

A Frequency-based Data Mining Approach to Enhance in-vehicle Network Intrusion Detection

Joakim Rosell^{1*)} Cristofer Englund^{1,3)} Arash Vahidi²⁾ Nishat I Mowla¹⁾ Ana Magazinius¹⁾
Eric Järpe³⁾

1) Department of Mobility and Systems, 2) Department of Computer Science, Digital Systems, RISE Research Institutes of Sweden, Lindholmspiren 3A, SE-417 56 Gothenburg, Sweden

3) Center for Applied Intelligent Systems Research (CAISR), Halmstad University, 301 18 Halmstad, Sweden

*) Corresponding author's e-mail: joakim.rosell@ri.se

Received on M, D, YYYY

Modern vehicles have numerous electronic control units (ECUs) that constantly communicate over embedded in-vehicle networks (IVNs) comprised of controlled area network (CAN) segments. The simplicity and size-constrained 8-byte payload of the CAN bus technology makes it infeasible to integrate authenticity and integrity-based protection mechanisms. Thus, a malicious component will be able to inject malicious data into the network with minimal risk for detection. Such vulnerabilities have been demonstrated with various security attacks such as the flooding, fuzzing, and malfunction attacks. A practical approach to improve security in modern vehicles is to monitor the CAN bus traffic to detect anomalies. However, to administer such an intrusion detection system (IDS) with a general approach faces some challenges. First, the proprietary encodings of the CAN data fields need to be omitted as they are intellectual property of the original equipment manufacturers (OEMs) and differ across vehicle manufacturers and their models. Secondly, such general and practical IDS approach must also be computationally efficient in terms of speed and accuracy. Traditional IDSs for computer networks generally utilize a rule or signature-based approach. More recently, the approach of using machine learning (ML) with efficient feature representation has shown significant success because of faster detection and lower development and maintenance costs. Therefore, an efficient data aggregation technique with enhanced frequency-based feature representation to improve the performance of ML-based IDS for the IVNs is proposed. The performance gain was verified with the Survival Analysis Dataset for automobile IDS.

KEY WORDS: Data mining, in-vehicle network, intrusion detection, random forest, feature selection.

1 Introduction

A typical modern vehicle contains a large number of mechanical and electrical components such as actuators, sensors, and Electronic Control Units (ECUs) that communicate over a complex in-vehicle network (IVN). This network consists of multiple segments that utilize various communication technologies such as Ethernet, FlexRay and Controller Area Network (CAN) ⁽¹⁾.

The IVN has generally been seen as a safety-critical distributed system, although with the introduction of connected and autonomous vehicles security has also been raised as an important issue. Unfortunately, some parts of the IVN were never created with security in mind and are vulnerable to multiple security attacks ^{(2),(3)}.

An example that is commonly raised is the classic CAN bus, often used in many network segments in a vehicle. The CAN bus technology provides a simple serial bus for transmitting short messages with up to 8 bytes of payload as shown in Fig. 1. Unfortunately, the short message length leaves no room for secure authentication and integrity protection mechanisms. The security impact of this problem has been highlighted by many researchers ^{(4),(5),(6)}.

The CAN bus is highly exposed, either directly through physical access to the bus, the diagnostic port (OBD-II) or by first compromising another connected component such as the Infotainment system ^{(5),(7)}. Hence it is assumed that an attacker can

inject malicious CAN frames into the bus and/or disrupt delivery of legitimate frames. In 2010, Koscher et al. demonstrated a number of attacks that exploit this weakness, including spoofing attacks to impersonate selected ECUs and manipulate certain vehicle functions such as lights, degrade vehicle functionality via undocumented commands, and denial of service by flooding the bus ⁽⁸⁾. This was later repeated by other researchers which also demonstrated that such attacks could be performed both via the OBD-II port and remotely ^{(9),(10)}. Such attacks reinforce the belief that "most modern automobile systems have been designed with safety, and not security in mind" ⁽¹¹⁾.

While there have been some attempts to add some layer of security to CAN bus ⁽¹²⁾ or to use other bus technologies such as CAN-FD ⁽¹³⁾ insecure CAN segments are still common in in-vehicle networks. Hence the idea of monitoring the bus to at least detect security attacks has gained popularity.

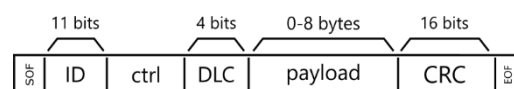


Fig. 1 Structure of a CAN frame.

