

Dynamic Voting based Explainable Intrusion Detection System for In-vehicle Network

Nishat I Mowla*, Joakim Rosell*, Arash Vahidi**

*Department of Mobility and Systems, **Department of Computer Science

RISE Research Institutes of Sweden

Sweden

nishat.mowla@ri.se, joakim.rosell@ri.se, arash.vahidi@ri.se

Abstract—A modern vehicle contains a large number of electronic components communicating over a large in-vehicle network. While the operation of this network is crucial, some implementations are vulnerable to a number of security attacks while lacking sufficient security measures. Intrusion detection systems have been proposed as a possible solution to this, with those using machine learning receiving much attention. However, such systems may be hard to interpret and understand. In this work, we propose an automotive intrusion detection system that utilizes Random Forest with a dynamic voting technique to provide a robust solution with interpretability through feature and model exploration. The proposed solution is evaluated using two publicly available datasets and demonstrates stable performance when compared to similar solutions.

Keywords—In-vehicle network, intrusion detection, random forest, ensemble learning, explainable AI

I. INTRODUCTION

It has been estimated that by 2025, 60% of all vehicles will be internet connected [1]. Modern vehicles will also contain a large number of electronic control units (ECUs) communicating over an in-vehicle network (IVN) and external communication interfaces to provide intelligent services and functionalities [2]. It is believed that the increasing connectivity and complexity will result in increased security risks and new attack vectors [3]. For example, an attacker might exploit a vulnerability in the infotainment systems or a communication interface, then access IVN to affect core vehicle functions or passenger safety. A number of real-world security attacks against vehicles have already been demonstrated. For example, researchers were able to gain access to Jeep Cherokee via its Wi-Fi interface, and may have been able to modify ECU firmware [4].

Some critical segments of the in-vehicle network use the CAN (Controlled Area Network) technology. This popular bus standard normally only allows transmission of 8 bytes of payload per message, which leaves little room for any security functionality such as message authentication codes [5]. Furthermore, firewalls may not work well in these segments due to the arrangement of the IVN and the external interfaces. Migrating to more secure bus technologies may not be feasible for many years, hence as a short term solution, the use of

This research was supported by the Swedish VINNOVA FFI project "CyReV: Cyber Resilience for Vehicles". Nishat I Mowla is the corresponding author.

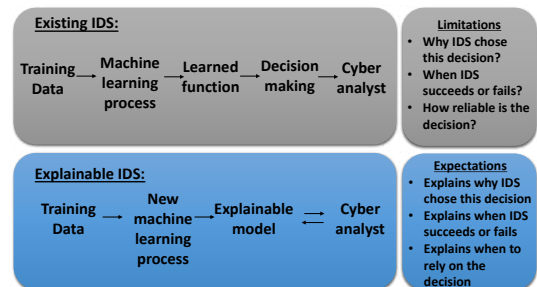


Figure 1: Existing IDS and explainable IDS.

intrusion detection system (IDS) have been proposed. An IDS could be implemented inside the vehicle, in a cloud back-end or as a hybrid solution [6]. An in-vehicle IDS could detect abnormal activity quicker, although the implementation needs to fit the constrained vehicle environment.

Signature-based IDS solutions use a database of patterns to recognize malicious activity, which may not work well to detect new threats. Anomaly-based IDS solutions instead try to learn a system's normal behaviour and recognize activity that deviates from that. Advances in machine learning has increased the popularity of such solutions, for example by using random forest (RF), artificial neural networks, one-class support vector machine (SVM), ensemble learning and clustering techniques [7], [8].

Random forest seems to perform very well in this context, but may be difficult to interpret and explain [9], [10]. Unfortunately, explainability is an important property when using artificial intelligence for making decisions [11], [12]. As shown in Fig. 1, an explainable IDS must be able to answer questions related to decisions it makes, its success and failure and when it can be trusted. Although there are still no standard guidelines for explainable AI solutions [13], tools such as SHAP [14], LIME [15], IBM's AIX360 [16] may improve explainability by highlighting importance of sample properties.

