Fingerprinting ECUs to Implement Vehicular Security for Passenger Safety Using Machine Learning Techniques

Samuel Bellaire(✉), Matthew Bayer, Azeem Hafeez, Rafi Ud Daula Refat, and Hafiz Malik

University of Michigan-Dearborn, Dearborn, MI 48128, USA
srbellai@umich.edu

Abstract. The Controller Area Network (CAN) protocol used in vehicles today was designed to be fast, reliable, and robust. However, it is inherently insecure due to its lack of any kind of message authentication. Despite this, CAN is still used extensively in the automotive industry for various electronic control units (ECUs) and sensors which perform critical functions such as engine control. This paper presents a novel methodology for in-vehicle security through fingerprinting of ECUs. The proposed research uses the fingerprints injected in the signal due to material imperfections and semiconductor impurities. By extracting features from the physical CAN signal and using them as inputs for a machine learning algorithm, it is possible to determine the sender ECU of a packet. A high classification accuracy of up to 100.0% is possible when every node on the bus has a sufficiently different channel length.

Keywords: Machine learning · Fingerprinting · Classification · k-NN · Gaussian naive bayes · Multinomial logistic regression · Vehicle cybersecurity · In-vehicle networks · CAN

1 Introduction

Nearly every vehicle on the road today utilizes Controller Area Network (CAN) as one of its primary in-vehicle networks (IVNs) to interface various sensors and electronic control units (ECUs) inside the vehicle [9]. The CAN standard was adopted in the automotive industry due to its high reliability and noise immunity, but one significant drawback of CAN is its lack of a security layer.

CAN bus is a message-based communication protocol where each message sent on the bus is given a unique identifier. When an ECU receives a message, it is impossible to determine the source of the message since there is no identifier for the sender in the CAN frame. This architecture makes CAN bus vulnerable to spoofing attacks, where an external attacker can gain control of any ECU on the bus and impersonate another ECU. In a vehicle, this type of attack could endanger the occupants if critical vehicle functions are disrupted at high speeds.

K. Arai (Ed.): IntelliSys 2022, LNNS 544, pp. 16–32, 2023.
https://doi.org/10.1007/978-3-031-16075-2_2

This was demonstrated in 2015 by Miller et al. [26], who discovered vulnerabilities in a 2014 Jeep Cherokee that allowed them remote access to the vehicle's engine controller and steering module, among other things. No modifications to the vehicle were required to disrupt these critical vehicle functions, and it was discovered that this type of exploitation could be accomplished at great distance through a cellular network.

In this paper, we propose a novel intrusion detection system to defend against spoofing attacks in IVNs. This system links CAN packets to the sender ECU by using unique characteristics extracted from voltage measurements of the CAN-High signal. The features are then used in machine learning algorithms to associate the message with an ECU and determine if the message is legitimate.

In the past, several researchers have used voltage characteristics to fingerprint ECUs [5,20]. Unlike them, we have used seven novel signal characteristics, extracted mostly from noise and rising edge transients, as features for our machine learning classifiers. The extraction and selection of these features are an important part of machine learning based IDS's. This is because the performance of a machine learning system depends on the quality of the selected features just as much as the algorithm itself. To increase the interpretability of our IDS, the feature extraction and selection process is explained in detail in Sect. 4. The effectiveness of the selected features were evaluated with k-nearest neighbors, Gaussian naive bayes, and multinomial logistic regression using a publicly available dataset.

The paper is organized as follows. In Sect. 2, the threat model is defined for the CAN bus network. In Sect. 3, the related work section is presented. In Sects. 4, 5, and 6, the machine learning pipeline is detailed (i.e. data acquisition & preprocessing, model selection & parameter tuning, and model validation & results). Finally, the paper is concluded in Sect. 7.

2 Threat Model

The primary threat model that this paper will examine is the case where an external attacker gains control of one or more ECUs in the vehicle. Systems such as the infotainment system, which has wireless connectivity features, allows the attacker to gain remote access without physical modifications to the vehicle. If the compromised ECU has access to the vehicle's CAN bus, then it would be capable of manipulating other ECUs on the bus to a certain degree, as shown in *Fig*. 1.

Once an attacker gains control of an ECU, there are three types of attacks they can launch on the CAN bus: spoofing, fabrication, and bus-off attacks. Each attack is explained in more detail below. In this paper, the primary focus is on the spoofing attack.

2.1 Spoofing Attack

Spoofing attacks, sometimes called impersonation attacks, occur when an attacker takes control of an ECU and mimics the behavior of another ECU on the bus. Typically, the ECU to be impersonated is disabled before the attacking ECU begins to send CAN messages in its place.

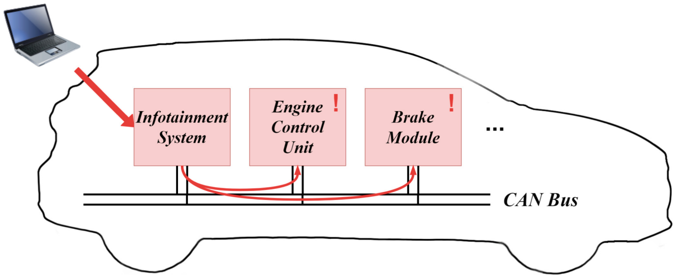

Fig. 1. IVN threat model

2.2 Fabrication Attack

Fabrication attacks occur when an attacker injects additional messages on the CAN bus through a compromised ECU. These messages can override previously sent messages to disrupt communication. A fabrication attack can also result in denial of service (DoS) if the adversary floods the bus with high-priority messages that block other ECUs from transmitting.

