply this technique to DBN, LSTM, GRU, and Random Forest ML models as a comparison. In each model, the performance of the usual predictions will be compared in one step with the 2-Step Prediction that we propose. Apart from measuring the level of prediction/detection accuracy we also measure the detection speed for each simulation.

II. Related Works

In this section, we will provide a review of related works regarding misbehavior detection on V2V. The use of simulations based on ML is a distinctive feature of any related works.

Kamel et al [5] created Framework For Misbehavior Detection (F^2MD) for simulating real-time misbehavior detection in vehicular networks (VEINS extension). This framework is using ML model classifier such as SVM, MLP, and LSTM as misbehavior detector. They use also the first version of the VeReMI dataset which is basically comprise of 5 type of attacks [6]. Then in the next stage, F2MD is used to generate a VeReMi-extension dataset which is contain 19 types of attacks [4]. However Kamel et al did not focus on comparing ML models but focused on the functionality of the framework itself.

Gyawali et al. [7] in their research proposed a misbehavior detection system based on hybrid collaborative ML and reputation based misbehavior detection systems. From their simulation, the best results for precision, recall, and F1-score were obtained by Bagging and Random Forest. For position verification schemes, apart from using their dataset, they also use the first version of the VeReMi dataset [6] as a comparison. They get the result that for Constant Position, Random, Random Offset, and Eventual Stop types of attacks, their scheme outperforms the VeReMi dataset for 30% attacker density. They use the VEINS 4.7 Framework application.

Aloqaily et al. [8] proposed a hybrid attack/misbehavior detection system using Deep Belief Network (DBN) and Iter-

978-1-6654-1368-8/21/$31.00 ©2021 IEEE

Authorized licensed use limited to: National University Fast. Downloaded on May 27,2022 at 18:51:22 UTC from IEEE Xplore. Restrictions apply.

ative Dichotomiser 3 (ID3) Decision Tree (D2H-IDS). Their deep belief function is used to reduce data dimensions, while ID3 is used as a selection and attack classification feature. In this study, they compared their scheme with a normal DBN, DBN2-IDS [9] and DBN3-IDS [2]. They use the NS-3 application to generate normal vehicle traffic data and combine it with the NSL-KDD attack dataset at the preprocessing stage. The types of attacks contained in their dataset are DoS, Probe, Remote to User (R2L), and User to Root (U2R) which is simulate using MatLab application.

III. Methodology

In this paper, the main concern is improving the performance of misbehavior/attacks detection on V2V networks based on ML.

A. Challenge

Detection of many types of attacks at the same time is not feasible enough, the good result only achieves in training using original quantity data, of course, there is a high bias with the legitimate vehicle because it comprises a large amount of legitimate vehicle data. Tends to be easier if we only want to determine a vehicle is legitimate or an attacker. The challenge that arises is how to simultaneously detect all types of attacks in one simulation and at the same time improve its accuracy performance. Another challenge is determining the best ML model for attack detection and at the same time knowing the advantages and disadvantages of each model in terms of accuracy dan timing.

B. Contributions

In this section, we propose our technique to make a better performance in the detection different types of attack : *2-step prediction* is a prediction using two different classifiers. It begins with two different training with two different datasets of labels. The first dataset only has two labels, namely legitimate and attacker, training on this dataset will produce a classifier which we call "0-1 Classifier". The second dataset is a dataset that has 14 attack labels. Why only 14 attack labels will be explained in Chapter IV. In this dataset, legitimate vehicles are not included. Training for the second dataset will produce a classifier which we call "14 Attack Classifier". When a prediction is made of a detected vehicle data, the vehicle data will be predicted in advance by the 0-1 Classifier to determine whether the detected vehicle is a legitimate or attacker vehicle, if it is detected as legitimate then the vehicle is considered harmless and the prediction process stops. But if the vehicle is detected as an attacker, the vehicle data will be passed on the second predictor which uses "14 Attack Classifier" to determine the type of attacker. A 2-step prediction scheme is presented in Fig 1.