Given that, the focus of this paper is to find a reliable IDS solution for in-vehicle networks that considers both model robustness and interpretability through exploration. The main contributions of this paper are as follows:

- 1) We propose a reliable CAN IDS inspired by the concept of "*explanation as exploration*" for addressing explainable AI goals.
- 2) Using *two publicly available standard datasets*, we perform an extensive experimental analysis to evaluate and assess the model's capacity to *generalize* to different data.
- 3) We evaluate the interpretability of the proposed solution and further use it to show that reasonable performance can be achieved with a subset of the available features and shorter training time.

The remaining of this document is organized as follows: related and previous work is discussed in Section II, the proposed mechanisms is described in Section III and evaluated in Section IV while Section V contains our conclusions

II. RELATED WORKS

In this section we consider previous work related to in-vehicle network intrusion detection systems followed by a closer look at machine learning-based intrusion detection systems.

A. Intrusion detection system for in-vehicle networks

An automotive IDS can monitor a wide range of input types to detect issues. Examples include message payload [17], interpreted payload content [18], timing information [19] and physical attributes [20].

The IDS itself generally belongs to one of two classes: a *rule/signature-based* system where the structure of valid/invalid messages are specified in a database, or an *anomaly-based* system where a mathematical model of normal behaviour is first constructed and later used to detect anomalous activities [21]. Rule-based solutions normally require expert knowledge to create the database, which must also be constantly updated as new attacks are discovered or as new devices or functionality is added to the vehicle. Anomaly-based solutions on the other hand rely mainly on the availability of data, which possibly makes them more suitable for use in automotive solutions.

B. Machine learning-based intrusion detection systems

Many recent IDS implementations employ machine learning for anomaly detection. Interesting machine learning techniques include various types of neural networks such as recurrent and reconstructing networks [22], [23], one-class support vector machines [24] and hidden Markov models [25]. Random Forest (RF) is another interesting technique and is an *ensemble method* that utilizes voting among a group of Decision Trees (hence the name) [26]. RF is less sensitive to some common machine learning problems such as overfitting and generally performs well in intrusion detection tasks [27]. Performance can sometimes be improved by increasing forest size and complexity, although this comes with an increased computation cost. It may be possible to find the optimum parameters by using a hyperparameter optimization method, such as grid search as demonstrated by Primartha *et al.* [28].

Algorithm 1 Dynamic voting with ensemble of random forest for feature and model exploration

Step 1: Initialization

Initialize models m : $RF_{q1}, RF_{q2}, RF_{q3}, RF_{grid}$

Initialize control model parameters: $q1, q2, q3$

where $q1 < q2 < q3$

Initialize $grid = [q1, q3]$

Initialize weights for each model m , $w_m = 0$

Pre-process $data(x_i, y_i)$

Aggregate (x_i, y_i) with windows

Extract feature set f from x_i

Step 2: Training participating models

for each model m from 1 to M do

for each feature set f do

 Model m fitting with initialized parameters

$score(m, x_i, y_i) = \text{Training score with } f$

Step 3: Weight generation

for each model m from 1 to M do

$w_m \leftarrow score(m, x_i, y_i)$

return w

Step 4: Dynamic voting for ensemble generation

for each feature set f do

 Dynamic soft voting with w

$\hat{y} = \arg \max_c \sum_{m=1}^M w_m p_{cm}$

For AI to reach a wider acceptance, it is important that its decision making process can be explained. This has already been studied in fields such as computer vision, natural language processing and biology. In the field of security, Wang *et al.* proposed a framework to explain why decision are made and what features they use for intrusion detection. In [29], the explainable AI concept is leveraged to enhance trust management by exploring the decision tree model for intrusion detection. Some recent work focuses on explainable AI to make intrusion detection decisions more interpretable [13] while most of the previous works on model interpretation focuses on other fields like computer vision, natural language processing, and biology. The lack of research in this direction makes it difficult for the cybersecurity experts to optimize their decisions according to the judgments of the model in practice.

III. PROPOSED MECHANISM

The proposed intrusion detection model has three main parts: a) data pre-process, b) explanation through exploration and training, c) model testing.

