Anomaly Detection for Connected Autonomous Vehicles Using LSTM and Gaussian Naïve Bayes

Pegah Mansourian[1(✉)], Ning Zhang[1], Arunita Jaekel[1], Mina Zamanirafe[1], and Marc Kneppers[2]

[1] University of Windsor, Windsor, ON, Canada
mansourp@uwindsor.ca
[2] Telus Communications Inc., Vancouver, Canada

Abstract. In the foreseen future, connected autonomous vehicles (CAVs) are expected to improve driving safety and experience considerably; however, cybersecurity remains a critical issue. CAN protocol, the de-facto standard for in-vehicle networks, provides no security mechanism, which makes it one of the most attack-prone parts. The lack of security mechanisms in CAN messages allows intruders to conduct devastating attacks, putting drivers' and passengers' lives at risk. An Intrusion Detection System (IDS) can monitor CAN network activities and detect suspicious behaviors resulting from an attack to help safeguard CAVs. The destructive behavior of an intruder is reflected as point and group anomalies in the sequence of CAN messages. Our study proposes an LSTM-based IDS for the CAN bus by exploiting the temporal correlations of the messages on the bus to detect anomalies. Specifically, it is a one-class classifier trained with attack-free data to predict the upcoming value of CAN messages. Then a Gaussian Naïve Bayes classifier is used to classify messages as normal and attack according to the resulting prediction errors. The proposed IDS is evaluated in terms of detection performance and compared with state-of-the-art one-class classifiers, including OCSVM, Isolation Forest, and Autoencoder, using two real-world datasets (Car Hacking Dataset and Survival Analysis Dataset). The proposed method outperforms baselines and achieves detection accuracy and F-score by nearly 100%.

Keywords: In-vehicle security · CAN · Anomaly detection · IDS · LSTM

1 Introduction

Connected autonomous vehicles (CAVs) incorporating a variety of sensors and onboard computing, expect to significantly improve road safety and efficiency of the transportation system. A CAV can have over 100 Electronics Control Units (ECUs), which help achieve a certain level of automation and gain essential information for decision makings. ECUs can handle different tasks, ranging from simple operations such as opening a window to more critical complex ones that

J. Zhao (Ed.): WiSATS 2023, LNICST 509, pp. 31–43, 2023.
https://doi.org/10.1007/978-3-031-34851-8_3

need communication between several ECUs, such as line detection and adaptive cruise control. Therefore, for a vehicle to work properly, an in-vehicle communication network is required to exchange messages of the ECUs with each other and the gateway to the out-of-vehicle world [15]. There are several protocols designed for the in-vehicle network, including Local Interconnect Network(LIN), Media-Oriented System Transport(MOST), FlexRay, and Controller Area Network (CAN). MOST is an expensive and high-bandwidth protocol, mainly used for the infotainment system, while LIN is a low-bandwidth protocol used for non-safety critical sensors and actuators. However, CAN is the dominant in-vehicle protocol due to its simplicity and reliability.

While the introduction of CAV brings comfort and safety features, the addition of components like ECUs, their distributed internal communications, and external network access generate new attack surfaces and raise security issues. An attacker can gain access to the in-vehicle network either by exploiting an external wireless communication, for instance, Wi-Fi and Bluetooth, or by physically tapping into the CAN bus via OBD-II Port and causing critical damage to the vehicle, driver, and passengers [8]. For CAN bus, encryption, authentication, and other security measures are not supported. Due to the broadcast nature of the CAN bus, lack of authentication (sender ID) and encryption, and weak access control, CAN protocol is prone to many attacks, e.g., DoS, masquerade, suspension, replay, fabrication, and remote access attacks. In an experiment, Koscher et al. connected to the OBD-II port of a car and sniffed the packets on the CAN bus with the CARSHARK component. Then by targeted probing, fuzzy, and reverse engineering techniques, they could identify the corresponding messages of each ECU. They successfully overrode Body Control Module (BCM) messages of door locks, headlights, and wipers and displayed the manipulated speedometer readings on the Driver Information Center [8].