1.1 Automotive intrusion detection

An intrusion detection system (IDS) monitors various system events in order to detect security issues. For example, a Host IDS (HIDS) monitors Operating System events such as file access

patterns to detect malicious activity while a Network IDS (NIDS) monitors network traffic to detect issues. Although a HIDS can certainly be used in vehicles (for example in the infotainment system or the network gateway), the focus of this work is use of a NIDS for detecting irregularities in the in-vehicle network.

An important property of an IDS is how detection is performed. A signature-based IDS utilizes a database of previously identified patterns while an anomaly-based IDS attempts to identify outlier events given a model of expected behavior. While signature-based IDS can be more accurate, it generally requires prior knowledge of attacks, information which currently is not available⁽¹⁴⁾. Hence recent research in automotive IDS has mostly focused on anomaly-based systems, in particular on using various machine learning and deep learning techniques⁽¹⁵⁾.

Another fundamental property of automotive IDS is where detection is performed. An IDS may operate solely within the vehicle or offload some or all work to a cloud environment. Hybrid solutions may perform preliminary data processing and analysis within the vehicle and the rest outside⁽¹⁴⁾.

1.2 Attacks and attack datasets

Normally some previously captured network traffic is required to construct and evaluate an anomaly-based NIDS. For automotive NIDS, and in particular for CAN networks, a number of public datasets exists with a range of different real and synthetic attacks⁽¹⁶⁾. Some common types of attacks observed in public datasets are:

1. **Flooding attacks**, where large amount of data is transmitted to halt or downgrade normal operation,
2. **Fuzzing attacks**, where the attacker transmits data with random sender and/or content to affect any functionality,
3. **Spoofing attacks**, where the attacker transmits data with falsified sender and/or content to affect some specific functionality.

For example, the attacker may spoof engine speed to cause a shutdown or flood a network segment to hinder brake messages from arriving and thus cause an accident.

Development an automotive IDS dataset involves some challenges. For example, recorded traffic may contain payload with proprietary encodings and unknown content. Furthermore, identifiers may differ across different manufacturers or even different models. In addition, some have chosen to obfuscate the dataset before publishing to protect trade secrets. Finally, the dataset may differ in the quantity and quality of the included data.

In this work the following datasets will be used: *Survival* and *GIDS* ("Car Hacking") datasets from HCRL⁽¹⁷⁾,⁽¹⁸⁾ and *TUEv2* dataset from Eindhoven University of Technology⁽¹⁹⁾. For a recent comparison of these and other public datasets refer to⁽¹⁶⁾.

1.3 Related work

Lee et al. consider time offset and interval between requests and responses to detect various attacks on the CAN bus⁽²⁰⁾. The authors also provide a public dataset with real spoofing, fuzzing and flooding attacks against a Kia Soul. Song et al. transform network metadata into bitmaps and use a deep learning image recognition model to identify attacks⁽²¹⁾. Seo et al. use a similar approach but use Generative Adversarial Network (GAN) with a shallower neural network⁽¹⁸⁾. Both use the same publicly available dataset which also includes various spoofing, fuzzing and flooding attacks. Olufowobi et al. identify injected frames using cumulative sum (CUSUM) change-point detection algorithm on the same dataset⁽²²⁾.

Hanselmann et al. combine autoencoders and recurrent neural networks for unsupervised learning with another public dataset⁽²³⁾. As use of such complex machine learning algorithms may exceed the vehicle resources, Loukas et al. consider a system where data gathering and aggregation happens in the vehicle, but more complex processing is done in the cloud⁽¹⁴⁾.

1.4 Contributions and outline

Proper representation and use of data are crucial for creating an efficient automotive intrusion detection system. In this paper a feature generation approach to facilitate and enhance the performance of anomaly-based IDSs is proposed. The contributions are:

- A proposed data-driven frequency-based aggregation mechanism to improve intrusion detection for the IVNs.
- Evaluation of the proposed method with different machine learning algorithms, different attacks and different datasets.
- Three different algorithms for anomaly detection are considered: random forest (RF), support vector machines (SVM) and linear models (LM).

The datasets, algorithms, and aggregation methods are described in Section 2. The IDS performance is evaluated in Section 3 and the results are discussed in Section 4.