2.3 Bus-Off Attack

A bus-off attack disables the compromised ECU, and it ceases all activity on the CAN bus. This can disrupt time-sensitive communications if other ECUs are dependent on the messages sent by the compromised ECU.

3 Related Work

As shown in *Fig.* 2, existing approaches for IVN security can be separated into two primary categories: message authentication code (MAC) based approaches [6,17,31,35,37] and intrusion detection system (IDS) based approaches. IDS's can be further broken down into parameter monitoring [16,21,29,32], information theory [23,30,38], machine learning [10,18,19,22,24,27,33,36], and fingerprinting [1–5,8,11–15,20,28,34] based approaches.

3.1 MAC Based Approach

MAC based systems are the most traditional implementations of IVN security. In CAN, this is accomplished by appending some form of encrypted sender authentication to the message.

One such implementation of a MAC based system is LCAP [17], developed in 2012 by Hazem and Fahmy. The system works by implementing a 16-bit "magic number", either occupying 2 bytes in the message or utilizing the extra identifier bits provided by extended CAN.

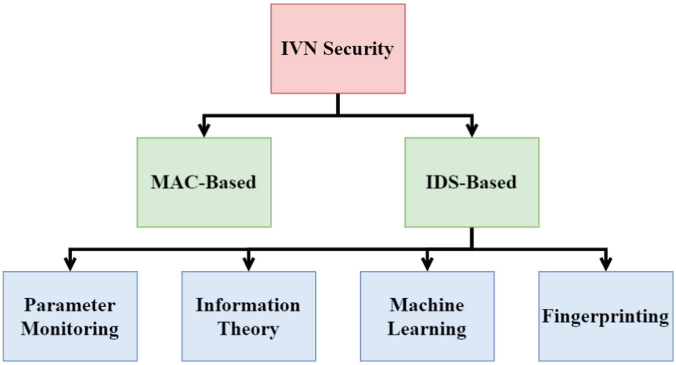

Fig. 2. IVN security categories

While MAC based security can be effective, it can be computationally expensive and be easily nullified if the adversary gains access to the encryption key. In addition, many protocols such as CAN already have a significant overhead without a MAC. In the case of LCAP, which can occupy 25% of the payload with its "magic number" [17], the data bandwidth is reduced even more.

3.2 Parameter Monitoring IDS Approach

Since many CAN bus messages are broadcasted periodically, IDS's based on parameter monitoring typically examine message frequency or the time difference between a remote frame and reply. A database of expected message frequencies or timings can be created, and deviations from these timings can be marked as anomalies, as seen in *Fig*. 3. This approach is excellent at detecting DoS and bus-off attacks, but can fall short if the adversary is timing-aware.

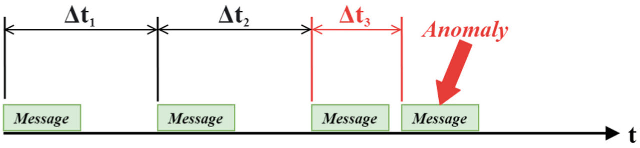

Fig. 3. Concept of frequency based IDS

In 2015, Taylor et al. [32] was able to develop a timing-based system that detected upwards of 96.2% of attacks that altered the frequency of messages on a CAN bus, from message deletion up to ten times the typical frequency.

3.3 Information Theory IDS Approach

IDS's based on information theory examine the information entropy of messages on the CAN bus. In 2016, Marchetti et al. [23] demonstrated that certain types of attacks can cause changes in entropy compared to the typical value for the message. This led to accurate detection of anomalies for most messages on the bus, but for the few IDs with varying entropy, the method proved to be less effective.

3.4 Machine Learning IDS Approach

Machine Learning IDS's utilize various artificial intelligence models, such as neural networks and AI-based classifiers, to implement IVN security. One example of such a system is the deep learning IDS developed by Kang et al. [19]. In their IDS, the probability distributions of bits in the CAN packet's payload was used as a set of features for the model. This approach was able to classify legitimate and malicious messages with an accuracy of 97.8%.

3.5 Fingerprinting IDS Approach

Fingerprinting is a method that examines the non-ideal characteristics of a transmitter, such as semiconductor impurities or parasitics in its communication channel. These characteristics can be exploited to create a unique fingerprint for each ECU. A rather extreme case can be seen in *Fig.* 4, where it would be easy to differentiate the two ECUs through features such as transient response length. This ability to differeniate between ECUs makes fingerprinting an excellent candidate to detect spoofing attacks. It is, however, ineffective against bus-off attacks.