As there are no security mechanisms in the CAN protocol, the next defense layer is to deploy an intrusion detection system (IDS) in accordance with the defense-in-depth strategy. IDS is a system to monitor the events and messages in the network to detect malicious events and violations. Depending on the detection approach, IDS classifies into signature-based, anomaly-based, and hybrid IDS. In a signature-based IDS, every network traffic is checked against a database that contains signatures of known attacks to detect a match or intrusion. Signature-based IDS are incapable of detecting unknown types of attacks or zero-day attacks. Anomaly-based IDS, on the other hand, can accomplish this goal by obtaining the expected profile of the network and detecting deviations from it as intrusions. It is more common these days because of the ability to work well with rarely available attack data and independently of human knowledge. Some anomaly-based IDS use the statistical [12,16] or information theory [13] features of CAN messages to identify anomalies, while others offer machine learning methods like SVM [1,2,18], HMM [9,14], SOM [3], GBDT [19], and NN [4,6,7,10,11,17,20].

In this paper, we propose an anomaly-based IDS for the CAN bus network by using machine learning. Attacks on the CAN bus are group (conditional) anomalies and only can be identified when examining them in a batch of data.

Hence in the proposed IDS, an LSTM module is designed to extract the normal model of the CAN network, considering the sequence of messages in time. It uses the latest historical CAN messages as input and predicts the current expected message based on them. Conditional anomalies in CAN messages can be reduced to point anomalies considering LSTM prediction errors, which can then be classified using conventional machine learning techniques. Unless an attack is conducted, the predicted value accords with the given value. When it comes to the attack case, the prediction errors raise the alarm about suspicious activity. In our experiments, we have found that the prediction errors produced by the attack and normal classes data follow Normal or Gaussian distributions with different mean and standard deviations. Among classifiers, the Gaussian Naïve Bayes (GNB) classifier focuses on the difference between Gaussian distributions of features across classes. Therefore, we have combined the LSTM module with a GNB classifier to find point anomalies in the prediction errors.

In the rest of the paper, first, we review the previous studies on CAN bus anomaly detection-based IDS in Sect. 2 and categorize them according to their methodologies. Section 3 describes the overall structure of the proposed LSTM-GNB method and each building block in detail. In Sect. 4, two datasets are used to conduct experiments, and the performance evaluation results and comparisons with the state-of-the-art baselines are provided. Finally, the conclusion is outlined in Sect. 5.

2 Related Works

The security of In-vehicle networks, especially the CAN bus, is one of the challenging research areas. Different approaches are proposed by researchers to detect attacks on the CAN bus. The detection methods can be categorized into statistical-based, information theory-based, and machine-learning-based methods.

In the entropy-based anomaly detection proposed by Muter et al., it is assumed that the entropy in the normal activity of the network is low. Once an attack happens it changes the frequency and content of messages in a way that the entropy would increase in the network. They calculated the entropy at each time step t and compared it to the normal situation to find the attacks. They evaluated their method on a real vehicle connecting its CAN-Body to a laptop and logging the messages with CANoe software. In experiments, replay, flooding, and spoofing attack messages were injected, and their entropy-based method almost successfully detected them. However, it has a limitation on finding spoofed messages that adhere to the normal one [13].

Since in-vehicle networks are more restricted and deterministic than other types of networks, Taylor et al. proposed a statistical-based method that makes use of the frequency and timing feature of CAN messages in a moving window. Then it analyzes them with statistical formulas to extract the normal profile of the network and detect messages that violate the normal profile as anomalies. They also used a 2011 Ford Explorer to generate CAN messages and injected

replay data into it with different rates and durations. The detection results were compared with OCSVM considering the Area Under Curve (AUC) metric. Despite the finding that AUC improves with increases in duration and rate, low-frequency and short-duration attacks might not be detected. [16].