2 Methodology

The intention of this work is to investigate a frequency-based (instead of a signal-based) data mining approach, and hence omit the issue of obfuscated CAN messages. Since the proprietary encodings of the CAN signals are not being disclosed by the original authors and may differ across different manufacturers and models, the monitored CAN ID and payload will be ignored.

To motivate this approach, let us take a look at the *Survival* dataset. The provided data includes three different real (not simulated) classes of attacks and one attack free class depicting normal driving on multiple vehicles. The attacks, which are thoroughly described in⁽¹⁶⁾ and⁽¹⁷⁾, last 25-100 seconds:

1. *Flooding attacks* by injecting messages at high frequency with the CAN ID 0x000, which is the highest priority CAN frame.
2. *Fuzzing attacks* by injecting arbitrary CAN frames with random ID and payload every 0.0003 seconds. Such messages may trigger unexpected behavior in the vehicle.
3. *Malfunction attacks* by spoofing messages from 3 different senders in HYUNDAI YF Sonata, KIA Soul, and CHEVROLET Spark vehicles.

It is trivial to use this dataset to build a signature or anomaly-based IDS with excellent theoretical performance: the ID field alone is a good indicator of whether a frame is normal or malicious. Unfortunately, this will most likely not translate to good real-world performance for several reasons. For one, the ID and payload fields may have been obfuscated before a dataset is published. Furthermore, the IDs can differ across manufacturers or even across models. Finally, the adversary may try to avoid detection by making minor adjustments to the ID field. For example, she may perform a flooding attack using high priority IDs other than 0x000.

2.1 Data pre-processing

The underlying mechanism of the frequency-based data aggregation technique, as shown in Fig. 2, omits the obfuscated CAN data fields within the CAN bus protocol and enables a generic, computationally efficient IDS with a frequency-based approach. However, certain pre-processing of the CAN bus dataset has been performed to meet such requirements.

The features included in the given CAN bus protocols provided are:

- Timestamp, t.
- CAN-frame identifier, CAN ID.
- Data length code, DLC.
- Payload.

And, through some pre-processing of these features, two more features were added to the CAN-bus protocols:

- Time difference between timestamps, dt .
- The time difference between each individual CAN-frame identifier, $dt_{ID1, \dots, n}$. Where first appearance is set to $t=0$.

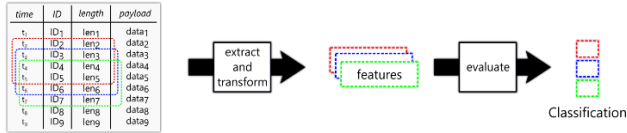


Fig. 2 Aggregation technique. A sliding window technique is adopted to aggregate the features analyzed. The length, i.e., the number of CAN-messages included within the sliding window, can distinctively be varied for each model.

As mentioned, information regarding the obfuscated CAN data fields should be omitted, hence the payload feature and the CAN ID feature were ignored when a sliding window technique, variable in length, was adapted to the pre-processed dataset and traversed through it. The length of the consecutive rows of features that each window generates are thus based on the length of the window, n . Every next consecutive row of features will be shifted by one element as the window has moved to the next consecutive CAN-frame of the dataset. Furthermore, three more features based on the aggregated data from the sliding window technique were calculated and included to the considered list of features as shown in Table 1.

- The time duration from the first frame (t set to 0) to the last frame in the window, win_dur .
- The mean time difference between the CAN-frames in the sliding window, win_mean_dt .
- The number of occurrences of the most frequent CAN ID per window, $win_most_freq_id$.

The same sliding window technique was applied on the attacked data, where a window was classified as attacked if it contained one or more attacked frames. However, the analyzed CAN datasets only consist of one to four attack injections making the number of transitions from attack-free frames to attacked frames limited. Hence, an initial attempt of generating many such transitions, i.e., going from attack-free frames to attacked frames, such segments were truncated from the generated feature vectors and used in the training of the models. The process of generating such transitions is easily explained through:

- Locate first consecutive attacked CAN-frame
- Slide the CAN-frames from the located attacked frame in i), bottom to top, through the window
- Remove the ii) CAN-frames from the dataset and repeat the process.

3 Experiments

3.1 Feature selection

Since computational speed and low model complexity are two prominent features of RF compared to other methods, the RF was initially used to model the self-assessed distraction level and to select the most descriptive variables before the modelling performance of Random Forest (RF) ⁽²⁴⁾, Support Vector Machine (SVM) ⁽²⁴⁾, and Linear Model (LM) ⁽²⁵⁾ were compared. The number of trees of the RF throughout all experiments were set to 20, and all results are achieved from a 5-fold cross validation taking 80 % of the data for training and 20 % for testing. Also, the sequential order of the feature vectors was randomly permuted, and all features were normalized through z-score normalization.