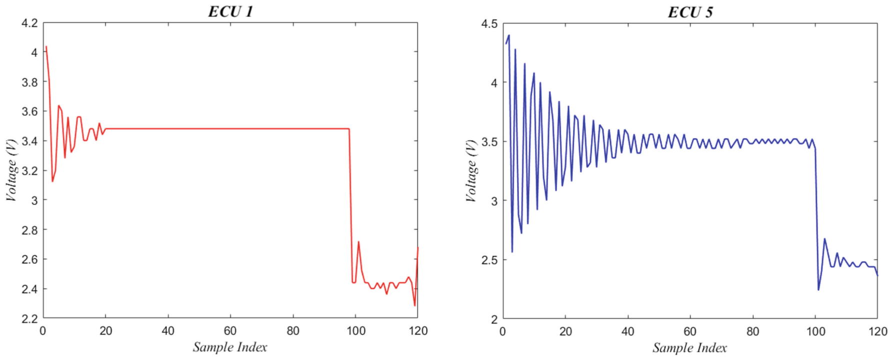

Fig. 4. Physical signal differences between ECUs

In 2018, Choi et al. developed VoltageIDS [5], a fingerprinting based IDS that utilizes various time-domain and frequency-domain features, such as mean, standard deviation, and kurtosis, to classify the source ECU of a signal. Using Linear SVM, they achieved F-Scores of up to 99.7% with just 70 samples per ECU.

The technology used in many IDS's for IVNs can also be utilized in broader applications. In 2020, Hady et al. [7] created the Enhanced Healthcare Monitoring System, an IDS for a hospital's network that combined patient bio-metric data with network data to act as features. In addition to ECUs, fingerprinting based approaches can also be applied to various other sensors and signals. Zuo et al. [39] collected radio frequency emission data from UAV video signals, and were able to successfully differentiate the signals from WiFi interference.

4 Data Acquisition and Preprocessing

4.1 Data Collection

Data collection was carried out using seven different MCP 2515 CAN bus shields mounted on Arduino UNO-R2 microcontrollers, each with a different channel length, varying from 2m to 10m. The CAN-High signal was recorded using a DSO1012A oscilloscope sampling at 20 MSa/s. For each of the ECUs, thirty records were captured with approximately 600 samples in each record, amounting to a pulse train 4–5 pulses in length. *Fig.* 5 shows a few pulses of CAN-High data from ECU 1.

4.2 Pulse Segmentation

Segmenting each record into individual pulses was done using a simple thresholding algorithm, the pseudo code for which can be seen in Algorithm 1.

Algorithm 1. Pulse Segmentation Algorithm

```
$k \leftarrow 0$
for $i \leftarrow 0$ to $N - 2$ do
    if $y(i) < 3$ and $y(i+1) \geq 3$ then
        $indexList(k) \leftarrow (i+1)$
        $k \leftarrow (k+1)$
        $i \leftarrow (i+B-1)$
    end if
end for
```

Whenever the raw signal $y(n)$ (which is of length N samples) transitions from below 3 V to above 3 V, the current index is saved in a list of indices. This list is later used to segment the data. In some cases the transient period of the signal's rising edge has a large amplitude and can cross the 3 V threshold multiple times. To bypass this issue, B samples can be skipped after finding a valid rising edge to ensure the transient has dissipated sufficiently. $B = 20$ was used in this case.

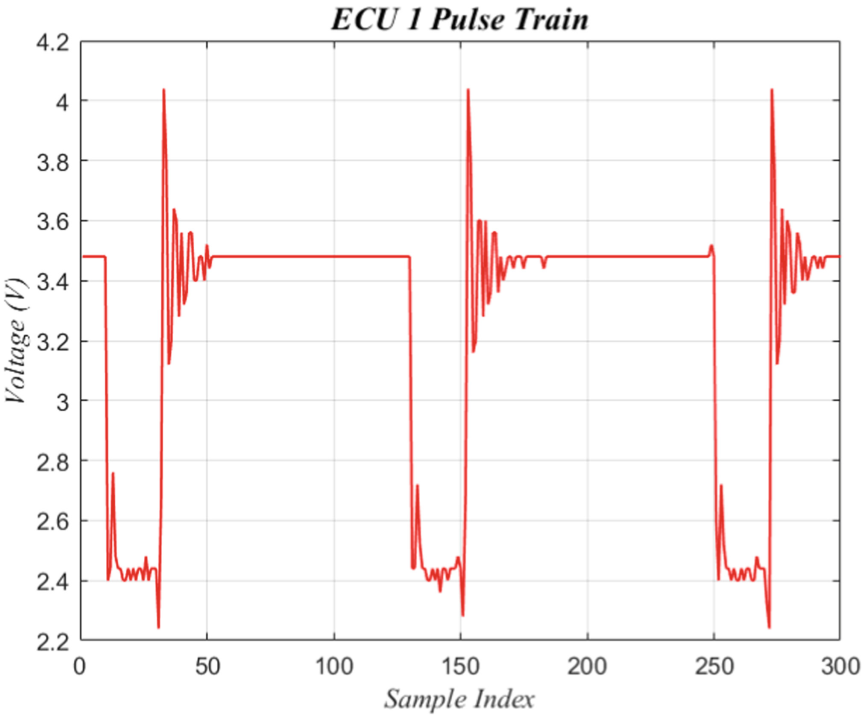

Fig. 5. Pulse train from ECU 1

Figure 6 shows an example of a single pulse extracted from the pulse train from $Fig.$ 5. By extracting individual pulses, it makes it easier to extract some signal attributes (e.g. from only the dominant bit, or only the transient period).