A. Data pre-process

In the CAN bus network, most of the normal messages arrive at a strict frequency and the majority of attacks means inserting extra messages which causes changes in the frequency patterns [30] [31]. This motivates our pre-processing to be a frequency-based approach. The detection model in the micro-controller retrieves CAN messages from the logged

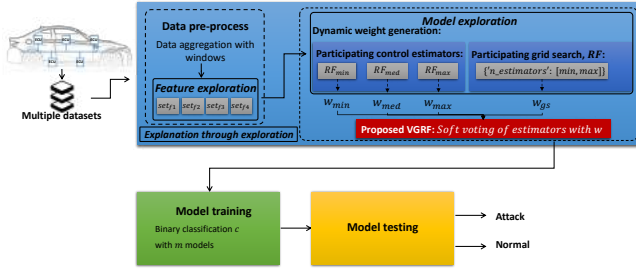


Figure 2: Proposed mechanism of dynamic voting for explanation through exploration.

CAN traffic dump. The adopted aggregation technique mentioned uses a sliding window technique that is run through the pre-processed features acquired. As the sliding window is variable in size, n , i.e., how many CAN messages are included in the window, the length of every consecutive row of features per window, also equals 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. Based on this generated aggregation, three more features were calculated, being: time duration from first frame (t set to 0) to the last frame in the window, win_dur , mean time difference between the CAN-frames in the sliding window, win_mean_dt , and number of occurrences of the most frequent CAN ID per window, $win_most_freq_id$. The time difference between timestamps, dt , and the time difference between each individual CAN ID, dt_ID are taken as the time-based features. A predefined message window that contains one or more attack messages are labeled as *attack* or *normal*, otherwise. The same feature format is adopted in both the datasets to establish a generalizability of the proposed solution.

B. Explanation through exploration and training

The quality of an explanation or interpretation of a machine learning-based detection system needs to be evaluated in an application-specific manner. It is equally important to understand the assumptions and limitations of the explanation/interpretation technique. There are still no general guidelines for explainable AI solutions [11] [13]. Among other techniques involving trust, mental modelling, human-in-the-loop, exploration-based solutions are useful to find interpretations and explanations [12] of certain decisions over others, given different choices, made by the detection model. Exploration is done through interaction and drill-down, among other things to get structure explanations. The model proposed in this paper leads to improve the transparency of intrusion detection which is also helpful for a cybersecurity analyst to have a better understanding of the detection model's decision. The proposed model approaches the explanation and interpretation through exploring feature contribution and model contribution. The **feature exploration** tries to find the most important features contributing to the decision by increasing the prediction probabilities in the right direction. This can be evaluated by finding out which set of features are increasing the overall accuracy or other performance metrics of the model. For the proposed

model, a univariate feature selection mechanism is followed to generate selected and ranked features with tailored intervals dividing the total number of features into four sections. The univariate feature selection expedites the selection of the most important features based on univariate statistical tests. As a result, the univariate feature selection enables strong statistical tests that can be used to select those features that are most important and have the strongest relationship with the output variable [32]. The **model exploration** is done through setting up three control models and evaluating the final model's decision based on the contribution of each of the three control models. The contributions from each control model is represented in the form of a vote to the final model. In particular, a set of participating random forest estimators are initialized as shown in Fig. 2. The proposed mechanism is divided into four main steps to design the proposed model, VGRF. In the first step, the participating models, parameters, and data are initialized as shown in Algorithm 1. In particular, three control random forest models with parameters $q1$, $q2$, $q3$ representing the number of trees are initialized where $q1 < q2 < q3$. A grid search model of random forest RF_{grid} is initialized with the range of estimators set to $[q1, q3]$. The weights w for each model m is set to 0. The second step is the training of the participating models, i.e., RF_{q1} , RF_{q2} , RF_{q3} , RF_{grid} . After the data pre-processing as described in the previous subsection, each participating model m is trained with each feature set f with their initialized parameters. The training scores achieved with feature set f are saved for generating the weights in the next step. In the third step, a weight w_m is set to each model m which is the score of model m from the previous step. The weights are returned as a list for the dynamic voting in the next step. In the fourth step, a dynamic soft voting is performed to generate the new ensemble of VGRF where the weight of each participating model m is dynamically adjusted based on their performing score during training. The sequence of weights from all the participating models m provide weight on the class probabilities p_{cm} for class c and model m before averaging. The soft voting predicts the class label $\hat{y}$ based on the argmax of the sums of the predicted probabilities weighted by the dynamic weights of the participating models. Accordingly, a binary classification problem is solved by the model training for the two classes, i.e., attack and normal. The exploration is done in two dimensions with feature exploration and control model exploration. For each feature set f , the performance of each control model is evaluated and their performance contributes as the dynamic weights which train the final model. This allows to understand which features and which control models are positively contributing to improving the detection performance. Therefore, the change in performance of the proposed model can be inferred from the participating feature set and control models as will be numerically shown in the performance evaluation section.