The feature selection method used was based on backward elimination and is summarized below.

1. Build M models and for each m , remove the m^{th} variable.
2. Use the validation/OOB dataset to estimate the performance of each model m .
3. The model m with the lowest error indicates the variable m that influences the model the least and will be removed.
4. A score is given to the removed variable, indicating in what order it was removed.
5. Remove the variable and restart from 1 with $M = M - 1$ variables until $M = 1$.

Table 1. Features used. Note that no information related to the obfuscated CAN data fields are present.

| # | Features meaning | Features |
|---|--------------------------------------------------------------|---------------------------------|
| 1 | Time difference between timestamps. | dt_1, \dots, dt_n [s] |
| 2 | DLC | DLC_1, \dots, DLC_n [bytes] |
| 3 | The time difference between each individual CAN ID. | $dt_{ID1}, \dots, dt_{IDn}$ [s] |
| 4 | Time length of window | win_dur [s] |
| 5 | Mean time difference between messages in window | win_mean_dt [s] |
| 6 | Number of occurrences of the most frequent message in window | $win_most_freq_id$ [a.u.] |

This method was applied on the *Survival* CAN-bus protocol containing Flooding attacks. A sliding window of size 10, and two thousand attack-free frames as well as two thousand attacked frames were used. As the length of the feature vectors generated by the sliding window depends on the size of the sliding window, the number of features for $n = 10$ is 33.

3.2 Performance Metrics

As the main purpose of the proposed approach is to classify whether the feature vectors of the CAN-bus traffic are attacked or attack-free, the obtained results are evaluated using different performance metrics. The performance metrics used for comparison are briefly described below.

$$F1\text{-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Error rate} = \frac{FN + FP}{TP + TN + FP + FN} = 1 - \text{Accuracy}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Matthews correlation} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}$$

Table 2 represents these performance metrics, and the model performances (F1-score) on the different types of attacks are shown in Fig. 7.

3.3 Modelling Performance

The results from the initial procedure to determine which features matters the most for the RF model were rather distinctive, where the features of row three in Table 1 matters the most for the models, i.e., “The time difference between each individual CAN ID, $dt_{ID1, \dots, n}$ ”. This might not be remarkable, as the time difference between the attacked frames in the Flooding-attack are intentionally set to be shorter, compared to attack-free CAN-bus protocols, while trying to flood the network. In order to investigate whether the models still could perform without this dominant feature, the experiments were initially run without the features of $dt_{ID1, \dots, n}$ while generating the models. The features of Table 1, except feature #3, were then used for the three different models

compared, where different lengths of the sliding window also were investigated and shown in Table 2.

Notable high performance of the models for shorter sliding windows are seen in Table 2. As this was unexpected some consideration regarding the different features used needs to be elaborated.

Table 2. Performance metrics of the different models applied on the CAN-bus protocol containing flooding attacks from the KIA Sonata car from the Survival dataset. Best performance metrics per length of window is marked in bold. The features analyzed are: dt, DLC, win_dur, win_mean_dt and win_most_freq_id (no. 1, 2, 4, 5 and 6 of Table 1).

The standard error of the mean is less than 0.01.