4.3 Feature Extraction

After singular pulses were isolated from the data, seven features were extracted from each pulse to form a feature set for the classifiers. Each of these features are described below.

Transient Response Length. The time required for the transient response to settle to within some threshold α of the signal's steady-state value V_S. This can be found by starting near the falling edge of the pulse, denoted by the sample N_F, where we have good confidence that the signal is in the steady-state, and working backwards until the first value that exceeds the threshold α is found. Pseudo code for this algorithm can be seen in Algorithm 2.

Maximum Transient Voltage. The maximum value of the signal observed during the transient period, as shown in Eq. 1

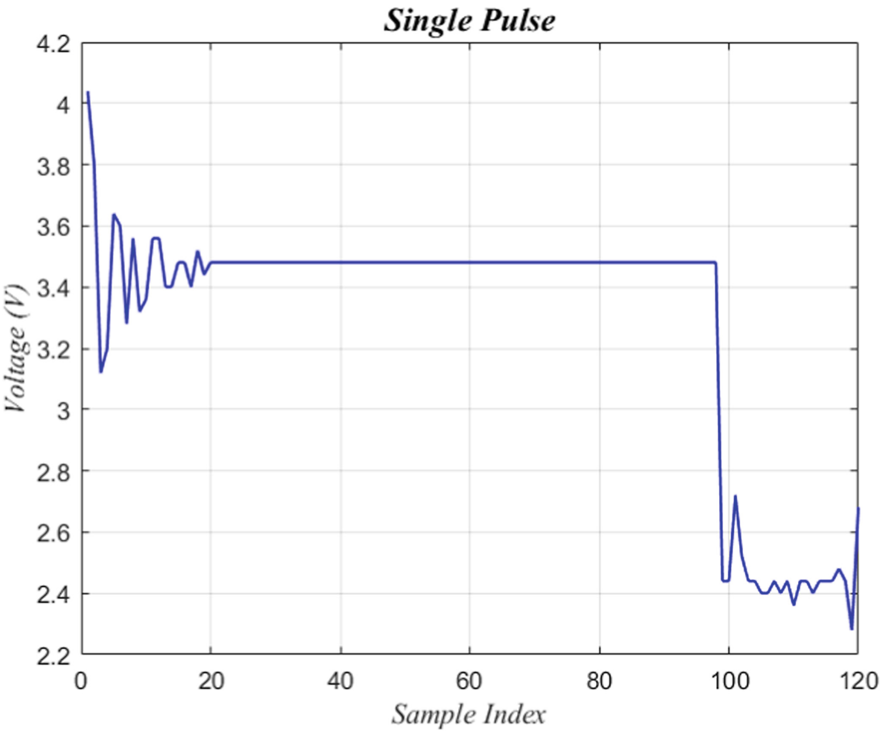

Fig. 6. Single pulse from ECU 1

$$V_{max} = max(y(n)) \tag{1}$$

Energy of the Transient Period. The sum of the squared signal over the transient period, as seen in Eq. 2, where N_T is the length of the transient period in samples.

$$E_y = \sum_{n=0}^{N_T-1} |y(n)|^2 \tag{2}$$

Average Dominant Bit Steady-State Value. The average value of the dominant bit during the steady state. As seen in Eq. 3, this can be done by averaging the samples from N_S to N_F, where N_S is the sample at the beginning of the steady-state, and N_F is the last sample before the falling edge of the pulse.

$$\overline{V_D} = \frac{1}{N_F - N_S + 1} \sum_{n=N_S}^{N_F} y(n) \tag{3}$$

Algorithm 2. Transient Length Algorithm

```
for $i \leftarrow N_F$ to 0 do
    if $|y(i) - V_S| > \alpha$ then
        $l_{tr} = i + 1$
        break
    end if
end for
```

Peak Noise Frequency. The frequency at which the peak value occurs in the magnitude spectrum. To determine this, the noise of the signal must first be isolated to ensure data independence in the Fast Fourier Transform (FFT). Thus, the ideal signal must be approximated from the pulse. Since the pulse begins at the rising edge, the only parameter that must be found to construct the ideal signal is the sample N_F at which the falling edge occurs. This can be done using a simple 3V threshold, as seen in Algorithm 3.

Algorithm 3. Finding Falling Edge Algorithm

```
for $i \leftarrow 0$ to $N - 1$ do
    if $y(i) \leq 3$ then
        $N_F = i$
        break
    end if
end for
```

With N_F calculated, the ideal signal can be acquired, as seen in *Fig.* 7. This signal is 3.5 V during the dominant bit and 2.5 V during the recessive bit. The ideal pulse can then be subtracted from the actual pulse, leaving the noise behind. Additionally, the mean value of the noise can also be subtracted to remove the DC component of the noise signal, as seen in *Fig.* 8.

The FFT of the noise signal with DC component removed can then be taken to determine the peak frequency of the pulse's noise, such as in *Fig.* 9.

