AttentionGuard: Transformer-based Misbehavior Detection for Secure Vehicular Platoons

Hexu Li*
Networked Systems Security (NSS) Group
KTH Royal Institute of Technology
Stockholm, Sweden
hexu@kth.se

Konstantinos Kalogiannis*
Networked Systems Security (NSS) Group
KTH Royal Institute of Technology
Stockholm, Sweden
konkal@kth.se

Ahmed Mohamed Hussain*
Networked Systems Security (NSS) Group
KTH Royal Institute of Technology
Stockholm, Sweden
ahmhus@kth.se

Panos Papadimitratos
Networked Systems Security (NSS) Group
KTH Royal Institute of Technology
Stockholm, Sweden
papadim@kth.se

Abstract

Vehicle platooning, with vehicles traveling in close formation coordinated through Vehicle-to-Everything (V2X) communications, offers significant benefits in fuel efficiency and road utilization. However, it is vulnerable to sophisticated falsification attacks by authenticated insiders that can destabilize the formation and potentially cause catastrophic collisions. This paper addresses this challenge: misbehavior detection in vehicle platooning systems. We present *AttentionGuard*, a transformer-based framework for misbehavior detection that leverages the self-attention mechanism to identify anomalous patterns in mobility data. Our proposal employs a multi-head transformer-encoder to process sequential kinematic information, enabling effective differentiation between normal mobility patterns and falsification attacks across diverse platooning scenarios, including steady-state (no-maneuver) operation, join, and exit maneuvers. Our evaluation uses an extensive simulation dataset featuring various attack vectors (constant, gradual, and combined falsifications) and operational parameters (controller types, vehicle speeds, and attacker positions). Experimental results demonstrate that *AttentionGuard* achieves up to 0.95 F1-score in attack detection, with robust performance maintained during complex maneuvers. Notably, our system performs effectively with minimal latency (100ms decision intervals), making it suitable for real-time transportation safety applications. Comparative analysis reveals superior detection capabilities and establishes the transformer-encoder as a promising approach for securing Cooperative Intelligent Transport Systems (C-ITS) against sophisticated insider threats.

CCS Concepts

• **Networks** → **Network security**; • **Security and privacy** → **Distributed systems security**; **Intrusion detection systems**.

*Equally contributing authors.

WiseML 2025, Arlington, VA, USA

ACM ISBN 979-8-4007-1531-0/2025/06
https://doi.org/10.1145/3733965.3733966

Keywords

Transformer Encoder, Anomaly Detection, Vehicular Platoons, V2X, Maneuvering

ACM Reference Format:
Hexu Li, Konstantinos Kalogiannis, Ahmed Mohamed Hussain, and Panos Papadimitratos. 2025. AttentionGuard: Transformer-based Misbehavior Detection for Secure Vehicular Platoons. In *Proceedings of the 2025 ACM Workshop on Wireless Security and Machine Learning (WiseML 2025), July 3, 2025, Arlington, VA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3733965.3733966

1 Introduction

Vehicular Ad-hoc Networks (VANETs) and Cooperative Intelligent Transport System (C-ITS) have emerged as promising technologies to enhance road safety, improve traffic efficiency, and provide a more comfortable driving experience [1, 20]. An important application in this domain is vehicle platooning, where multiple vehicles travel in close formation with reduced inter-vehicle distances, coordinated through Vehicle-to-Everything (V2X) communications [21, 22]. While platooning offers substantial benefits, such as reduced fuel consumption, improved traffic throughput, and enhanced safety, it also introduces significant security vulnerabilities that must be addressed [12].

The vehicle platooning security challenges stem from the dependence on the continuous exchange of Cooperative Awareness Messages (CAMs) containing kinematic data (i.e., position, speed, acceleration) among vehicles. Cryptographic approaches provide a first line of defense, mitigating external to the platoon attackers and safeguarding against Sybil adversaries [15]; but they remain ineffective against insider threats–authenticated vehicles that deliberately transmit falsified mobility data [23]. Such falsification attacks can destabilize platoon formations and potentially lead to catastrophic collisions, particularly during coordinated maneuvers, such as joining or exiting a platoon [12].

Misbehavior Detection Schemes (MDSs) are an essential second line of defense, identifying anomalous patterns in V2X communications that may indicate malicious activity [13, 14]. Traditional MDS approaches typically rely on pre-defined rules, thresholds, or plausibility checks, which, while computationally efficient, often fail to adapt to the complex dynamics of vehicular environments. More

recent approaches leverage Machine Learning (ML) approaches to improve detection capabilities [14]; however, many challenges persist, including (i) the ability to distinguish between legitimate maneuvers and malicious behavior [12], (ii) computational constraints on resource-limited On-Board Units (OBUs) [3], and (iii) the need for real-time misbehavior detection to enable timely mitigation [11].