IV. Experimental Results

A. Dataset

1) Description: VeReMi-extension which from now on we just refer to as VeReMi is a dataset for the evaluation of misbehaviour detection mechanisms for Vehicular ad-hoc network (VANET). The initial dataset contains a number of simple and complex attacks [4]. List of Misbehavior/Attack in VeReMi : (1) Constant Position, (2) Constant Position Offset, (3) Constant Speed, (4) Constant Speed Offset, (5) Data Replay, (6) Data Replay Sybil, (7) Delayed Messages, (8) Disruptive, (9) Denial of Service (DoS), (10) Denial of Service Disruptive, (11) Denial of Service Disruptive Sybil, (12) Denial of Service Random, (13) Denial of Service Random Sybil, (14) Eventual Stop, (15) Grid Sybil, (16) Random Position, (17) Random Position Offset, (18) Random Speed, (19) Random Speed Offset.

VeReMi dataset consists of message logs per vehicle and originaly it has four main data fields: (1) Position, (2) Velocity, (3) Acceleration, and (4) Heading

2) Preprocessing: The dataset that will be used in the training process will go through a preprocessing step first. At this stage, only two types of data fields are selected: *Position* and *Velocity*, with the hope of shortening the detection time and making it more useful for future research. So we will get the dataset to consist of the vehicle's position and velocity progression. We will see an example of a vehicle data in the training dataset in Table I.

TABLE I: Representative of a Vehicle Data

21	**0**	**pos/0**	175.64	177.51	179.40	181.33	183.50
		pos/1	734.37	721.44	708.52	695.60	682.70
		spd/0	1.892	1.884	1.956	1.973	1.968
		spd/1	-12.930	-12.875	-12.863	-12.969	-12.939

Description of table I is as follows:

- **21**: vehicle sender ID
- **0**: type Vehicle, 0 is legitimate vehicle (non-attacker), 1 is an attacker vehicle.
- **pos**: position vehicle according to GPS coordinate x (pos/0) and y (pos/1)
- **spd**: speed/velocity vehicle according to velocity vector x (spd/0) and y (spd/1) in meter/second

In this example, the vehicle only has 5 (five) messages data, consist of 5 (five) pos/0, 5 (five) pos/1, 5 (five) spd/0 ,and 5 (five) spd/1.

For the purpose of the training process, the dataset will be provided in 6 different number messages for each vehicle : *5 messages* (5msg), *10 messages* (10msg), *15 messages* (15msg), *20 messages* (20msg), *25 messages* (25msg), and *30 messages* (30msg), with density of 30% attacker.

We will also examine the effect of increasing the number of messages per vehicle on the performance of each model.

In the Methodology section, 14 types of attacks are mentioned instead of 19 types of attacks as the original dataset. Through several training simulations using a dataset with 19 types of attacks, there is a tendency that several types of attacks have the same type. So that some attacks will be easier to detect if they are grouped into one small group. This can be seen from observations through the confusion matrix detection test results of each simulation. So the new list of Misbehavior/Attack that will be used in the next simulation

Authorized licensed use limited to: National University Fast. Downloaded on May 27,2022 at 18:51:22 UTC from IEEE Xplore. Restrictions apply.