The work done by Marchetti et al. is also a statistical anomaly detection. In this research, the authors model the normal behavior of the CAN bus by keeping track of all possible CAN ID sequences in a data structure called the transition matrix. In the detection phase, if an observed sequence of IDs does not match the transition matrix, it is labeled as an anomaly. Their method worked well with spoofed messages, reaching 100% detection, but very poor on replay messages since they were already seen in the attack-free data used in the training phase [12].

Although statistical and entropy-based approaches are very light-weighted and resource-friendly, they might fail to detect unseen (zero-day) attacks, especially when occurring within the content of messages. To overcome this challenge, machine learning has become the dominant approach to solving problems in unknown environments, where prior expert knowledge is not required to define patterns and features of the system. It has been widely used in recent studies done in the CAN bus anomaly detection area. Among machine learning-based methods, some of them are designed with conventional ML methods. The most popular conventional algorithms are SVM [1,2,18], HMM [9,14], SOM [3], and GBDT [19].

The neural network has shown good performance in recent studies on CAN bus anomaly detection. Kang et al. proposed a DNN structure to detect attacks in the CAN bus. To prevent the vanishing gradient problem in DNN, Restricted Boltzmann Machine is used to pre-train the model's weights. They evaluated their method on a simulated dataset generated by OCTANE and reached a 99.9 percent detection rate with seven hidden layers deep network [7].

A denoising autoencoder with a deep neural network structure is proposed by Lin et al. The model is trained with normal data; hence the anomalies cannot be reconstructed perfectly, and the error would raise an anomaly flag. The other novelty of their work is that they have used an evolutionary optimization algorithm to determine the best structure of the network in terms of the number and size of hidden layers. Three datasets were used to test the performance of this model - two generated and one publicly available OTIDS dataset. It is compared to ANN, K-means, and Decision Tree in terms of precision, recall, and F-score and achieved around 0.98 F-score, which is higher than the baselines [10].

Javed et al. designed an IDS for CAN, called CANintelliIDS. Their principal purpose is to utilize the correlation of data from multiple ECUs to improve detection performance. To this end, they have applied a neural network consisting of two layers of CNN followed by the GRU layer on a continuous flow of CAN messages. The proposed method has shown higher detection performance, around 93% F-score, compared to conventional ML and pure CNN, over OTIDS dataset [6].

Considering the fact that a temporal correlation exists in the messages exchanged on the CAN bus, Taylor et al. introduced LSTM into the network structure. The model predicts the next expected DATA payload for each ECU separately, and the aggregated bit log loss error over the whole word is used for detecting anomalies. By doing so, they could achieve approximately 0.99 AUC on most of the CAN IDs, tested on data collected from a 2012 Subaru Impreza with five types of anomalies: interleave, drop, discontinuity, unusual, and reverse [17].

LSTM is also used by Zhu et al. for anomaly detection. They have distributed the LSTM training on edge devices to reduce the response time and computation power [20]. Longari et al. proposed an LSTM-based Autoencoder as an unsupervised method, called CANnolo, to detect anomalies of independent ECUs and validated it by their own generated CAN message dataset [11]. In the CANet proposed by Hanselmann et al., the LSTM predictions of separate ECUs are reconstructed by feeding into an autoencoder to take the correlation of ECUs into account as well [4].

3 Proposed Method

Each ECU connected to the CAN bus periodically broadcasts measured data in CAN messages. As seen from Fig. 1, the content of messages generated by each ECU is correlate over time. The payload of transmitted messages at each time is affected by the flow of previous message payloads. This observation leads us to use LSTM units in our proposed anomaly detection method.

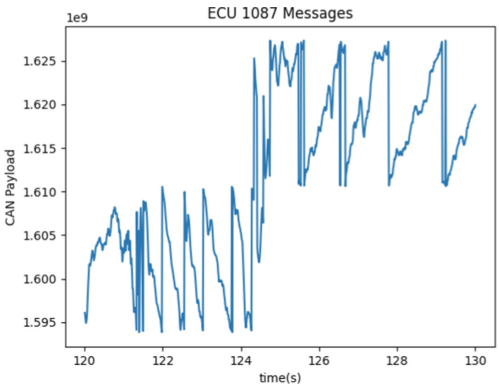

Fig. 1. A snapshot of the CAN message data contents over ten seconds period in the Car Hacking Dataset.