In this paper, we propose a novel transformer-based architecture for misbehavior detection in vehicular platoons that addresses the aforementioned challenges. Our approach leverages the temporal context awareness of transformer encoders to effectively differentiate between normal mobility patterns and sophisticated falsification attacks, even during complex maneuvers.

Contributions. We summarize the main contributions of this work as follows: (i) A reliable and timely MDS capable of identifying malicious behavior in diverse mobility scenarios, including steady-state and maneuvering conditions. (ii) Evidence on the capability of transformer-based approaches to identify attacks even when the inference-phase data input differs from the training dataset. (iii) Insights on the deployment of transformer-based MDSs, in vehicles or on the edge, depending on acceptable reaction times and detection performance.

Paper Organization. Sec. 2 reviews related work on misbehavior detection in vehicular networks, focusing on ML approaches for platooning security. Sec. 3 presents our system and adversary model, detailing platooning environments and attack scenarios. Sec. 4 describes our transformer-based framework, including the preprocessing, architecture, and implementation details. Sec. 5 evaluates our approach and analyzes practical deployment considerations for vehicle and infrastructure-based detection systems. Finally, Sec. 6 concludes with key findings and future research directions.

2 Related Work

Vehicle platooning systems rely on reliable V2X communications and continuous CAM exchange. The coordinated nature of platooning operations (e.g., middle-join, exit) [11, 12] necessitates robust misbehavior detection mechanisms to ensure operational integrity and satisfy safety-critical requirements. Traditional ML approaches demonstrated significant efficacy in V2X misbehavior detection [14]. These include Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and k-Nearest Neighbor (k-NN), enabling systematic identification of anomalous patterns in network messages and vehicle behaviors.

Such approaches heavily depend on feature engineering and model adaptation to dynamic vehicular environments. Grover et al. [8] developed a comprehensive feature set encompassing geographical position, acceptance range, Received Signal Strength (RSS), speed, and packet delivery metrics for misbehavior classification in VANET. They utilized multiple classifiers, including RF and k-NN, and validated the feasibility of resource-constrained misbehavior detection. Additionally, Gyawali et al. [6] integrated ML with reputation-based systems, enhancing detection accuracy through Dempster-Shafer (DS)-based feedback combination. Ercan et al. [3] proposed a distributed Intrusion Detection System (IDS) utilizing ensemble learning techniques, demonstrating high performance through combining k-NN and RF classifiers.

Deep Learning (DL) approaches have shown particular promise. Liu [16] implemented a Long Short-Term Memory (LSTM)-based architecture for anomaly detection in the VeReMi dataset [24], improving on False Positive (FP) rates compared to conventional detectors. Hsu et al. [9] developed a hybrid architecture combining Convolutional Neural Network (CNN) and LSTM networks, achieving 95.35% detection accuracy across multiple misbehavior attacks. Advanced collaborative approaches have further enhanced detection capabilities. Mangla et al. [17] proposed a fusion framework integrating SVM, Multi-Layer Perceptron (MLP), and LSTM classifiers, achieving 99.99% accuracy in multi-class misbehavior detection. Gurjar et al. [7] addressed privacy concerns through Federated Learning (FL), enabling distributed model training while preserving data privacy. Recently, Transformer-based IDS have been developed as a way to detect attacks in In-Vehicle Networks (IVNs) [2, 18, 19].

Comparison with Existing Work. Our MDS incorporates a Transformer-encoder that captures the unique mobility characteristics of different maneuvering (attack and benign) scenarios. Compared to the Transformer-based solutions, our aim is not to detect attacks within IVN; we detect adversarial behavior in platoon formations by analyzing the platooning vehicles' sensor readings and mobility patterns.

3 System and Adversary Model

We consider a set of V2X-enabled vehicles forming a platoon on a highway and traveling in unison. During the journey, other V2X-enabled vehicles can request to join the formation to gain platooning benefits. The platoon leader can accept such a request and designate a join position based on the requesting car's capabilities and target destination. Similarly, platooning vehicles can decide to exit their formation during the trip, e.g., when approaching their destination, by initiating an exit request and performing the maneuver when available.