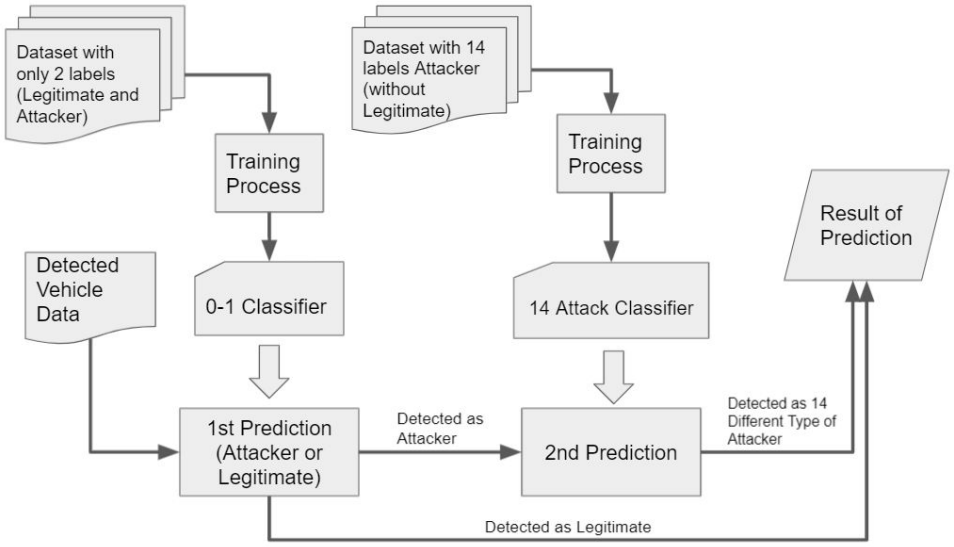

Fig. 1: 2-Step Prediction Scheme

process is as follows: (1) Constant Position, (2) Constant Position Offset, (3) *Constant Speed, Constant Speed Offset*, (4) *Data Replay, Data Replay Sybil*, (5) Delayed Messages, (6) *Disruptive, DoS Disruptive, DoS Disruptive Sybil*, (7) DoS, (8) *DoS Random, DoS Random Sybil*, (9) Eventual Stop, (10) Grid Sybil, (11) Random Position, (12) Random Position Offset, (13) Random Speed, (14) Random Speed Offset.

B. Model Machine Learning

In the *Related Works*, several ML models provide the best performance in detecting attacks or misbehaviors on ITS. We would like to compare the performance of these models when implemented using our proposed method. Therefore, we apply 2-step prediction to the following ML models:

- Deep Belief Network (DBN) [10]
- Long Short-Term Memory (LSTM) [11]
- Gated Recurrent Units (GRU) [12] [13]
- Random Forest (RF) [14]

Hyperparameters for each ML model are tuned using the HyperOpt-Sklearn application which is quite compatible with the Scikit-learn ML library for python. HyperOpt-Sklearn is an open-source library based on Bayesian optimization that can help determine the hyperparameters of a ML model [15].

C. Result

At this stage, the results of the 2-Step Prediction simulation will be compared with the usual simulation with 1 prediction stage which we call 1-Step Prediction. Simulations were carried out on four ML models DBN, LSTM, GRU, and RF using Python 3.8 in Anaconda environment. The amount of data used in the training process is balanced, meaning that the amount of data for each label is equal.

1) Prediction Accuracy: First we will see the simulation results of 1-Step Prediction in the table II. Average accuracy for 1-Step Prediction are DBN 0.56, LSTM 0.82, GRU 0.83, and RF 0.80. Standard deviation for each model are DBN 0.057, LSTM 0.03, GRU 0.029, and RF 0.027. It can be seen that increasing the number of messages/vehicles does not affect accuracy significantly, because the standard deviation value is still far below the average accuracy, especially for LSTM, GRU, and RF. However, for DBN the accuracy is slightly affected when compared to the other 3 models, but still not too large. The same thing is seen in the 2-Step Prediction in table III, where the average accuracy are DBN 0.75, LSTM 0.92, GRU 0.92, and RF 0.91. Standard deviation are DBN 0.057, LSTM 0.023, GRU 0.025, and RF 0.022.

TABLE II: Accuracy of 1-Step Prediction

	5msg	10msg	15msg	20msg	25msg	30msg
DBN	0.459	0.538	0.596	0.567	0.611	0.602
LSTM	0.785	0.796	0.833	0.826	0.857	0.860
GRU	0.803	0.801	0.831	0.859	0.863	**0.864**
RF	0.759	0.778	0.794	0.816	0.825	0.824

TABLE III: Accuracy of 2-Step Prediction

	5msg	10msg	15msg	20msg	25msg	30msg
DBN	0.643	0.770	0.773	0.736	0.799	0.793
LSTM	0.897	0.911	0.926	0.941	0.946	**0.959**
GRU	0.901	0.900	0.923	0.942	0.951	0.957
RF	0.887	0.890	0.913	0.924	0.939	0.934

