Deep Bayesian Learning for Car Hacking Detection

Laha Ale[1], Scott A. King[1], and Ning Zhang[2]

[1]Department of Computing Sciences, Texas A&M University-Corpus Christi
[2]Department of Electrical and Computer Engineering, University of Windsor
{lale.islander, scott.king}@tamucc.edu, ning.zhang@uwindsor.ca
†https://github.com/ainilaha/ppl_car_hacking

Abstract

With the rise of self-drive cars and connected-vehicles, cars are equipped with various devices to assistant the drivers or support self-drive systems. Undoubtedly, cars have become more intelligent as we can deploy more and more devices and software on the cars. Accordingly, the security of assistant and self-drive systems in the cars becomes a life threatening issue as smart cars can be invaded by malicious attacks that cause traffic accidents. Currently, canonical machine learning and deep learning methods are extensively employed in car hacking detection. However, machine learning and deep learning methods can easily be overconfident and defeated by carefully designed adversarial examples. Moreover, those methods cannot provide explanations for security engineers for further analysis. In this work, we investigated Deep Bayesian Learning models to detect and analyze car hacking behaviors. The Bayesian learning methods can capture the uncertainty of the data and avoid overconfident issues. Moreover, the Bayesian models can provide more information to support the prediction results that can help security engineers further identify the attacks. We have compared our model with deep learning models and the results show the advantages of our proposed model. The code of this work is publicly available†.

1 Introduction

With the rise of self-driving cars and Vehicle to Everything (V2X), cars are connected to various devices through wireless internet. Cars are exposed to malicious attacks which may lead to severe traffic accidents. To mitigate these issues, various detection methods have be proposed to analyze and detect intrusions. However, the current existing methods can be overfitting and blinded by carefully designed attacks. Moreover, these methods have no good explanations for the prediction results that support security engineers on further analysis.

Controller Area Network (CAN) is a key component for a connected-vehicle and vulnerable to malicious attacks. Therefore, many researchers have attempted to find reliable intrusion detection methods to detect and analyze the attacks. Anomaly detection methods have been adopted to detect the attacks aimed at vehicles [1, 2]. However, the anomaly detection can only distinguish normal operations and abnormal ones and without further information attached to the prediction results. Seo *et al*. [3] proposed an Intrusion Detection System (IDS) based on Generative Adversarial Networks (GAN) [4]. More recently, Convolutional Neural Networks (CNN) have also been adopted to detect the attacks. Song *et al*. [5] proposed a heavy deep learning model based on Inception-ResNet [6] to detect CAN intrusions, and Hossain *et al*. [7] used a simple CNN and achieved about the same performance. Although the existing methods can achieve considerably high accuracy in detecting the attacks, they cannot capture the uncertainty of the attacks. Moreover, those canonical machine

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

learning and deep learning methods require a massive number of training examples with human effort to label the data set.

In this work, we adopted Bayesian Deep Learning to detect the CAN intrusion. With the proposed Bayesian method, we can provide uncertainty of the prediction and provide more information for security engineers. The Bayesian networks can also capture the uncertainty and capture adversarial attacks. The contributions of this work can be summarized as:

- We developed a Deep Bayesian model for detecting attacks against vehicles that can determines uncertainty of its predictions.
- We introduce a Controller Area Network Intrusion Detection System (IDS) with Deep Bayesian and its training process.
- We present an analysis of the results of using Bayesian predictions on attack detection.

2 Bayesian Networks for Intrusion Detection

The Intrusion Detection System (IDS) works as shown in Fig. 1. The attackers can access the CAN through various wireless networks. The injected attack actions may control the function of Electronic Control Units (ECUs). The ECUs are the components of the control system of connected vehicles. Therefore, the attacks on ECUs of vehicles through the CAN may raise severe security issues. To mitigate this, we developed a DBL model to detect the attacks. Due to the noise of action or disguise, we cannot eliminate Aleatoric uncertainty (statistical uncertainty). IDS can raise warnings or suspend the abnormal operations based on the confidence level of the predictions to reduce risk. The continuous learning by the model requires the storage of operations and labeling of data by a security engineer in situations where the model has low confidence.