It is exactly at these moments that the attacker can choose to strike, to maximize its attack potential effect (harm). More specifically, we study a threat model applicable to Vehicular Communication (VC) systems [20], and more specifically to platooning [1, 12]. An attacker possesses valid cryptographic credentials and can join platoons to perform its attacks: the attacker cannot impersonate another vehicle, but it can alter the kinematic properties (i.e., position, speed, acceleration) of the CAMs it transmits. Based on the dataset in [12], the attacks can introduce a constant or a gradual offset to each of the kinematic values; or alter them intelligently, and in unison, in a physics-consistent way, termed "combined" attacks. These nine types of attacks are executed against platoons in different mobility scenarios: a join, an exit, or a steady-state scenario.

4 Proposed Framework

We propose an MDS framework consisting of three phases, namely: (i) data pre-processing, (ii) model training, and (iii) model evaluation.

Data Pre-Processing. The available simulated data consists of platoons of up to seven vehicles traveling on a straight highway for 118.9 seconds, recording seven mobility properties (*distance*, *relativeSpeed*, *acceleration*, *controllerAcceleration*, *speed*, *posx*, and

posy). To account for varying vehicle insertion times (appearing in the simulation), we utilize a *mask* array indicating valid features for each car. Finally, we classify data as benign (0), or attack (1) if they deviate from the ground truth; for scenarios under the same seed, the data deviate only when an attack affects the vehicle controller. This results in an array of shape (7, 1189, 7) for the features and an array of shape (7, 1189) for the labels and the mask. Table 1 summarizes the total number of traces and the ratio between the benign and attack traces processed.

Table 1: Outcome of preprocessing for each seed.

Seed	Total Traces	Valid Labels	Ratio of 1	Ratio of 0	0/1 Ratio
0-02-1	1454	8307624	0.2111	0.7889	3.7369
2016-02-1	1620	9568553	0.2863	0.7137	2.4926
2083-02-1	1449	8286245	0.2217	0.7783	3.5101
2084-02-1	1458	8376577	0.2209	0.7791	3.5278
2085-02-1	1426	8119617	0.2199	0.7801	3.5469
2086-02-1	1434	8168407	0.2174	0.7826	3.6007

To ensure all features contribute equally during the *model training* phase, we normalize values using a global mean μ and variance σ^2. Specifically, let $\mathbf{x} \in \mathbb{R}^7$ be the raw feature vector at a given time step. We first compute:

$$\mu = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n, \quad \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \mu)^2, \tag{1}$$

where N is the total number of feature vectors across all files. Then, each feature is normalized to:

$$\hat{\mathbf{x}} = \frac{\mathbf{x} - \mu}{\sigma + \epsilon}, \tag{2}$$

with $\epsilon \approx 10^{-8}$ to avoid division by zero. The z-score transformation ensures that each dimension has a near-zero mean and unit variance, reducing scale discrepancies among different scenarios. After normalization, we split the data into equal-sized sliding windows (10 data points) and introduce padding where applicable.

Considered Model. We utilize the Transformer-encoder architecture because it can generate output data by capturing the context of the input data [25]. Unlike the sequential processing of data performed by Recurrent Neural Networks (RNNs), Transformers can at once process the entire data sequence, allowing for extracting relationships and context within the sequence. The overall Transformer architecture (introduced in [25]) includes two primary components: the *encoder* and the *decoder*. The encoder is tasked with processing the input data sequence, transforming it into a continuous representation that encapsulates contextual information. This is achieved through multiple layers, each incorporating a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. The self-attention mechanism enables the encoder to assign varying degrees of importance to different parts of the input sequence, thereby generating a comprehensive representation of the data.

The complete architecture of our transformer block is presented in Fig. 1. For this domain-specific problem, our approach consists of two model training approaches: (i) a *general* platooning model utilizing all vehicle data as input, (ii) a *vehicle-specific* model trained only on its own mobility data. The aim is to provide insights on the feasibility of deploying general or vehicle-specific models to ensure the vehicular formation security.

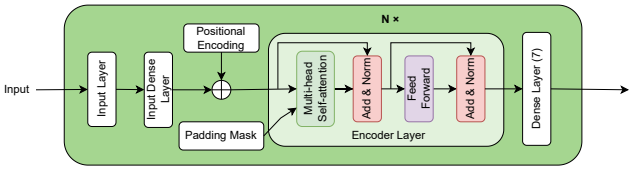

Figure 1: Structure and details of the implemented transformer-encoder.