Average Noise. The average value of the noise signal whose length is N samples, such as the one shown in *Fig.* 8. See Eq. 4

$$\overline{V} = \frac{1}{N} \sum_{n=0}^{N-1} y(n) \tag{4}$$

Standard Deviation of Noise. The standard deviation of the noise signal whose length is N samples, such as the one shown in *Fig.* 8. See Eq. 5

$$\sigma = \sqrt{\frac{\sum (x_i - \overline{V})^2}{N}} \tag{5}$$

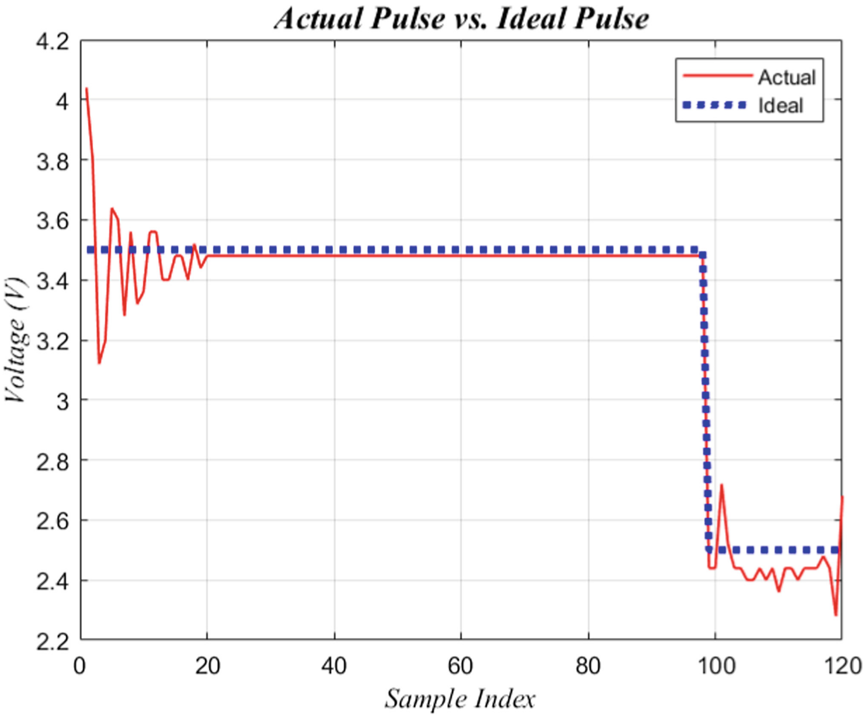

Fig. 7. Actual vs. Ideal signal

4.4 Outlier Removal

Outlier removal was performed as in Eq. 6. For a given class c_i, the j^{th} data point x_{ij} is removed if any of its features differ from the class mean μ_i of that feature by more than 3 standard deviations.

$$|x_{ij} - \mu_i| > 3\sigma_i \tag{6}$$

5 Model Selection and Parameter Tuning

Three different machine learning models were used to perform ECU classification using the extracted features: k-Nearest Neighbor (k-NN), Gaussian Naive Bayes (GNB), and Multinomial Logistic Regression (MLR). All of these models were implemented using MATLAB's Statistics and Machine Learning Toolbox [25]. A brief overview of each of these methods can be seen below.

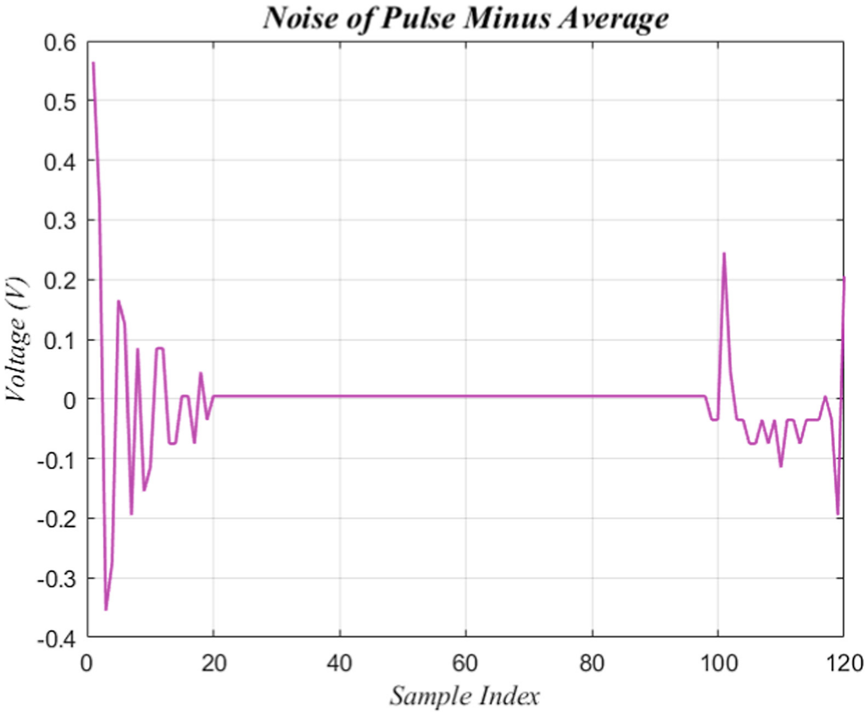

Fig. 8. Noise of pulse from ECU 1

5.1 K-Nearest Neighbor

k-NN is a simple machine learning algorithm that maps observed data points x_i to an n-dimensional feature space. $Fig.$ 10 shows an example of three different features represented in a 3-dimensional space.