| Model | Win size | F1-score | Prec | Recall | Err rate | Acc | Mathews corr |
|-------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| LM | 15 | 0,881 | 1,000 | 0,788 | 0,101 | 0,899 | 0,815 |
| | 10 | 0,867 | 1,000 | 0,766 | 0,106 | 0,893 | 0,805 |
| | 7 | 0,835 | 0,999 | 0,717 | 0,128 | 0,871 | 0,766 |
| | 5 | 0,726 | 0,990 | 0,573 | 0,197 | 0,799 | 0,647 |
| | 3 | 0,570 | 0,957 | 0,407 | 0,305 | 0,695 | 0,475 |
| | 2 | 0,699 | 0,571 | 0,902 | 0,388 | 0,611 | 0,274 |
| 1 | 0,613 | 0,506 | 0,780 | 0,491 | 0,509 | 0,023 | |
| SVM | 15 | 0,996 | 1,000 | 0,992 | 0,004 | 0,996 | 0,991 |
| | 10 | 0,976 | 0,993 | 0,959 | 0,024 | 0,976 | 0,953 |
| | 7 | 0,932 | 0,992 | 0,878 | 0,064 | 0,936 | 0,877 |
| | 5 | 0,853 | 0,966 | 0,764 | 0,131 | 0,869 | 0,754 |
| | 3 | 0,890 | 0,909 | 0,871 | 0,108 | 0,892 | 0,785 |
| | 2 | 0,913 | 0,942 | 0,887 | 0,084 | 0,916 | 0,833 |
| 1 | 0,937 | 0,965 | 0,910 | 0,062 | 0,939 | 0,878 | |
| RF | 15 | 0,991 | 0,995 | 0,987 | 0,009 | 0,991 | 0,983 |
| | 10 | 0,991 | 0,994 | 0,988 | 0,009 | 0,991 | 0,983 |
| | 7 | 0,980 | 0,987 | 0,973 | 0,020 | 0,980 | 0,961 |
| | 5 | 0,977 | 0,980 | 0,975 | 0,023 | 0,978 | 0,955 |
| | 3 | 0,968 | 0,973 | 0,964 | 0,031 | 0,969 | 0,938 |
| | 2 | 0,948 | 0,946 | 0,951 | 0,052 | 0,948 | 0,896 |
| 1 | 0,943 | 0,951 | 0,936 | 0,057 | 0,944 | 0,887 | |

Looking at the features wind_dur (#4 in Table 1) and win_mean_dt (# 5 in Table 1) for shorter window sizes. For example, when the window size, n , is set to one ($n=1$), i.e., no window, the features of wind_dur (#4 in Table 1) are zero. And the feature of win_mean_dt (# 5 in Table 1), while $n=1$, are equal to dt₁ (# 1 in Table 1). Further, the feature win_most_freq_id (#6 in Table 1) was considered to inadvertently affect the models differently for shorter windows compared to longer. For example, win_most_freq_id is always one for $n=1$. And for $n=2$ and $n=3$ win_most_freq_id could occasionally be considered as “label leakage”, i.e., indicating when a window is labeled as attacked, while the values are larger than one. For $n>5$, win_most_freq_id >1 will still indicate an attack, but values between 1 and n are more distributed for larger n . This could explain the dip in performance for the SVM model (and LM) in Fig. 3 where the experiment was ran again also omitting this win_most_freq_id feature. Also, in Fig. 3 are the results from the training data included in gray. As will be pointed out later in Fig. 7, there is no significant change in the results of omitting the win_most_freq_id feature (blue lines compared to black lines) for the RF and SVM model.

Shown in Fig. 3b, the false negative as well as the false positive of the RF method are shown for varying length of the sliding window when the analyzed features are as in Fig. 3a., i.e., dt, DLC, win_dur and win_mean_dt (no. 1, 2, 4 and 5 of Table 1). Both the false negative and the false positive results of the RF model clearly decrease with longer sliding windows.

The exact same method, after the feature selection procedure, was adapted on the other Survival datasets including the so-called

fuzzing attacks and malfunction attacks. However, for comparison between adapting the generated attack transitions mentioned in 2.1 Data pre-processing, and not adapting such transitions by just sliding the window over the whole data set (including the attacks), the F1-score of the different models with varying window size is shown in Fig. 7. Hence, Fig. 3a and Fig. 7a (black lines) could be used for comparison over the approach to the attacked data. In Fig. 7, these results are combined with the results from analyzing the five features used in Table 2 (blue lines) as well as the results of analyzing all features in Table 1 (red lines). Commonly for Fig. 7 is that the F1-score of the models per length of the sliding window performed much poorer for smaller n . Most affected was the LM.

Additionally, the same method was adapted on the TUEv2 dataset. This dataset was chosen as it contains CAN-bus data with generated Flooding attacks from a different vehicle manufacturer than the other dataset analyzed, and the models’ performances are shown in Fig. 7d. Here, the feature win_most_freq_id (#4 in Table 1) will definitely act as label leakage since the Flooding attacks of the TUEv2 dataset do not interweave normal traffic in their attack scenario. Hence, the win_most_freq_id is either 1 or n (n being the length of sliding window), generating F1-score of 1.0 for all n , except for $n=1$ (blue lines).