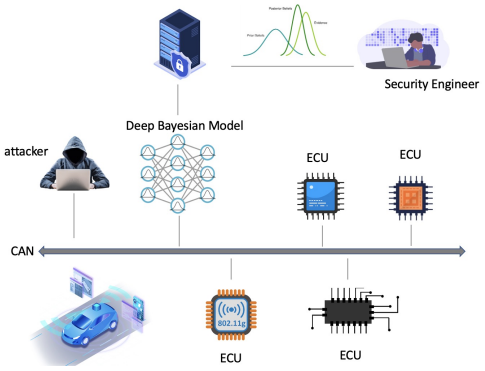


Figure 1: Detection and Training Process.

Unlike training neural networks, Bayesian networks weigh uncertainty [8, 9], which can be given

$$P(w|D) = \frac{P(D|w)P(w)}{\int P(D|w')P(w')\mathrm{d}w'}. \tag{1}$$

We need to specify the prior density $P(w)$, using training data D to determine the likelihood $P(D|w)$. The normalisation constant is $\int P(D|w')P(w')\mathrm{d}w' = P(D)$, which is often intractable; therefore, a variational posterior $q(w|\theta)$ is often computed to approximate the posterior $P(w|D)$. The difference between them can be computed with Kullback–Leibler divergence, given by

$$D_{KL}(q(w|\theta)||P(w|D)) = \int q(w|\theta) \log\left(\frac{q(w|\theta)}{P(w|D)}\right) \mathrm{d}w \tag{2}$$

To update θ and minimize the difference between $q(w|\theta)$ and $P(w|D)$, the loss function can be given,

$$\begin{aligned} L(\theta|D) &= D_{KL}(q(w|\theta)||P(w)) - \mathbb{E}_{q(w|\theta)}(\log P(D|w)) \\ &= \int q(w|\theta)(\log q(w|\theta) - \log P(D|w) - \log P(w))\mathrm{d}w. \end{aligned} \tag{3}$$

Since the integral over w is computationally expensive, the above function can be re-written as expectation and then use reparameterization [8] to update the parameters θ.

$$L(\theta|D) = \mathbb{E}_{q(w|\theta)}(\log q(w|\theta) - \log P(D|w) - \log P(w)). \tag{4}$$

The rest of the update process is the same as standard backpropagation [10], and updates the parameters from the last layer to the first layer.

3 Results Analysis and Discussion

We use a real-world dataset, which can be requested from the website [11]. We adopt TensorFlow [1] to build the deterministic deep learning model and TensorFlow probability [2] for the Deep Bayesian Learning (DBL) models. Each operation and attack action has fixed-length binary numbers, and multiple (sequential) actions can be represented as a 2-dimensional feature that can be fed into convolutional neural networks. As we can see from Fig. 2 and Fig. 3, the deterministic deep learning model can achieve slightly higher categorical accuracy and less loss than DBL models. However, the deterministic deep learning mode is slightly overfitting after 150 epochs. Furthermore, the deterministic deep learning model cannot provide rich information such as correlation as shown in Fig. 4 and Fig. 5 that can assist security engineers to further analyze and identify the risks.