All simulation results for each model have a significant increase in accuracy comparison of 1-Step and 2-Step Prediction. The best BDN accuracy in 1-Step increased by 18% in 2-Step see Fig 2. This shows a significant increase from 1-Step to 2-Step Prediction. Although compared to the other three models, the DBN accuracy is lower. The best result in this simulation is achieved by LSTM and GRU see Fig 3 and Fig 4. The increase in performance from 1-Step to 2-Steps is quite significant, with maximum results for LSTM and GRU reaching 95% accuracy. These two models obtain an accuracy value that is not much different, this is reasonable considering that basically, the GRU

Authorized licensed use limited to: National University Fast. Downloaded on May 27,2022 at 18:51:22 UTC from IEEE Xplore. Restrictions apply.

algorithm is almost the same as LSTM, only different in the type of gate and memory. RF also has a significant increase in accuracy with a value that is close to the accuracy of LSTM and GRU see Fig 5.

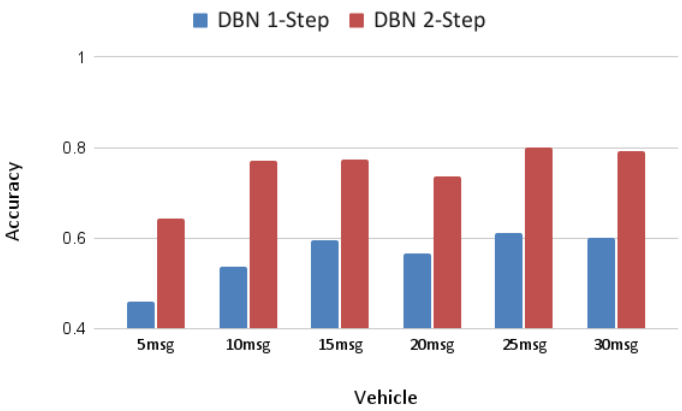

Fig. 2: DBN Comparison 1 and 2 Step Prediction

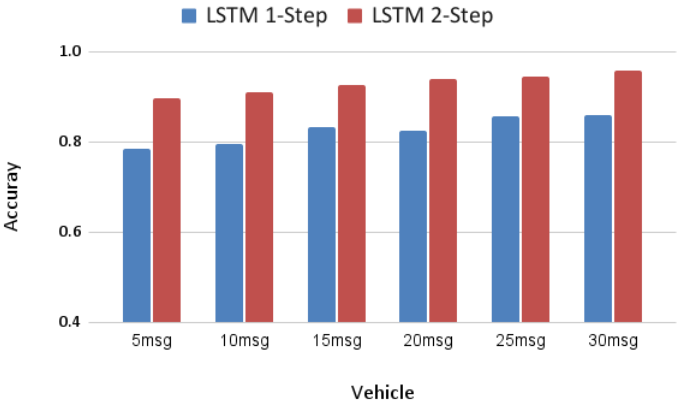

Fig. 3: LSTM Comparison 1 and 2 Step Prediction

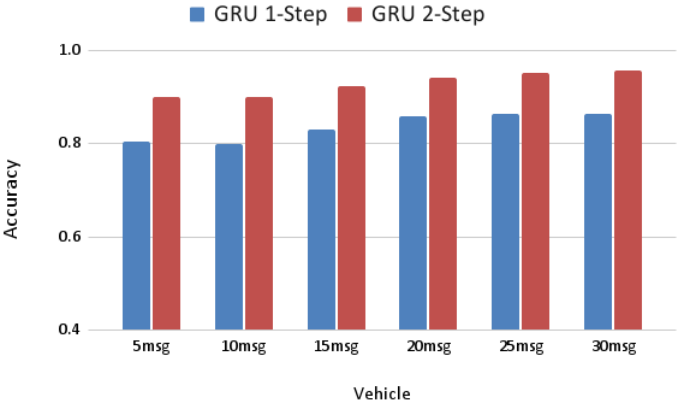

Fig. 4: GRU Comparison 1 and 2 Step Prediction

2) Prediction Timing: Apart from comparing the accuracy, it is also necessary to compare the prediction timing between 1-Step and 2-Step Prediction. This simulation is run on a workstation with a 3 GHz Processor, 16 Gbytes RAM, and