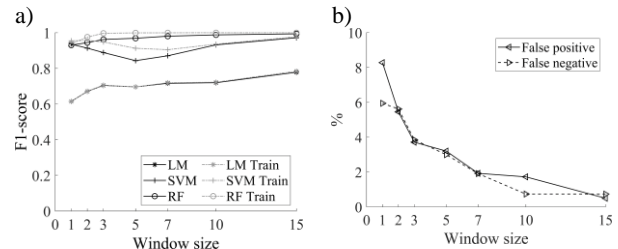


Fig. 3 a) F1-score for the different models with varying size of the sliding window. b) Performance metrics of false negative and false positive over the RF method with varying size of the sliding window. Features analyzed are: dt, DLC, win_dur and win_mean_dt (no. 1, 2, 4 and 5 of Table 1).

3.4 Real-time Efficiency

An important characteristic of an automotive IDS is the response latency. If the amount of time between intrusion and detection is high (e.g., hours), then the IDS output cannot be used by real-time intrusion prevention mechanisms.

To investigate this, two main properties are considered: how much data past the intrusion point is required to make an assessment, and whether the solution can operate within the constraints of a vehicle. For the sake of simplicity, only the RF algorithm is considered in these experiments.

3.4.1 Response latency

The response latency is the amount of time between attack and detection. However, since this depends on both hardware and traffic characteristics (including network speed and load), a simpler experiment is considered: in a window of N frames, the N^{th} element represents the last received frame, $N - 1^{\text{th}}$ element represents the previous frame and so on. A good detection latency should translate to detecting more attacks closer to arrival (i.e., closer to entry N).

Using a window of 10 frames and a forest of size 10, this was tested on the entire GIDS dataset. As seen in Fig. 4, a significant number of attacks are detected within the first 2-3 frames.

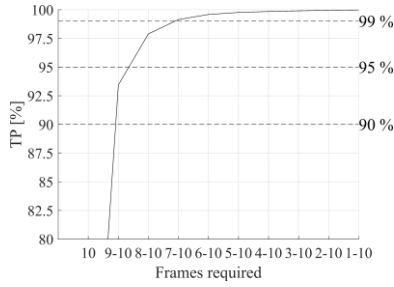


Fig. 4 Detection latency within a window of 10 frames ($N=10$).

3.4.2 Resource usage

To achieve low detection latency, it might also be relevant to execute the IDS entirely within the vehicle. Unfortunately, in comparison to the cloud, the vehicle may have very limited resources allocated to security functionality.

For example, to compute dt_ID one needs to keep track of previous frame from each sender. If the memory to do this is limited some information may be forgotten and the data aggregation may occasionally yield incorrect values which can in turn affect the IDS performance. This is illustrated in Fig. 5 for some attacks in the *Survival* dataset.

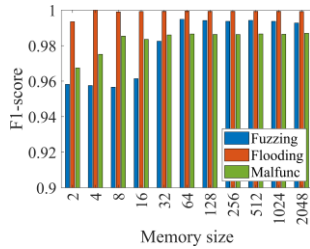


Fig. 5 Effect of aggregation with limited memory.

Another issue to consider is the complexity of the random forest. The effect of forest size (number of trees) and tree complexity (by maximum tree depth) on the same dataset is seen in Fig. 6.

4 Results and discussion

As noted in Table 2, the random forest method outperforms both linear model and SVM on all performance metrics. As illustrated in Fig. 7, this behavior is constantly repeated for different types of attacks and different datasets.

It can also be seen that increasing the window size quickly improves performance although a window of 10 CAN frames seems to provide a reasonably good performance. This is clearly visible in Fig. 3 and Fig. 7 although it is hard to draw any conclusions without looking at more datasets and attack variations.

As attacks are implemented differently in different datasets, interesting patterns in the data can sometimes be observe. For example, compared to the *Survival* dataset the *TUEv2* dataset seems to inject frames at a higher frequency. Even when using shorter windows, this creates a strong correlation between the $win_most_freq_id$ and the $dt_ID_{1,...,n}$ features and the IDS output. The effects of omitting these features have been investigated during each attack, and as seen in Fig. 7 the performance is still acceptable.

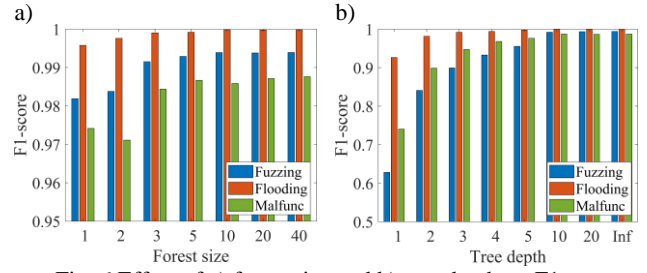


Fig. 6 Effect of a) forest size and b) tree depth on F1-score.