Considering the binary nature of the problem (benign vs attack characterization), we utilize the masked binary-cross entropy loss function, presented in Equation 3. Let $y_i \in \{0, 1\}$ be the ground-truth label for step i, and $\text{mask}_i \in \{0, 1\}$ indicate whether that step is valid (i.e. not padded). The model outputs a logit $\hat{z}_i \in \mathbb{R}$. We define a *positive weight* $\alpha > 1$ to handle the class imbalance inherent in our data. Thus, for each valid step ($\text{mask}_i = 1$), the BCE loss is:

$$\mathcal{L}_{\text{BCE}} = -\sum_i \text{mask}_i \left[(1 - y_i)\, \log\big(\sigma(\hat{z}_i)\big) \;+\; y_i\, \alpha\, \log\big(\sigma(\hat{z}_i)\big) \right] \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function. All steps with $\text{mask}_i = 0$ do not contribute to the loss. Finally, we normalize by the total number of valid steps to get an average loss.

Model Evaluation. To drive the discussion towards the location deployment of MDSs, in-vehicle, on the edge, or at a Roadside Unit (RSU), we run the inference phase of the general platooning model with two different inputs: either locally, with just the vehicle own mobility data, or after collecting the entire platoon mobility information. Additionally, we evaluate each vehicle's model performance against the general platooning model.

Our evaluation of attack detection employs several metrics, *recall*, *precision*, F_1 *score*, and *accuracy*; giving us, respectively, the fraction of true positive steps among all actual positives, the fraction of correctly identified positives among all predicted positives, the harmonic mean of the previous two metrics (useful when dealing with unbalanced classes), and, finally, the indicator of correct predictions. Furthermore, we use the Receiver Operating Characteristic (ROC), the proportion of correctly detected messages (True Positive (TP)) over incorrectly identified messages (FP), to measure the ability of our identifier to correctly classify the input data. Additionally, we provide the Area Under the Curve (AUC) to facilitate the classifier comparison.

5 Performance Evaluation

5.1 Training Setup

The transformer-encoder is implemented using TensorFlow [5] and Keras [4]; for training, we used a batch size of 128, chosen empirically, and a learning rate of $5\epsilon^{-5}$ for the Adaptive Moment Estimation (ADAM) optimizer. We set the positive weights for all models depending on the ratio of benign and attack labels, as described in Table 1, e.g., to 3.3 for the general platooning model. For

positional encoding, we utilize the $\sin(\cdot)$ and $\cos(\cdot)$ functions [25]. As part of the training and inference process, we define the window size to be 10, i.e., we gather 1s worth of data, and consider a variable step size; from 1 to 10, signifying the required data before making the next prediction (from 100ms to 1s, respectively). Table 2 outlines all the parameters used in the model training setup.

Table 2: Training parameters.

Transformer-encoder		Platoons	
Parameters	**Value**	**Parameters**	**Value**
Batch Size	128	Cars	6 or 7 (Join)
Data Split Ratio	80/20	Spacing Controller	PATH, Flatbed
Learning Rate	$5\epsilon^{-5}$	Headway Controller	Ploeg, Consensus
Window size	10	Leader Speed	50, 80, 100, 150 kmph
Step size	[1, 5, 10]	Spacing	5m, 5m, 0.5s, 0.8s
Positive Weight/Positional Encoding	Dataset/Model dependent	Sim. Duration	120s
Loss Function	Binary Crossentropy	Attacker Position	[0, 2]

On the other hand, the input dataset describes platoons of 6 vehicles, increasing to 7 when a join needs to be performed, that travel on the highway with different speeds. The platoons themselves utilize different controllers, either Constant Vehicle Spacing (CVS), where the vehicles try to keep a constant spacing distance; or Constant Time Headway (CTH), where the vehicles' intra-platoon distances are based on speed and the time it would take to reach the bumper of the car ahead. The different controllers and speeds contribute to the diverse mobility characteristics of the vehicles. Finally, we consider attackers positions either at the front, as the platoon leader, or at position three (Vehicle 2), as a regular platoon follower; the latter corresponds to the position just in front of the join or exit positions [12]. The analysis of our data was performed on an Ubuntu machine, using 128 GB of Random Access Memory (RAM), an AMD Ryzen Threadripper PRO 5965WX with 24 physical cores and 48 logical cores, and an NVIDIA GeForce RTX 4090 with 24 GB of DDR5 memory.