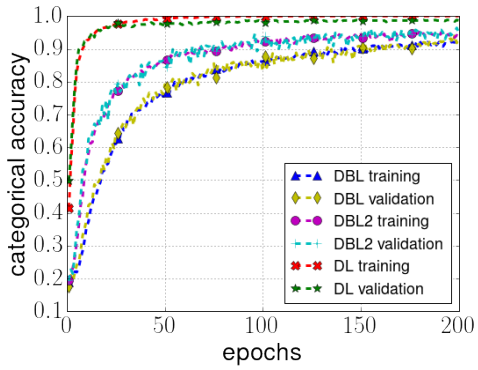


Figure 2: Categorical Accuracy

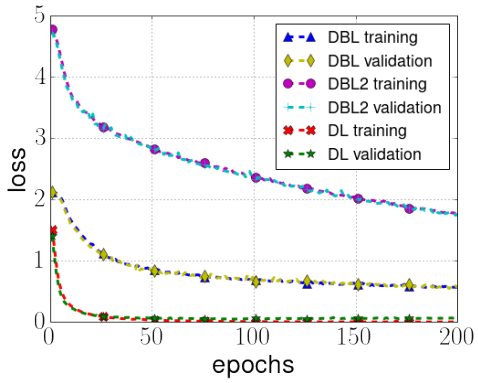


Figure 3: Loss

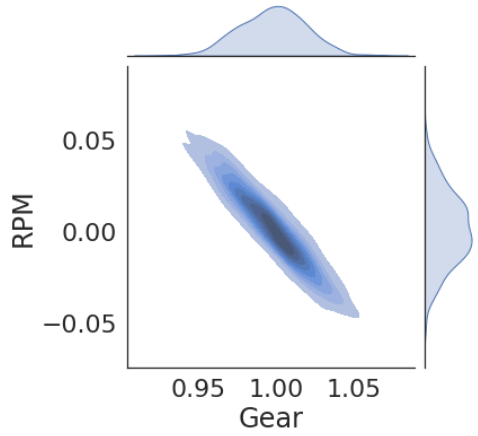


Figure 4: Correlation

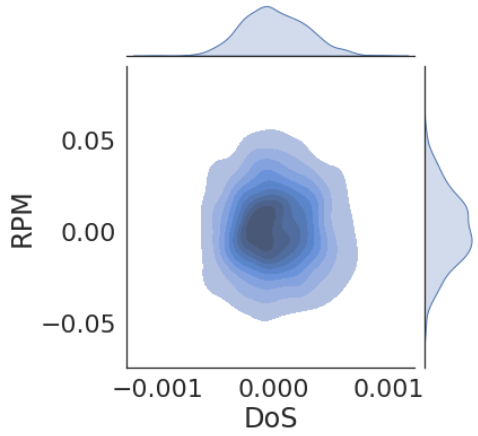
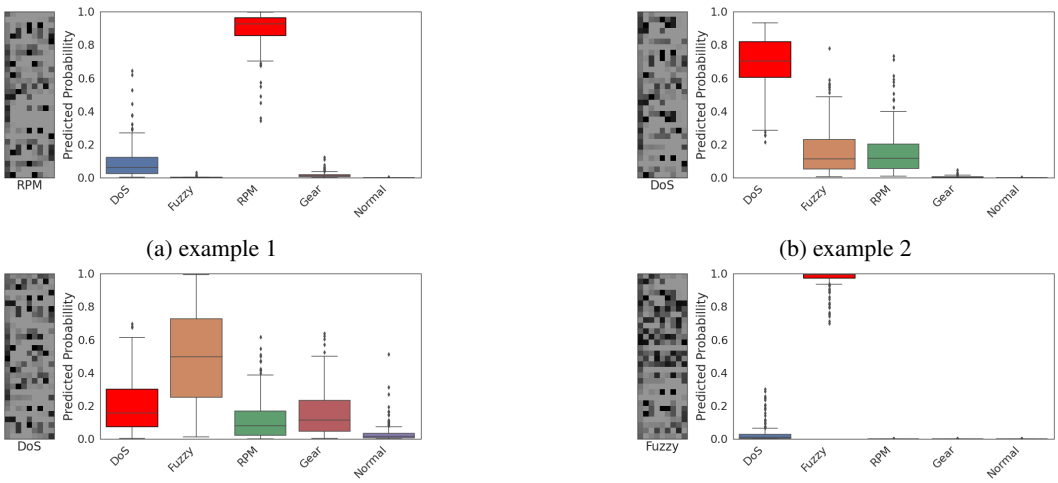


Figure 5: Correlation

Moreover, BDL models can show the uncertainty of its predictions instead of being overconfident like the deterministic models. The models show they do not know when the confidence is low. Our dataset [3] has five actions types, including normal, DoS, Fuzzing, RPM, and Gear Spoofing attacks, and the true label is colored red in Fig. 6. Although the model has classified the operations correctly in the example Fig. 6-(a) and (b), it still shows the uncertainty of the predictions, which is given as the probability of each class. In Fig. 6-(c) the model classified the operation incorrectly but was not overly confident about the prediction. In the real world, we can take countermeasures given the confidence of the prediction of BDL models but not with deterministic models. Note that probability or so called confidence is not actually show the confidence because results are normalized.

[1] https://www.tensorflow.org

[2] https://www.tensorflow.org/probability



(a) example 1

(b) example 2

(c) example 3

(d) example 4

Figure 6: Prediction Examples

Theoretically, we can reduce the Epistemic uncertainty by collecting more data and improving the performance of the models. However, the Bayesian or probabilistic models often require more data to reduce the Epistemic uncertainty than deterministic models. Moreover, it is challenging to set up prior distributions and incorporate prior knowledge into complexity models like Bayesian networks.