When an unknown data point u is input to the model, k-NN calculates the distance between u and every observed data point x_i, and uses the k closest observations to make a classification decision.

5.2 Gaussian Naive Bayes

GNB is a specific case of the Naive Bayes (NB) classification method that uses a Gaussian distribution. To predict the class of an unknown data point u, the GNB classifier leverages Bayes' Theorem (see Eq. 7). Since GNB is an implementation of NB, it makes the "naive" assumption that all of the features are independent, which greatly simplifies the calculation of $P(u)$. After calculating $P(c_i|u)$ for each class c_i, a prediction is made based on the highest probability.

$$P(c_i|u) = \frac{P(u|c_i)P(c_i)}{P(u)} \tag{7}$$

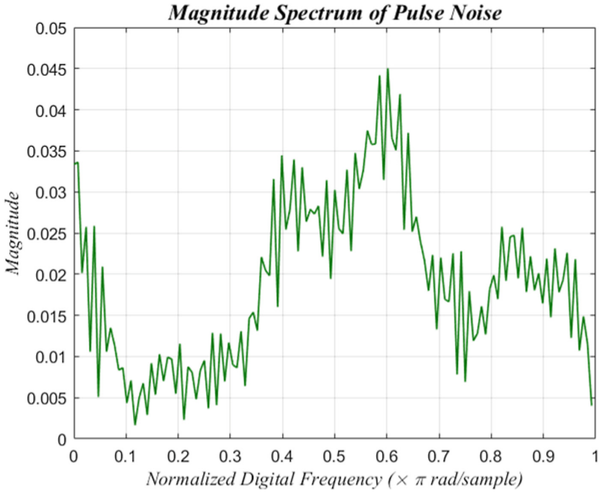

Fig. 9. Magnitude spectrum of pulse noise

5.3 Multinomial Logistic Regression

MLR is a generalization of linear regression to a classification problem with more than two classes. Similarly to GNB, MLR assigns a probability for u belonging to class c_i for each class i. The classification decision is then made based on the highest probability.

5.4 Model Tuning and Parameter Selection

The tuning phase was implemented through adjusting various parameters of each model to determine which sets of parameters result in the best model performance. Each of the three methods implemented using MATLAB have their own unique parameters that can be changed when fitting the model.

K-NN Tuning. A list of parameters that were tested for the k-NN classifier can be seen below. Ultimately, the parameters selected were $k = 1$, and a euclidean distance metric with equal distance weighting.

- Value of k from 1 → 31
- Distance calculation method (euclidean, hamming, cityblock)
- Distance weights (equal, inverse, inverse squared)

A set of three features was used in the k-NN classifier, which can be seen listed below.

- Transient response length
- Maximum transient voltage
- Energy of the transient period

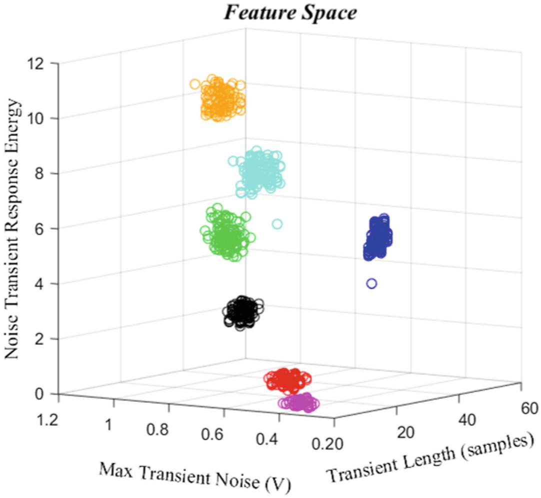

Fig. 10. Example of a 3-D feature space

GNB Tuning. A list of parameters adjusted for the GNB classifier can be seen below. For this classifier, a kernel distribution with a normal kernel type produced the best results.

- Distribution type (kernel, multinomial, multivariate multinomial, normal)
- Kernel type (box, epanechnikov, Gaussian, triangular, normal)

For the GNB classifier, an extra feature was added compared to k-NN for a 4-dimensional feature set. These features are listed below.

- Transient response length
- Maximum transient voltage
- Energy of the transient period
- Average Dominant Bit Steady-State Value

MLR Tuning. A list of parameters adjusted for the MLR classifier can be seen below. A hierarchical model with a logit link function was utilized for MLR.

- Model type (nominal, ordinal, hierarchical)
- Link function (logit, probit, log-log, complementary log-log)

For the MLR classifier, poor performance was observed for smaller feature sets. The model performed best when all seven features, detailed in Sect. 4, were used in the classification algorithm.