The complete structure of the proposed method is depicted in Fig. 2. It consists of three modules: LSTM network, prediction error calculator, and Gaussian Naïve Bayes (GNB) classifier. The input to our IDS is a 3D vector of integer data in the shape of (#samples, #lookback, #features), and the output is the

probability of samples belonging to two classes of normal activities and attack. We will discuss each module in detail.

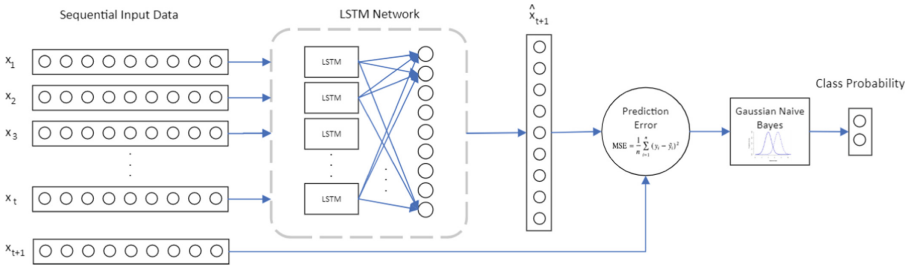

Fig. 2. The overall structure of the proposed LSTM-GNB IDS method.

3.1 Input Data

The input layer is fed with a sequence of data - the length of it is called lookback. We have removed additional features like arbitration ID and timestamp from the dataset and kept nine other features of CAN messages, including the Data Length Control (DLC) and eight bytes of payload as the data. To convert the raw data into the sequence format required by LSTM, we performed a moving window on the data to cut the data into sequential samples.

3.2 LSTM Network

Non-recurrent neural networks, such as MLP and CNN, do not have memory in their structure and predict the output only based on the given input at the current time. So, it is hard to use them in applications where data has sequential and temporal properties. In our case, the CAN message payload generated by an ECU can be considered in a sequence, and its value varies based on the previous observations. Recurrent Neural Network, RNN for short, offers the required ability, by introducing a hidden state as a memory in its structure, to maintain a history of observed data and make predictions based on it.

LSTM, introduced by Hochreiter and Schmidhuber [5], is an enhanced version of RNN. The internal structure of the LSTM unit can be seen in Fig. 3. The improvement involves solving the vanishing gradient problem of RNN by changing the learning strategy from learning what information to remember to learning what information to forget. To do so, three learnable gates in the LSTM structure control the hidden state to remember or forget information flow. First, the forget gate determines how much of the past information should be omitted. Second, the input gate controls the amount of data flow from current input x_t and cell state c_t based on their importance. Third, the output gate tunes how much the unit's current state is outputted. Assuming that i_t, f_t, o_t, c_t, and h_t

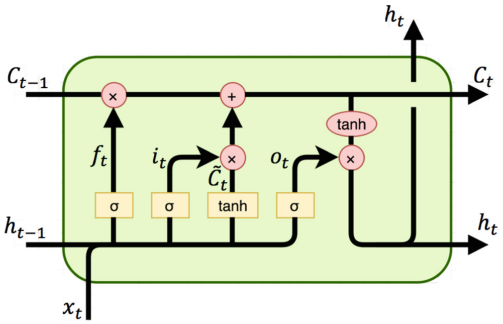

Fig. 3. Internal structure of LSTM unit.

denote the input gate, forget gate, output gate, cell state, and hidden state at timestep t, respectively, the mathematical working structure of LSTM can be calculated as:

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \odot C_{t-1} + b_i) \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \odot C_{t-1} + b_f) \\ C_t &= f_t \odot C_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \odot C_t + b_o) \\ h_t &= o_t \odot \tanh(C_t). \end{aligned} \tag{1}$$