C. Model testing

After the model is trained with all the different combinations of the random forest models created, the model is tested with the best performing feature set f . The comparative models include the individual control random forest models, the grid search model, a soft voting model without the grid search and the proposed soft voting model with all the participating models. All the models are then tested on the test data of each of the datasets to predict the outcome of the binary class problem.

IV. PERFORMANCE EVALUATION

A. Datasets

For the performance evaluation, we have tested our proposed model on the following standard in-vehicle network datasets:

- *Survival Analysis Dataset (SAD)*: The survival analysis data includes CAN-bus data from three different vehicles, being; Hyundai Sonata, Kia Soul and Chevrolet Spark. Normal driving data and three different injection attacks for each vehicle are included. The attack packets were injected for five seconds every 20 seconds for the three attack scenarios where the attacks do have a real effect on the vehicle they are injected to. A more thorough description of the SAD is found in [33] and [34]. The following three attack scenarios: Flooding, Fuzzy, and Malfunction were generated that can severely impair in-vehicle functions or deepen the intensity of an attack and the degree of damage. This dataset, however, has a fewer number of attacks.
- *ORNL*: The ORNL dataset contains real advanced attacks to the existing collection of open datasets. The dataset contains anonymized CAN data recorded during ambient driving and during several network attacks. The dataset contains real CAN data collected from a passenger vehicle with a variety of physically verified CAN attacks such as fuzzing fabrication attack, many targeted fabrication attacks, and masquerade attacks. The attacks are stealthy by manipulating only the necessary portions of the data field and sending a single manipulated message per ambient message of the same ID. Two advanced attacks that include no fabrication of injected messages are also included [34] making it a more complete dataset.

B. Evaluation metrics

The performance of the participating models were evaluated using a variety of performance metrics: accuracy, precision, recall, F1-score:

- Accuracy gives an estimate of the proportion of correctly predicted vehicle behavior to the total number of vehicle behaviors.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Accuracy is a suitable measure for balanced datasets but for imbalanced datasets, other performance metrics such

as precision, recall, and F-1 score are more suitable as they are less biased.

- Precision gives the proportion of correctly predicted malicious behaviors to the total predicted malicious behavior.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

- Recall gives the proportion of correctly predicted malicious behaviors to the total actual malicious behavior.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

- F1-score gives the weighted average of precision and recall.

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} \quad (4)$$

C. Performance analysis of the proposed model

A total of 25784 and 27997 instances from the survival analysis dataset and ORNL dataset were used for experimentation. A 60 to 40 and 50 to 50 of training to test data split was used for survival analysis dataset and ORNL dataset, respectively. A gaussian naive bayes model, a KNN model with 7 neighbors, a multi-layer perceptron model with 300 iterations, three control random forest models with 50, 100, and 150 trees, as RFmin, RFmed, and RFmax respectively, are trained. In addition, a grid search model (i.e., GRF) with the range of 50 to 150 trees is trained. A voting classifier (i.e., VRF) is trained with the three control random forest models where a soft voting mechanism is used. The soft votes are weighted by the F1-scores achieved from the training of the three control random forest models. For the proposed VGRF model, the soft vote of the grid search is also incorporated to generate a stable ensemble model which takes the votes of all the participating random forest models. Table I and II shows the model training performance with different feature sets and 10-fold cross validation. In the survival analysis dataset, the proposed VGRF model outperforms all the other state-of-the-art models in one out of four tests. In the case of the ORNL dataset, the proposed VGRF model outperforms all the other datasets in three of the tests. Table III and IV shows the model testing performance evaluated with accuracy, precision, recall, and F1-score. We select the feature set with the highest F1-score from the training to evaluate the model for testing. The SAD dataset outperforms all four tests while the highest scores in all the four metrics are achieved by the VRF model in ORNL which is close to the performance of the VGRF model.