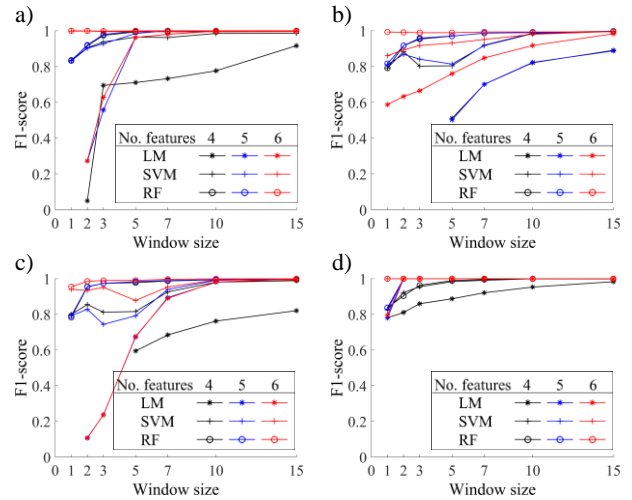


Fig. 7 F1-score for the different models with varying size of the sliding window: *Survival* flooding (a), fuzzing (b) and malfunction attacks (c) in addition to the *TUEv2* Flooding attacks (d).

In red: 6 features analyzed, i.e., all features of Table 1.
 In blue: 5 features analyzed, omitting $win_most_freq_id$.
 In black: 4 features analyzed, omitting $win_most_freq_id$ and $dt_ID_{1,...,n}$.

Beside performance, computation efficiency is another important property that has been investigated. As observed in Fig. 4, the proposed method manages to detect most attacks early in their window, often after receiving only 2-3 CAN frames. While this is not a conclusive analysis of the IDS latency, it is believed that this is a good indicator of the expected real-world performance.

In situations where intrusion detection is performed in the vehicle, the footprint of the detection method will be of some importance. For example, a very lightweight IDS could be placed anywhere in the vehicle while other IDSs would be limited to high-end components such as the network gateway or even require cloud assistance.

As illustrated in Fig. 5, memory constraints during data aggregation can affect IDS performance. However, it was observed that aggregation memory could be significantly reduced (from 2^{11} to 2^5 entries) without any major loss in performance.

The complexity of the method used in the IDS can also affect both CPU and memory usage. A very simple method may be CPU and memory efficient but result in worse IDS performance while a very complex method may not be able to run at real-time in the vehicle. To investigate this, the performance of the RF method with different levels of forest and tree complexity was analyzed. As illustrated in Fig. 6, very simple configurations result in somewhat reduced performance. However, even with modest complexity the performance improves very rapidly. For the analyzed dataset and attacks, it seems 5-10 trees of depth 5-10 nodes perform reasonably well.

5 Conclusion

This paper proposes an efficient data aggregation mechanism for use in automotive intrusion detection systems. In particular, the suggested method is able to extract meaningful frequency information from a window of CAN messages. To demonstrate the effectiveness of this approach, multiple datasets from various sources and three different attacks: flooding, fuzzing and spoofing, have been apply to it. According to a wide range of performance metrics, the proposed method successfully detects such attacks given relatively short windows of data. Further experiments demonstrate the algorithm performance with linear models, SVM and random forest. It is demonstrated that the random forest method yields the best result and continues to perform well even as the forest complexity is reduced for use in a resource-constrained environment. In the future, we aim to evaluate the performance of the proposed solution of this paper with multiple other publicly available CAN bus datasets and a wider range of fabrication, suspension, and masquerade attacks. The goal is to assess the quality of generalization of our proposed technique. Furthermore, we also aim to experiment with more suitable and robust machine learning models that elevates the proposed technique.

6 Acknowledgments

This work was supported by Swedish Governmental Agency for Innovation Systems (Vinnova) through the CyReV project under the agreements 2018-05013 and 2019-03071.