In Eq. 1, $W_h = \begin{pmatrix} W_{hi} \\ W_{hf} \\ W_{hc} \\ W_{ho} \end{pmatrix} \in \mathbb{R}^{\mathrm{dh} \times 4\mathrm{dh}}$ is hidden-to-hidden weight matrix, $W_x = \begin{pmatrix} W_{xi} \\ W_{xf} \\ W_{xc} \\ W_{xo} \end{pmatrix} \in \mathbb{R}^{\mathrm{dx} \times 4\mathrm{dh}}$ is input-to-hidden weight matrix, $b = \begin{pmatrix} b_i \\ b_f \\ b_c \\ b_o \end{pmatrix} \in \mathbb{R}^{4\mathrm{dh}}$ is the bias, and $C_0, h_0 \in \mathbb{R}^{4\mathrm{dh}}$ are the initial value of cell state and hidden state, respectively. Also, $\sigma(.)$ denotes the sigmoid activation function and $\odot$ is the Hadamard product operator.

The LSTM network consists of one LSTM layer and one Dense layer. It extracts the temporal features in the flow of messages in the learning phase and then predicts the expected value of the CAN payload based on the extracted features in the detection phase. The hidden LSTM layer, consisting of 32 units and the tanh activation function, performs the temporal correlation extraction. It is followed by a dense fully-connected output layer with nine neurons, each outputting one predicted feature of data.

The LSTM network is trained only with attack-free CAN messages to extract a model of an in-vehicle network operating normally. Using the ADAM optimizer, the MSE metric, and a learning rate of 0.001, the optimization is performed. The early stop method is implemented to avoid overfitting. Following ten epochs of training, if the trained model fails to improve on the validation data, the training is terminated.

3.3 Prediction Error Calculator

After learning the normal behavior of ECU, the model can be used to differentiate the normal and attack data based on their corresponding prediction error. At each point of time t, the expected status of the ECU message is predicted, denoted by $\hat{x_t}$, based on a sequence of historical data $[x_{t-lookback}, \ldots, x_{t-2}, x_{t-1}]$ given to the LSTM network as the input. The Mean Squared Error (MSE) is used in this module, which can be defined as:

$$MSE = \|x_t - \hat{x_t}\|. \tag{2}$$

The measured prediction error indicates how close the prediction of the LSTM network is to the currently observed value of the CAN message. Since the forecast is according to the normal profile of the network, it tends to be nearly zero for unmodified messages.

3.4 GNB Classifier

In the case of anomaly, the distribution of prediction error would be a Gaussian distribution with different parameters compared to the normal case. According to this observation, we have proposed to use a Gaussian Naïve Bayes classifier on the prediction errors to classify them into two classes of normal and attack data.

A Gaussian Naïve Bayes (GNB) classifier is integrated into the proposed IDS to classify messages based on the prediction error. GNB is an extension of the Naïve Bayes classifier, assuming the real-valued input data has Gaussian distribution with different parameters in different classes. It is a simple yet powerful classifier.

Having $p(C|H)$ as the conditional probability of data belonging to class C by considering hypothesis H and $p(C)$ and $p(H)$ as prior probability of class C and hypothesis H, the Bayes' theorem is stated as:

$$p(H|C) = \frac{p(C|H)p(H)}{p(C)}, \tag{3}$$

where $p(H|C)$ is the posterior probability and indicates the probability of hypothesis H given the class C data. The GNB makes use of Bayes' theorem to calculate the posterior probability of all possible hypotheses based on the prior possibilities, derived from given input data, and choose the one that maximizes the likelihood.

If an attacker injects or drops a message, the sequence of data does not follow the normal profile of the network and the LSTM module fails to predict it perfectly. As a result, if taking a look into the distribution of the predic-

tion errors, the attack and normal data follow different Gaussian distributions. Figure 4 shows an example of prediction error distribution of two different features, using by LSTM module on data of one ECU.