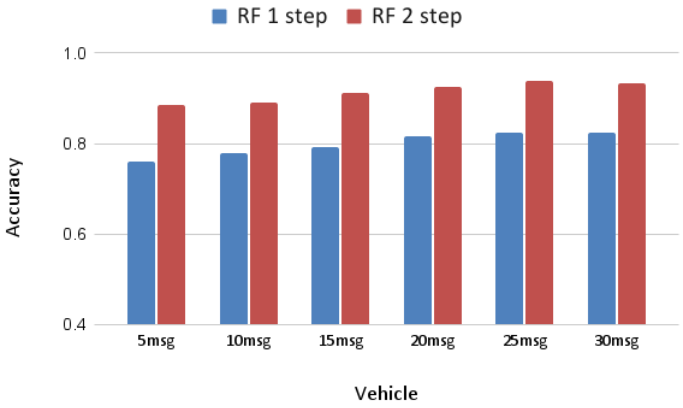

Fig. 5: RF Comparison 1 and 2 Step Prediction

2 Gbytes VGA memory. The dataset to be used in timing comparison is 20% of the test dataset of each original dataset from the type of dataset 5 messages to 30 messages. Because it uses original data, the number of vehicle for each type of message is not exactly the same. The timing for each ML model will be compared during 1-Step Prediction and 2-Step Prediction. In general, more time is needed to do 2-Step Prediction. This is normal because the data must pass 2 classifiers before the data type can be predicted.

LSTM has the slowest prediction time for 2-Step Prediction, especially for a dataset of 25 messages and 30 messages, where the average time required to predict the type of one vehicle is 1.05 millisecond and 1.1 millisecond see table IV. GRU is also one of the slowest models to perform 2-Step Prediction. For the 25 messages and 30 messages dataset type, the prediction time for one vehicle is about 0.6 millisecond and 0.8 millisecond. Event though it is slightly better than LSTM see table V. It seems that the use of 2 gates on the GRU is quite influential on the processing speed compared to 3 gates on the LSTM. Whereas the best prediction speed is owned by RF see table VI, with an average prediction time of 0.05 millisecond to predict the type of one vehicle. DBN has better prediction speed for 2-Step Prediction than LSTM and GRU, although it is not as good as Random Forest, i.e. the average prediction time is 0.1 millisecond. In DBN the difference in prediction speed between 1-Step and 2-Step is quite different. There is a difference of about 0.1 milliseconds between 1-Step and 2-Step Prediction, see table VII. Even so, DBN timing value is still better than LSTM and GRU for all messages per vehicles.

TABLE IV: LSTM Timing Prediction Comparison

	Average Timing per Vehicle (ms)	
	1-Step Prediction	**2-Step Prediction**
5msg	0.187037	0.572246
10msg	0.26089	0.644283
15msg	0.263169	0.726482
20msg	0.331702	0.85803
25msg	0.377783	1.059195
30msg	0.51523	1.162597

Authorized licensed use limited to: National University Fast. Downloaded on May 27,2022 at 18:51:22 UTC from IEEE Xplore. Restrictions apply.

TABLE V: GRU Timing Prediction Comparison

	Average Timing per Vehicle (ms)	
	1-Step Prediction	**2-Step Prediction**
5msg	0.181037	0.567611
10msg	0.178704	0.544212
15msg	0.229926	0.687511
20msg	0.252248	0.671141
25msg	0.318604	0.838061
30msg	0.31948	0.837791

TABLE VI: RF Timing Prediction Comparison

	Average Timing per Vehicle (ms)	
	1-Step Prediction	**2-Step Prediction**
5msg	0.031919	0.050371
10msg	0.032266	0.053524
15msg	0.03259	0.052473
20msg	0.034058	0.053812
25msg	0.03423	0.054856
30msg	0.034722	0.057679

TABLE VII: DBN Timing Prediction Comparison

	Average Timing per Vehicle (ms)	
	1-Step Prediction	**2-Step Prediction**
5msg	0.017059	0.106221
10msg	0.017088	0.106576
15msg	0.018409	0.100628
20msg	0.023695	0.122993
25msg	0.021548	0.111955
30msg	0.023292	0.117797

V. Discussion

LSTM and GRU have high accuracy in detecting attacks on V2V, but the time required for detection tends to be slow. Meanwhile, DBN has a fairly fast detection time but has a lower detection accuracy. One thing that is promising is obtained from Random Forest which has a fairly high level of accuracy and faster detection times than others. In this case, there is an opportunity to develop DBN and Random Forest so that they can have even higher accuracy without reducing performance in detection timing speed. Other than that, Observations of the model performance can still be developed by taking into account the precision, recall, and F1-score factors.