D. Feature and control model contributions

The proposed explanation model was compared with Local Interpretable Model-agnostic Explanations (LIME) [15] in which the authors propose a concrete implementation of local surrogate models for finding feature importance. For both the datasets (in proposed explanation), SAD and ORNL, the most contributing features were found, ordered in decreasing importance, that is also linked to the most contributing control models and least contributing control models. The results

TABLE I: Survival analysis dataset training F1-score.

Features	30	60	90	113
NB	0.9118	0.9412	0.9412	0.9412
KNN	0.6776	0.6912	0.7079	0.7148
SVM	0.7804	0.8306	0.8412	0.8463
MLP	0.7979	0.8053	0.8277	0.8289
RF min	0.9536	0.9900	0.9902	0.9913
RF med	0.9545	0.9916	0.9909	0.9908
RF max	0.9554	0.9912	0.9908	0.9917
GRF	0.9550	0.9908	0.9904	0.9913
VRF	0.9558	0.9912	0.9919	0.9922
VGRF	0.9547	0.9908	0.9912	0.9928

TABLE II: ORNL dataset training F1-score.

Features	30	60	90	113
NB	0.2861	0.3351	0.3916	0.4176
KNN	0.3323	0.3405	0.3264	0.3256
MLP	0.3910	0.5120	0.5655	0.5666
RF min	0.7297	0.7749	0.8264	0.8371
RF med	0.7392	0.7835	0.8386	0.8436
RF max	0.7382	0.7839	0.8388	0.8473
GRF	0.7398	0.7835	0.8407	0.8429
VRF	0.7458	0.7886	0.8468	0.8508
VGRF	0.7470	0.7881	0.8483	0.8537

are shown in Table V. In the proposed explanation, the most contributing features in descending order were ID (10 frames), dt(10 frames), dt_ID(10 frames), win_most_freq_id, win_dur, win_mean_dt. The same order of importance was also observed by the LIME feature values. Among the control models, the models are affected by the selected feature set which also affects their individual performances. For VGRF to give its best performance, the most contributing control model in both the SAD and ORNL dataset was RF max and its dynamic weights were the highest. RF min was the least contributing control model and its dynamic weight was the least to make VGRF achieve its highest performance.

E. Average training time

A common trend in all the datasets is that the average training time increases with the increasing number of features. However, the training performance in terms of F1-score is not always increasing proportionately with increasing number of features. In case of our proposed model VGRF, with the SAD dataset, the highest F1-score is achieved at 113 features which has an average training time of 110s which is closer to the

Model	Accuracy	precision	recall	f1-score
NB	0.9526	0.9527	0.9526	0.9525
KNN	0.8188	0.8527	0.8188	0.8064
SVM	0.8940	0.9022	0.8940	0.8915
MLP	0.8355	0.8584	0.8355	0.8369
RF min	0.9929	0.9929	0.9929	0.9929
RF med	0.9933	0.9933	0.9933	0.9933
RF max	0.9948	0.9948	0.9948	0.9948
GRF	0.9940	0.9940	0.9940	0.9940
VRF	0.9943	0.9943	0.9943	0.9943
VGRF	0.9948	0.9948	0.9948	0.9948

TABLE IV: ORNL dataset testing performance with best training.

Model	Accuracy	precision	recall	f1-score
NB	0.7216	0.7740	0.7216	0.7416
KNN	0.8319	0.8069	0.8319	0.7993
MLP	0.8531	0.8452	0.8531	0.8481
RF min	0.9508	0.9535	0.9508	0.9478
RF med	0.9507	0.9534	0.9507	0.9477
RF max	0.9519	0.9544	0.9519	0.9490
GRF	0.9516	0.9543	0.9516	0.9488
VRF	0.9554	0.9576	0.9554	0.9530
VGRF	0.9548	0.9571	0.9548	0.9523

TABLE V: Explainability of feature and control models

Dataset	Proposed explanation	LIME value [15]
Most contributing features	ID	ID
	dt	dt
	dt_ID	dt_ID
	win_most_freq_id	win_most_freq_id
	win_dur	win_dur
	win_mean_dt	win_mean_dt
Most contributing control model	RF max	-
Least contributing control model	RF min	-

training time measured at 90 features (i.e., 95.03s). The ORNL dataset achieves the highest F1-score with 113 features at an average training time of 202.95s which is closer to the F1-score achieved with 90 features with an average training time of 142.51s.