6 Model Validation and Results

Model validation and testing was performed using a 70–30 split in the data set, which amounted to approximately 560 training data points and around 280 testing data points, split evenly across all 7 ECUs. Accuracy was the primary metric used in evaluating model performance, as in eq. 8. N_C denotes the number of correct classifications, and N_T the number of total classifications made.

$$Acc. = \frac{N_C}{N_T} \tag{8}$$

Table 1. Confusion matrix for 1-NN and GNB classifiers (7 ECUs)

Predicted Class									
–	–	E_1	E_2	E_3	E_4	E_5	E_6	E_7	Acc. (%)
True Class	E_1	41	0	0	0	0	0	0	100.0
	E_2	0	40	0	0	0	0	0	100.0
	E_3	0	0	42	0	0	0	0	100.0
	E_4	0	0	0	42	0	0	0	100.0
	E_5	0	0	0	0	36	0	0	100.0
	E_6	0	0	0	0	0	42	0	100.0
	E_7	0	0	0	0	0	0	42	100.0
	Acc. (%)	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

Table 2. Confusion matrix for MLR classifier (7 ECUs)

Predicted Class									
–	–	E_1	E_2	E_3	E_4	E_5	E_6	E_7	Acc. (%)
True Class	E_1	36	0	0	0	1	0	0	97.3
	E_2	0	37	0	0	0	1	0	97.4
	E_3	0	0	38	0	0	0	1	97.4
	E_4	0	0	0	35	0	1	3	89.7
	E_5	0	0	0	0	37	0	0	100.0
	E_6	1	3	0	0	0	34	0	89.5
	E_7	0	0	0	0	0	0	41	100.0
	Acc. (%)	97.3	92.5	100.0	100.0	97.4	94.4	91.1	95.9

As can be seen in Table 1, both the k-NN classifier and GNB classifier deliver perfect classification for this dataset, with an accuracy of 100.0%. The MLR classifier, while faring much worse with a per-class accuracy as low as 89.5%, still delivers an overall accuracy of 95.9% as shown in Table 2.

The reason that the accuracy is so high for this data set is because of the distinct separation between ECUs in the feature space, as can be seen in *Fig.* 10. Since each of the ECUs in the dataset have a different channel length, they all have different transient characteristics that can be easily distinguished.

7 Conclusion

Connected vehicles are becoming more prevalent every day, and with autonomous vehicle technology on the horizon, security for IVNs is becoming more important than ever before. The fingerprinting methodology we presented in this paper was able to correctly identify the sender ECU of a CAN packet with a high degree of accuracy. Perfect classification was achieved when the ECU channel lengths are sufficiently different, and up to 95.95% accuracy was achieved when this is not possible. Such a system would be cost-effective and could be easily implemented into an IVN with minimal modifications to the network, only requiring an additional ECU to be inserted as the IDS. Thus, we believe that this IDS would be an effective solution to combating spoofing attacks in IVNs.

Acknowledgment. The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for supporting this work through the project # DRI-KSU-934. This research is also partly supported by National Science Foundation (NSF) under the award # 2035770.

References

1. Avatefipour, O., Hafeez, A., Tayyab, M., Malik, H.: Linking received packet to the transmitter through physical-fingerprinting of controller area network. In: 2017 IEEE Workshop on Information Forensics and Security (WIFS), IEEE (2017)
2. Cho, K.T., Shin, K.: Viden: attacker identification on in-vehicle networks. In: 2017 ACM SIGSAC Conference on Computer and Communications Security, ACM, pp. 1109–1123 (2017)
3. Cho, K.T., Shin, K.G.: Fingerprinting electronic control units for vehicle intrusion detection. In: USENIX Security Symposium, pp. 911–927 (2016)
4. Choi, W., Jo, H.J., Woo, S., Chun, J.Y., Park, J., Lee, D.H.: Identifying ecus using inimitable characteristics of signals in controller area networks. IEEE Trans. Veh. Technol. **67**(6), 4757–4770 (2018)
5. Choi, W., Joo, K., Jo, H.J., Park, M.C., Lee, D.H.: Voltageids: low-level communication characteristics for automotive intrusion detection system. IEEE Trans. Inf. Forensics Secur. **13**(8), 2114–2129 (2018)
6. Doan, T.P., Ganesan, S.: Can Crypto FPGA Chip to Secure Data Transmitted through CAN FD Bus using AES-128 and SHA-1 Algorithms with a Symmetric Key. Technical report, SAE Technical Paper (2017)
7. Hady, A.A., Ghubaish, A., Salman, T., Unal, D., Jain, R.: Intrusion detection system for healthcare systems using medical and network data: a comparison study. IEEE Access **8**, 106576–106584 (2020)
8. Hafeez, A.: A robust, reliable and deployable framework for in-vehicle security (2020)