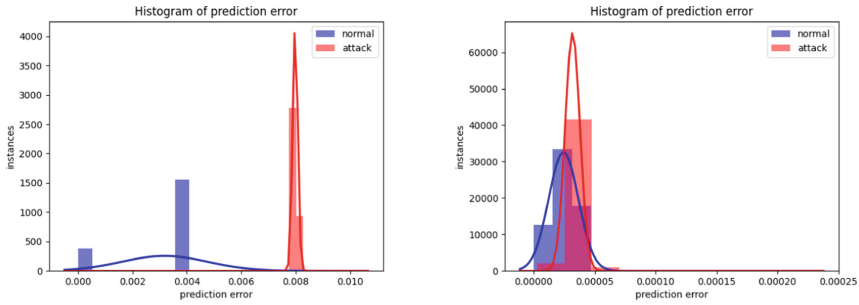

Fig. 4. Prediction error distribution of normal and attack CAN messages and their corresponding Gaussian distribution for two different features.

The GNB classifier module generates the final output, which is the probability of the input sample x_t belonging to classes of normal or attack. Our method selects the class with a higher probability and outputs it. If the observed CAN message is identified as malicious, our IDS will raise the alarm and notify the driver and the manufacturing company.

4 Performance Evaluation

This section discusses the experiments performed to evaluate the efficiency of our proposed LSTM-GNB IDS. Two datasets used for the evaluation are described first, followed by the evaluation procedure and the results.

4.1 Dataset

To evaluate our proposed method, we have made use of two datasets publicly available for CAN bus: Car Hacking Dataset and Survival Analysis Dataset for automobile IDS. The Car Hacking dataset is gathered by connecting two Raspberry Pi devices to the OBD-II port of a real car, one for logging the network traffic and one for injecting fabricated messages to the CAN bus. Four types of attacks, including DoS, fuzzy, RPM gauge, and gear spoofing, are implemented in this dataset. Each message in the data samples consists of a timestamp, CAN ID, DLC, DATA [0-7], and a flag indicating normal (R) or attack data (T).

The other dataset, Survival Analysis Dataset for automobile IDS, contains CAN bus messages of three cars from different vendors: Sonata, Spark, and Soul. For each vehicle, three attacks are performed and logged: Flooding, Fuzzy, and Malfunctioning. The structure of the data is similar to the Car Hacking Dataset. The summary of both datasets is given in Tables 1 and 2.

Table 1. The summary of Survival Analysis Dataset for automobile IDS.

Attack Type	Number of Messages		
	Sonata	Soul	Spark
Flooding	149,547	181,901	120,570
Fuzzy	135,670	249,990	65,665
Malfunction	132,651	173,436	79,787
Attack-Free	117,173	192,516	136,934

Table 2. The summary of the Car Hacking Dataset.

Attack type	Number of Messages		
	Total	Normal Messages	Injected Messages
Normal	988,987	988,987	0
DoS Attack	3,665,771	3,078,250	587,521
Fuzzy Attack	3,838,860	3,347,013	491,847
Gear Spoofing	4,443,142	3,845,890	597,252
RPM Spoofing	4,621,702	3,966,805	654,897

4.2 Results

Several experiments are conducted to evaluate the proposed method and compare it to other baselines. The Keras framework with the backend of Tensorflow is used to build and train the models, on a computer with 16 GB RAM and a Core i7 processor.

Each dataset consists of normal and attacks data files. The normal data is used to train and validate the LSTM network. The other datasets are split into two parts. Twenty percent of them are used to train the GNB, and the rest are fed into the network as test data to evaluate the performance of our method. In addition, the hyperparameters associated with the model are tuned by the grid search method to achieve the highest performance.

Attacks on the CAN bus rarely occur, while the majority of CAN messages represent the normal operation of the vehicle. As a result, the labeled data collected from the CAN bus forms an imbalanced dataset with the anomaly data as the minority class or outliers. Traditional classification algorithms fail in the imbalanced data case and tend to bias toward the overabundant subclass. However, some One Class Classification (OCC) algorithms can deal with imbalanced data. Based on this, the efficiency of our proposed approach was compared to some OCC approaches as a baseline. The baseline includes two one-class conventional classification methods, OCSVM (One-Class Support Vector Machines) with Stochastic Gradient Descent (SGD) and Isolation Forest, as well as a neural network-based method called LSTM Autoencoder. A comparison between the performance of the proposed LSTM-GNB IDS and baselines is given in Table 3.