References

- (1) W. Zeng, M. A. Khalid and S. Chowdhury, "In-vehicle networks outlook: Achievements and challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1552-1571, 2016.
- (2) U. E. Larson and D. K. Nilsson, "Securing vehicles against cyber attacks," in *Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead*, 2008.
- (3) T. Rosenstatter, "A State-of-the-Art Report on Vehicular Security," 2017. [Online]. Available: <https://autosec.se/wp-content/uploads/2018/04/1.2-holisec-state-of-the-art.pdf>.
- (4) T. Hoppe, S. Kiltz and J. Dittmann, "Security threats to automotive CAN networks—practical examples and selected short-term countermeasures," in *International Conference on Computer Safety, Reliability, and Security*, Berlin, Heidelberg, 2008.
- (5) M. Wolf, A. Weimerskirch and C. Paar, "Security in automotive bus systems," in *Workshop on Embedded Security in Cars*, 2004.
- (6) P. Carsten, T. R. Andel, M. Yampolskiy and J. T. McDonald, "In-vehicle networks: Attacks, vulnerabilities, and proposed solutions," in *Proceedings of the 10th Annual Cyber and Information Security Research Conference*, 2015.
- (7) S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *USENIX Security Symposium*, 2011.
- (8) K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham and S. and Savage, *Experimental Security Analysis of a Modern Automobile in The Ethics of Information Technology*, Routledge, 2020, pp. 119-134.
- (9) C. Miller and C. Valasek, "Adventures in automotive networks and control units," in *Def Con 21*, 2013.
- (10) A. Greenberg, "WIRED," [Online]. Available: <https://www.wired.com/2016/08/jeep-hackers-return-high-speed-steering-acceleration-hacks/>.
- (11) S. F. Lokman, A. T. Othman and M. H. Abu-Bakar, "Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1-17, 2019.
- (12) D. K. Nilsson, U. E. Larson and E. Jonsson, "Efficient in-vehicle delayed data authentication based on compound message authentication codes," in *2008 IEEE 68th Vehicular Technology Conference*, 2008.
- (13) S. Woo, H. J. Jo, I. S. Kim and D. H. Lee, "A practical security architecture for in-vehicle CAN-FD," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2248-2261, 2016.
- (14) G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon and D. Gan, "Cloud-based cyber-physical intrusion detection for vehicles using deep learning," *IEEE Access*, vol. 6, pp. 3491-3508, 2017.
- (15) O. Y. Al-Jarrah, C. Maple, M. Dianati, D. Oxtoby and A. Mouzakitis, "Intrusion detection systems for intra-vehicle networks: A review," *IEEE Access*, vol. 7, pp. 21266-21289, 2019.
- (16) M. E. Verma, M. D. Iannacone, R. A. Bridges, S. C. Hollifield, B. Kay and F. L. Combs, "ROAD: The Real ORNL Automotive Dynamometer Controller Area Network Intrusion Detection Dataset (with a comprehensive CAN IDS dataset survey & guide)," *arXiv:2012.14600*, 2020.
- (17) M. L. Han, B. I. Kwak and H. K. Kim, "Anomaly intrusion detection method for vehicular networks based on survival analysis," *Vehicular communications*, vol. 14, pp. 52-63, 2018.
- (18) 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.
- (19) G. Dupont, A. Lekidis, J. den Hartog and S. Etalle, "Automotive Controller Area Network (CAN) Bus Intrusion Dataset v2. 4TU.ResearchData,," 2019. [Online]. Available: <https://doi.org/10.4121/uuid:b74b4928-c377-4585-9432-2004dfa20a5d>.
- (20) H. Lee, S. H. Jeong and H. K. Kim, "OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame," in *In 2017 15th Annual Conference on Privacy, Security and Trust (PST)*, 2017.
- (21) 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.
- (22) H. Olufowobi, U. Ezeobi, E. Muhati, G. Robinson, C. Young, J. Zambreno and G. Bloom, "Anomaly detection approach using adaptive cumulative sum algorithm for controller area network," in *Proceedings of the ACM Workshop on Automotive Cybersecurity*, 2019.
- (23) M. Hanselmann, T. Strauss, K. Dormann and H. Ulmer, "CANet: An unsupervised intrusion detection system for high dimensional CAN bus data," *IEEE Access*, vol. 8, pp. 58194-58205, 2020.
- (24) V. N. Vapnik, "An overview of statistical learning theory,," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988-999, 1999.
- (25) J. Friedman, T. Hastie and R. Tibshirani, *The elements of statistical learning*, vol. 1(10), New York: Springer series in statistics, 2001.