4 Conclusions

In this work, we proposed an Intrusion Detection System using a Deep Bayesian Learning (DBL) model. We compare the DBL model with deterministic deep learning by using both types of models on a real-world dataset and show the advantages of the DBL model. Specifically, the DBL model can provide more information about its prediction than deterministic models. The additional information can help security engineers with further analysis of abnormal behaviours and label the data. The DBL model can provide the uncertainty of the prediction and show its confidence of the predictions. However, the DBL model has a higher Epistemic uncertainty and is difficult to incorporate prior knowledge and reduce training cost. For future work, we will investigate training methods that can reduce Epistemic uncertainty.

References

[1] M. Marchetti and D. Stabili, "Anomaly detection of can bus messages through analysis of id sequences," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1577–1583.

[2] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 1110–1115.

[3] E. Seo, H. M. Song, and H. K. Kim, "Gids: Gan based intrusion detection system for in-vehicle network," in *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, 2018, pp. 1–6.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 3, no. January, pp. 2672–2680, 2014.

[5] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Communications*, vol. 21, p. 100198, 2020.

[6] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.

[7] M. Delwar Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "An effective in-vehicle can bus intrusion detection system using cnn deep learning approach," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

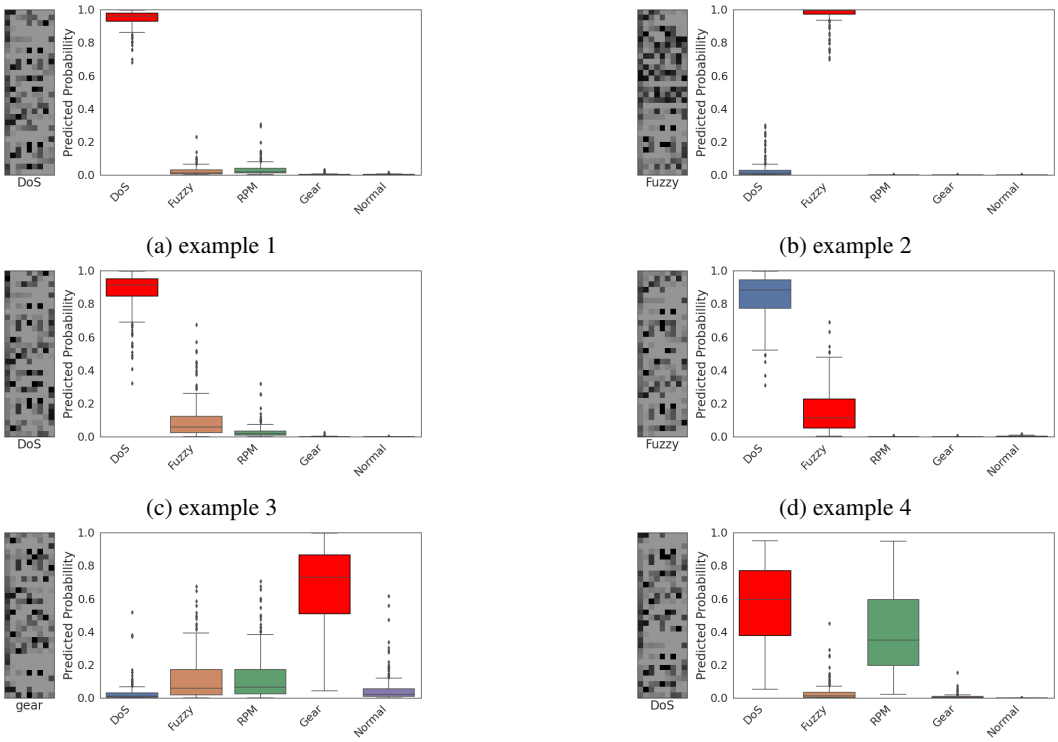
[8] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 1613–1622.

[9] A. G. Wilson and P. Izmailov, "Bayesian deep learning and a probabilistic perspective of generalization," in *Advances in Neural Information Processing Systems*, vol. 2020-Decem, no. NeurIPS, 2020.

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Representations by Back-propagating Errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[11] H. M. Song and H. K. Kim, "Car-hacking dataset." [Online]. Available: https://ocslab.hksecurity.net/Datasets/CAN-intrusion-dataset

5 Appendix



(a) example 1 (b) example 2

(c) example 3 (d) example 4

(e) example 5 (f) example 6

Figure 7: More Prediction Examples.

Note that most of the predictions have high confidence. The prediction examples show the uncertainty of the predictions.