Table 3. Comparison of proposed LSTM-GNB IDS with Baseline results on the Car Hacking Dataset.

Attack Types	Baseline Methods	Recall	Precision	F-Score	Accuracy
DoS	OCSVM	1.000	0.835	0.910	0.968
	Isolation Forest	1.000	0.718	0.836	0.937
	LSTM-AE	1.000	1.000	1.000	1.000
	LSTM-GNB	**1.000**	**1.000**	**1.000**	**1.000**
Fuzzy	OCSVM	0.987	0.793	0.879	0.965
	Isolation Forest	0.994	0.621	0.764	0.921
	LSTM-AE	0.998	0.977	0.987	0.989
	LSTM-GNB	**1.000**	**0.994**	**0.997**	**0.998**
Gear Spoofing	OCSVM	0 (TP = 0)	0	–	0.833
	Isolation Forest	0 (TP = 0)	0	–	0.835
	LSTM-AE	0.984	0.894	0.937	0.982
	LSTM-GNB	**1.000**	**1.000**	**1.000**	**1.000**
RPM Spoofing	OCSVM	0 (TP = 0)	0	–	0.826
	Isolation Forest	0 (TP = 0)	0	–	0.834
	LSTM-AE	0.995	0.893	0.987	0.982
	LSTM-GNB	**1.000**	**1.000**	**1.000**	**1.000**

Table 4. LSTM-GNB results on the Survival Analysis Dataset.

Attack Types	Baseline Methods	Recall	Precision	F-Score	Accuracy
Flooding	Spark	1.000	1.000	1.000	1.000
	Soul	1.000	1.000	1.000	1.000
	Sonata	1.000	1.000	1.000	1.000
Malfunction	Spark	1.000	1.000	1.000	1.000
	Soul	0.984	1.000	0.992	0.999
	Sonata	1.000	1.000	1.000	1.000
Fuzzy	Spark	0.996	0.997	0.997	0.998
	Soul	0.994	1.000	0.997	0.997
	Sonata	0.996	1.000	0.998	0.998

The Survival Analysis Dataset for automobile IDS consists of normal and attack data gathered from three different car vendors, Sonata, Spark, and Soul. The results on this dataset are shown in Table 4.

As can be seen from the results, our proposed method outperforms the baselines. OCSVM and Isolation Forest fail to detect spoofing attacks. It is because anomalies that happen in the spoofing attack are not point anomalies, but contextual anomalies. Individually, they can be the same as normal data instances and cannot be found by comparing them. However, they are anomalous when happening in the wrong position in a sequence of data points. The ability of our LSTM model to consider each data in a sequence boosts its detection performance.

The other observation from the results is that in both datasets, fuzzy attacks are the hardest type to be detected. In this type of attack, random messages with random IDs are injected into the network. This results in declaring some normal data as an anomaly by mistake and increased false alarms. However, the false positive rate is yet low enough compared to the baseline.

5 Conclusion

In this work, an anomaly-based IDS is proposed to detect attacks on the in-vehicle CAN bus. The proposed method consists of an LSTM trained with normal CAN messages to extract the usual sequential behavior of each ECU. The trained network is used to predict the next expected payload of ECU based on past observations and compare it to the current received value from ECU. The idea behind the proposed method is that when an attack happens, the trained LSTM network will fail to predict correctly, thus the prediction error is higher than normal. To classify them based on the prediction error a GNB classifier is used in the method. The performance of our anomaly detector is evaluated against three one-class classifiers, using two datasets. The results show that our method yields better performance than the baselines, due to its ability to consider the position of messages of each ECU in a time sequence, not individually, and extract more meaningful correlations.