V. CONCLUSION

In this paper, we have proposed a reliable and interpretable model for in-vehicle network intrusion detection system. The proposed IDS model is suitable for detecting attacks in CAN bus with a unique frequency-based approach. In particular, we have proposed an interpretable two-class attack detection model that leverages a dynamic voting-based exploration of features and control models. Moreover, our proposed approach can generalize to different datasets evaluated in terms of standard performance metrics such as F1 score and time comparison. In future, we will implement the proposed approach to other in-vehicle network datasets and explainability metrics.

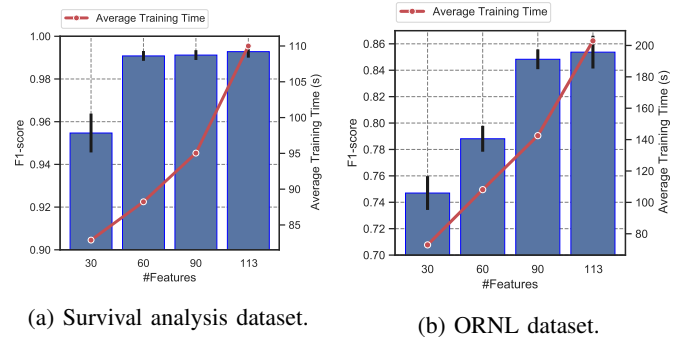


Figure 3: Training time trade-off with model performance.