9. Hafeez, A., Malik, H., Vatefipour, O., Raj Rongali, P., Zehra, S.: Comparative study of CAN-bus and flexray protocols for in-vehicle communication. Technical report, SAE Technical Paper (2017)
10. Hafeez, A., Mohan, J., Girdhar, M., Awad, S.S.: Machine Learning based ECU detection for automotive security. In: 2021 17th International Computer Engineering Conference (ICENCO), IEEE (2021)
11. Hafeez, A., Ponnapali, S.C., Malik, H.: Exploiting channel distortion for transmitter identification for in-vehicle network security. In: SAE International Journal of Transportation Cybersecurity and Privacy, 3(11-02-02-0005) (2020)
12. Hafeez, A., Rehman, K., Malik, H.: State of the Art Survey on Comparison of Physical Fingerprinting-Based Intrusion Detection Techniques for in-Vehicle Security. Technical report, SAE Technical Paper (2020)
13. Hafeez, A., Tayyab, M., Zolo, C., Awad, S.: Finger printing of engine control units by using frequency response for secure in-vehicle communication. In: 2018 14th International Computer Engineering Conference (ICENCO), IEEE, pp. 79–83 (2018)
14. Hafeez, A., Topolovec, K., Awad, S.: ECU fingerprinting through parametric signal modeling and artificial neural networks for in-vehicle security against spoofing attacks. In: 2019 15th International Computer Engineering Conference (ICENCO), IEEE, pp. 29–38 (2019)
15. Hafeez, A., Topolovec, K., Zolo, C., Sarwar, W.: State of the Art Survey on Comparison of CAN, FlexRay, LIN Protocol and Simulation of LIN Protocol. Technical report, SAE Technical Paper (2020)
16. Han, M.L., Lee, J., Kang, A.R., Kang, S., Park, J.K., Kim, H.K.: A statistical-based anomaly detection method for connected cars in Internet of things environment. In: Hsu, C.-H., Xia, F., Liu, X., Wang, S. (eds.) IOV 2015. LNCS, vol. 9502, pp. 89–97. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27293-1_9
17. Hazem, A., Fahmy, H.: LCAP- a lightweight CAN authentication protocol for securing in-vehicle networks. In: 10th ESCAR Embedded Security in Cars Conference, vol. 6 (2012)
18. Jain, N., Sharma, S.: The role of decision tree technique for automating intrusion detection system. In: International Journal of Computational Engineering Research, vol. 2(4) (2012)
19. Kang, M.J., Kang, J.W.: Intrusion detection system using deep neural network for in-vehicle network security. PLoS one **11**(6), e0155781 (2016)
20. Kneib, M., Huth, C.: Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks. In: 2018 ACM SIGSAC Conference on Computer and Communications Security, ACM, pp. 787–800 (2018)
21. Lee, H., Jeong, S.H., Kim, H.K.: OTIDS: a novel intrusion detection system for in-vehicle network by using remote frame. In: 2017 15th Annual Conference on Privacy, Security, and Trust (PST), IEEE (2017)
22. Marchetti, M., Stabili, D.: Anomaly detection of CAN bus messages through analysis of id sequences. In: 2017 IEEE Intelligent Vehicles Symposium (IV), IEEE, pp. 1577–1583 (2017)
23. Marchetti, M., Stabili, D., Guido, A., Colajanni, M.: Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms. In: 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow (RTSI), IEEE (2016)
24. Markovitz, M., Wool, A.: Field classification, modeling and anomaly detection in unknown can bus networks. Veh.Commun. **9**, 43–52 (2017)

25. MathWorks: MATLAB Statistics and Machine Learning Toolbox (2021)
26. Miller, C., Valasek, C.: Remote exploitation of an unaltered passenger vehicle. Black Hat USA (2015)
27. S. N. Narayanan, S. Mittal, and A. Joshi. Using data analytics to detect anomalous states in vehicles (2015) arXiv:1512.08048
28. Refat, R.U.D., Elkhail, A.A., Hafeez, A., Malik, H.: Detecting CAN bus intrusion by applying machine learning method to graph based features. In: Arai, K. (ed.) IntelliSys 2021. LNNS, vol. 296, pp. 730–748. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-82199-9_49
29. Song, H.M., Kim, H.R., Kim, H.K.: Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network. In: 2016 International Conference on Information Networking (ICOIN), IEEE, pp. 63–68 (2016)
30. Stabili, D., Marchetti, M., Colajanni, M.: Detecting attacks to internal vehicle networks through hamming distance. In: 2017 AEIT International Annual Conference, IEEE (2017)
31. Sugashima, T., Oka, D.K., Vuillaume, C.: Approaches for secure and efficient in-vehicle key management. In: SAE International Journal of Passenger Cars - Electronic and Electrical Systems, 9(2016-01-0070):100–106 (2016)
32. 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), IEEE, pp. 45–49 (2015)
33. 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), IEEE, pp. 130–139 (2016)
34. Tayyab, M., Hafeez, A., Malik, H.: Spoofing attack on clock based intrusion detection system in controller area networks. In: Proceedings of the NDIA Ground Vehicle Systems Engineering Technology Symp, pp. 1–13 (2018)
35. Ueda, H., Kurachi, R., Takada, H., Mizutani, T., Inoue, M., Horihata, S.: Security authentication system for in-vehicle network. SEI Tech. Rev. **81**, 5–9 (2015)
36. Wasicek, A.R., Pesé, M.D., Weimerskirch, A., Burakova, Y., Singh, K.: Context-aware intrusion detection in automotive control systems. In: ACM/IEEE 6th International Conference on Cyber-Physical Systems (ICCPS), IEEE, pp. 41–50 (2015)
37. Wolf, M., Weimerskirch, A., Paar, C.: Security in automotive bus systems. In: Workshop on Embedded Security in Cars (2004)
38. Wu, W., Huang, Y., Kurachi, R., Zeng, G., Xie, G., Li, R., Li, K.: Sliding window optimized information entropy analysis method for intrusion detection on in-vehicle networks. IEEE Access **6**, 45233–45245 (2018)
39. Zuo, M., Xie, S., Zhang, X., Yang, M.: Recognition of UAV video signal using RF fingerprints in the presence of Wifi interference. IEEE Access **9**, 88844–88851 (2021)