VI. Conclusion and Future Works

In this paper, we have presented techniques for attack detection on communication between vehicles at ITS. We get some conclusions: The 2-Step Prediction technique is able to improve the accuracy performance of each model ML even better because this technique focuses on how to classify the type of attack after separating the attacker's vehicle from the legitimate vehicle first. The best accuracy for training results and predictions using clustering and 2-Step prediction is GRU and LSTM, while the lowest accuracy is DBN. The slowest 2-step Prediction speed is LSTM, while the best speed for this technique is Random Forest.

The increase in the number of messages per vehicle does not have much effect on the detection speed of the model. Even though the Accuracy of DBN is slightly affected by the increase in the number of messages per vehicle compared to the other 3 models. The LSTM detection speed is slightly affected by the increase in the number of messages per vehicle compared to the other 3 models. Doing some clustering on the VeReMi dataset which consists of 19 types of attacks into 14 types of attacks is sufficient to increase the predictive accuracy performance of each ML model.

For the next step, attack detection on ITS will be developed in real time using existing ML models or models that will be further developed.

Acknowledgment

This research was supported by Ministry of Education and Culture of Republic of Indonesia. We would also like to thank our colleagues from Institut d'Electronique, de Microelectronique et de Nanotechnology (IEMN) of Université Polytechnique Hauts-de-France (UPHF).

References

[1] M. A. Javed, E. B. Hamida, A. Al-Fuqaha, and B. Bhargava, "Adaptive security for intelligent transport system applications," vol. 10, no. 2, pp. 110–120.

[2] G. Zhao, C. Zhang, and L. Zheng, "Intrusion detection using deep belief network and probabilistic neural network," in *22017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 639–642, IEEE.

[3] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, "On the effectiveness of machine and deep learning for cyber security," in *2018 10th International Conference on Cyber Conflict (CyCon)*, pp. 371–390, IEEE.

[4] J. Kamel, M. Wolf, R. W. van der Hei, A. Kaiser, P. Urien, and F. Kargl, "VeReMi extension: A dataset for comparable evaluation of misbehavior detection in VANETs," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE.

[5] J. Kamel, M. R. Ansari, J. Petit, A. Kaiser, I. B. Jemaa, and P. Urien, "Simulation framework for misbehavior detection in vehicular networks," vol. 69, no. 6, pp. 6631–6643.

[6] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "VeReMi: A dataset for comparable evaluation of misbehavior detection in VANETs,"

[7] S. Gyawali and Y. Qian, "Misbehavior detection using machine learning in vehicular communication networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE.

[8] M. Aloqaily, S. Otoum, I. A. Ridhawi, and Y. Jararweh, "An intrusion detection system for connected vehicles in smart cities," vol. 90, p. 101842.

[9] B. Wang, S. Sun, and S. Zhang, "Research on feature selection method of intrusion detection based on deep belief network," in *Proceedings of the 2015 3rd International Conference on Machinery, Materials and Information Technology Applications*, pp. 556–561, Atlantis Press, 2015/11.

[10] albertbup, "A python implementation of deep belief networks built upon numpy and tensorflow with scikit-learn compatibility," 2017. Available at https://github.com/albertbup/deep-belief-network [Online].

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, p. 1735–1780, Nov. 1997.

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling,"

[13] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation,"

[14] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, pp. 278–282 vol.1, 1995.

[15] B. Komer, J. Bergstra, and C. Eliasmith, "Hyperopt-sklearn: Automatic hyperparameter configuration for scikit-learn," pp. 32–37.

Authorized licensed use limited to: National University Fast. Downloaded on May 27,2022 at 18:51:22 UTC from IEEE Xplore. Restrictions apply.