REFERENCES

- [1] The Need for High Speed: IEEE Experts Predict 60 Percent of the Vehicles on the Road Will Be Internet-Enabled by 2025. Available at <https://www.prnewswire.com/news-releases/the-need-for-high-speed-ieee-experts-predict-60-percent-of-the-vehicles-on-the-road-will-be-internet-enabled-by-2025-221479981.html>
- [2] N. E. Team, Future Advances in Body Electronics, Nov. 2015, [online] Available: <https://www.nxp.com/docs/en/white-paper/BODYDELECTRWP.pdf>.
- [3] W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou, and K. Li. "A survey of intrusion detection for in-vehicle networks." *IEEE Transactions on Intelligent Transportation Systems* 21, no. 3 (2019): 919-933.
- [4] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle", 2015.
- [5] M. J. Dworkin, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, 2016.
- [6] S. Gyawali, Y. Qian, and R. Q. Hu. "Machine learning and reputation based misbehavior detection in vehicular communication networks." *IEEE Transactions on Vehicular Technology* 69, no. 8 (2020): 8871-8885.
- [7] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988-999, 1999.
- [8] J. Friedman, T. Hastie and R. Tibshirani, The elements of statistical learning, vol. 1(10), New York: Springer series in statistics, 2001.
- [9] P. L. Fung, M. A. Zaidan, H. Timonen, J. V. Niemi, A. Kousa, J. Kuula, K. Luoma et al. "Evaluation of white-box versus black-box machine learning models in estimating ambient black carbon concentration." *Journal of Aerosol Science* 152 (2021): 105694.
- [10] B. M. Doornenbal, B. R. Spisak, and P. A. van der Laken. "Opening the black box: Uncovering the leader trait paradigm through machine learning." *The Leadership Quarterly* (2021): 101515.
- [11] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman. "Metrics for explainable AI: Challenges and prospects." *arXiv preprint arXiv:1812.04608* (2018).
- [12] A. Holzinger. "From machine learning to explainable AI." In 2018 world symposium on digital intelligence for systems and machines (DISA), pp. 55-66. IEEE, 2018.
- [13] M. Wang, K. Zheng, Y. Yang, and X. Wang. "An explainable machine learning framework for intrusion detection systems." *IEEE Access* 8 (2020): 73127-73141.
- [14] S. Lundberg, "SHAP," 2020. [Online]. Available: <https://github.com/slundberg/shap>.
- [15] M. T. C. Ribeiro, "LIME," 2020. [Online]. Available: <https://github.com/marcotcr/lime>
- [16] AI Explainability 360 (v0.2.0)," 2019. [Online]. Available: <https://github.com/TrustedAI/AIX360>.
- [17] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "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, Sep 2016.
- [18] A. R. Wasicek, M. D. Pesé, A. Weimerskirch, Y. Burakova, and K. Singh, "Context-aware intrusion detection in automotive control systems," in *Proc. 5th ESCAR USA Conf.*, Jun. 2017, pp. 21-22.
- [19] M. R. Moore, R. A. Bridges, F. L. Combs, M. S. Starr, and S. J. Prowell, "Modeling inter-signal arrival times for accurate detection of CAN bus signal injection attacks: a data-driven approach to in-vehicle intrusion detection," in *Proceedings of the 12th Annual Conference on Cyber and Information Security Research - CISRC '17*. ACM Press, 2017, pp. 1-4. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3064814.3064816>
- [20] W. Choi, K. Joo, H. J. Jo, M. C. Park, and D. H. Lee. "Voltageids: Low-level communication characteristics for automotive intrusion detection system." *IEEE Transactions on Information Forensics and Security* 13, no. 8 (2018): 2114-2129.
- [21] U. E. Larson, D. K. Nilsson, and E. Jonsson, "An approach to specification-based attack detection for in-vehicle networks," in 2008 IEEE Intelligent Vehicles Symposium, Jun 2008, pp. 220-225.
- [22] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *Proc. IEEE International Conference on Data Science and Advanced Analytic (DSAA)*, Oct. 2016, pp. 130-139.
- [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] A. Theissler, "Anomaly detection in recordings from in-vehicle networks," in *Proc. Big Data Appl. Princ. (BIGDAP)*, vol. 23, Sep. 2014, pp. 1-12
- [25] S. N. Narayanan, S. Mittal, A. Joshi et al., "Using data analytics to detect anomalous states in vehicles," *arXiv:1512.08048*, 2015.
- [26] J. Zhang, and M. Zulkernine. "Network intrusion detection using random forests," 2005
- [27] N. Kaja, A. Shaout, and D. Ma. "An intelligent intrusion detection system." *Applied Intelligence* 49, no. 9 (2019): 3235-3247.
- [28] R. Primartha, and B. A. Tama. "Anomaly detection using random forest: A performance revisited." In 2017 International conference on data and software engineering (ICoDSE), pp. 1-6. IEEE, 2017.
- [29] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano. "Explainable artificial intelligence (xai) to enhance trust management in intrusion detection systems using decision tree model." *Complexity* 2021 (2021).
- [30] A. Taylor, N. Japkowicz, and S. Leblanc. "Frequency-based anomaly detection for the automotive CAN bus." In 2015 World Congress on Industrial Control Systems Security (WCICSS), pp. 45-49. IEEE, 2015.
- [31] C. Young, H. Olufowobi, G. Bloom, and J. Zambreno. "Automotive intrusion detection based on constant can message frequencies across vehicle driving modes." In *Proceedings of the ACM Workshop on Automotive Cybersecurity*, pp. 9-14. 2019.
- [32] A. Jović, K. Brkić, and N. Bogunović. "A review of feature selection methods with applications." In 2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO), pp. 1200-1205. Ieee, 2015.
- [33] 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.
- [34] M. E. Verma, M. D. Iannaccone, 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.



Nishat I Mowla received her B.S. in Computer Science from Asian University for Women, Chittagong, Bangladesh, in 2013. She received her M.S. and Ph.D. degree in Computer Science and Engineering from Ewha Womans University, Seoul, Korea in 2016 and 2020, respectively. She is currently a senior researcher at the department of Mobility and Systems of RISE, Sweden. Her research interests include network security, machine intelligence, and network traffic analysis.



Joakim Rosell received his M.Sc. and Ph.D. degrees at the department of physics at Lund University, Sweden, in 2005 and 2016, respectively. He has worked as a subject matter expert and research engineer for automotive LiDAR sensors at Volvo GTT and is now working as a senior researcher at the department of Mobility and Systems of RISE, Sweden. His research interests include machine learning, perception and sensors as well as computer vision.



Arash Vahidi received his Licentiate and Ph.D. in Formal Methods in Automation from Chalmers University of Technology, Sweden in 2002 and 2004 respectively. He has in past worked with cryptographic systems and network security systems and is currently a senior researcher at RISE as a member of the cybersecurity unit. His research interests include security in embedded systems, security in multi-tenant computer systems and security in AI systems.