References

1. Al-Saud, M., Eltamaly, A.M., Mohamed, M.A., Kavousi-Fard, A.: An intelligent data-driven model to secure intravehicle communications based on machine learning. IEEE Trans. Industr. Electron. **67**(6), 5112–5119 (2019)
2. Avatefipour, O., et al.: An intelligent secured framework for cyberattack detection in electric vehicles' can bus using machine learning. IEEE Access **7**, 127580–127592 (2019)
3. Barletta, V.S., Caivano, D., Nannavecchia, A., Scalera, M.: Intrusion detection for in-vehicle communication networks: an unsupervised Kohonen Som approach. Future Internet **12**(7), 119 (2020)
4. Hanselmann, M., Strauss, T., Dormann, K., Ulmer, H.: Canet: an unsupervised intrusion detection system for high dimensional can bus data. IEEE Access **8**, 58194–58205 (2020)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
6. Javed, A.R., Ur Rehman, S., Khan, M.U., Alazab, M., Reddy, T.: Canintelliids: detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU. IEEE Trans. Netw. Sci. Eng. **8**(2), 1456–1466 (2021)
7. Kang, M.J., Kang, J.W.: A novel intrusion detection method using deep neural network for in-vehicle network security. In: 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring), pp. 1–5. IEEE (2016)
8. Koscher, K., et al.: Experimental security analysis of a modern automobile. In: 2010 IEEE Symposium on Security and Privacy, pp. 447–462. IEEE (2010)

9. Levi, M., Allouche, Y., Kontorovich, A.: Advanced analytics for connected car cybersecurity. In: 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), pp. 1–7. IEEE (2018)
10. Lin, Y., Chen, C., Xiao, F., Avatefipour, O., Alsubhi, K., Yunianta, A.: An evolutionary deep learning anomaly detection framework for in-vehicle networks-can bus. IEEE Trans. Ind. Appl. (2020)
11. Longari, S., Valcarcel, D.H.N., Zago, M., Carminati, M., Zanero, S.: Cannolo: an anomaly detection system based on LSTM autoencoders for controller area network. IEEE Trans. Netw. Serv. Manag. **18**(2), 1913–1924 (2020)
12. Marchetti, M., Stabili, D.: Anomaly detection of can bus messages through analysis of id sequences. In: 2017 IEEE Intelligent Vehicles Symposium (IV), pp. 1577–1583. IEEE (2017)
13. Müter, M., Asaj, N.: Entropy-based anomaly detection for in-vehicle networks. In: 2011 IEEE Intelligent Vehicles Symposium (IV), pp. 1110–1115. IEEE (2011)
14. Narayanan, S.N., Mittal, S., Joshi, A.: Obd_securealert: an anomaly detection system for vehicles. In: 2016 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 1–6. IEEE (2016)
15. Nilsson, D.K., Phung, P.H., Larson, U.E.: Vehicle ECU classification based on safety-security characteristics. In: IET Road Transport Information and Control-RTIC 2008 and ITS United Kingdom Members' Conference, pp. 1–7. IET (2008)
16. Taylor, A., Japkowicz, N., Leblanc, S.: Frequency-based anomaly detection for the automotive can bus. In: 2015 World Congress on Industrial Control Systems Security (WCICSS), pp. 45–49. IEEE (2015)
17. Taylor, A., Leblanc, S., Japkowicz, N.: Anomaly detection in automobile control network data with long short-term memory networks. In: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 130–139. IEEE (2016)
18. Theissler, A.: Anomaly detection in recordings from in-vehicle networks. Big Data Appl. **23**, 26 (2014)
19. Tian, D., et al.: An intrusion detection system based on machine learning for CAN-bus. In: Chen, Y., Duong, T.Q. (eds.) INISCOM 2017. LNICST, vol. 221, pp. 285–294. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74176-5_25
20. Zhu, K., Chen, Z., Peng, Y., Zhang, L.: Mobile edge assisted literal multi-dimensional anomaly detection of in-vehicle network using LSTM. IEEE Trans. Veh. Technol. **68**(5), 4275–4284 (2019)