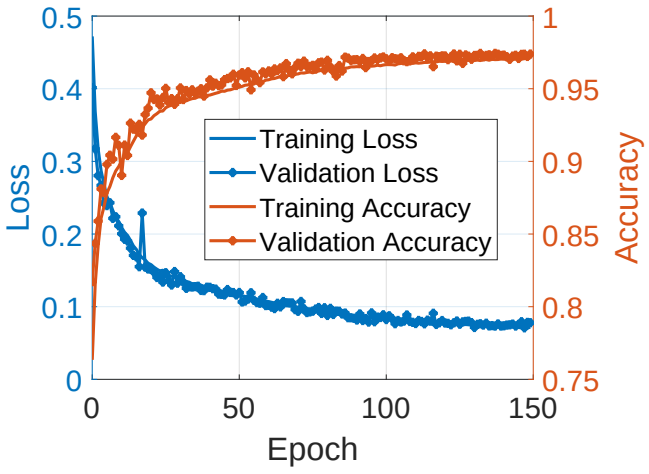

Figure 2: Model training/validation Loss/Accuracy as a function of number of epochs.

5.2 Evaluation Results

Fig. 2 illustrates the loss and accuracy values per epoch for a maximum of 150 epochs during the training phase of the general platooning model. We observe that both values steadily converge, indicating that the model can learn from our input. During the training phase, we reach an accuracy of 0.96 on the validation and training datasets. We omit the training plots for the individual car models.

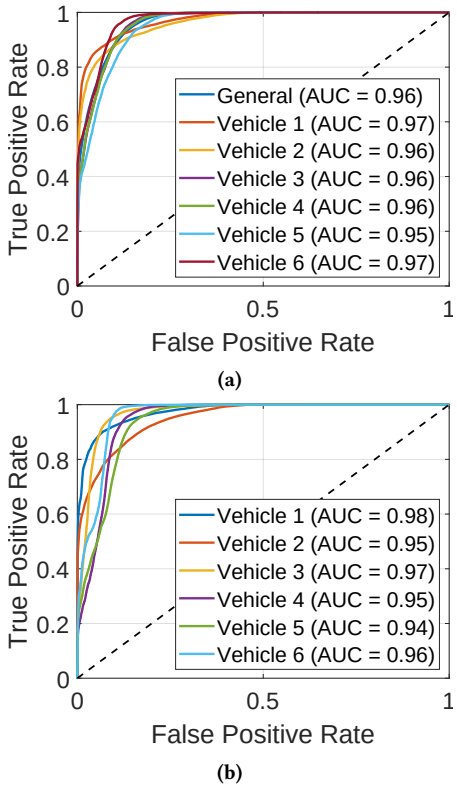

Figure 3: ROC curves for step = 5: (a) Global (General) Model, (b) Vehicle Model.

We consider model input, with a step equal to 5, a balance between timely predictions and detection performance (shown in Tables 3 and 4). Fig. 3 presents the ROC curves comparing the detection performance of *AttentionGuard*. In Fig. 3a, the general platoon-level model (with the platoon input) achieves an AUC of 0.96, demonstrating robust classification capabilities. Individual vehicle inputs show similar but lower performance, with Vehicles 3 and 6 exhibiting better detection rates at lower FP rates. In comparison, vehicles 1 and 2 show steeper initial positive rates, but with a lower convergence as the input increases. Vehicle 0 (platoon leader) is not shown, as the attacks have no effect, and the classifier results in >0.99 across the evaluation metrics. In Fig. 3b, we showcase the performance of the individual car models: the majority perform better, but notably, vehicles 2 and 5 show higher FP rates. These vehicles are the furthest away from both attacking vehicles (0 and 2, respectively), making distinguishing between misbehavior and benign movement harder. Across all models, these

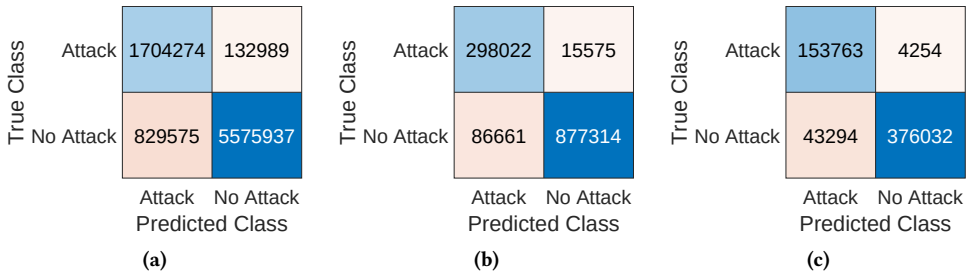

Figure 4: Confusion matrix for step = 5: (a) Global Model, (b) Vehicle 3 Model, and (c) Vehicle 6 Model.

Table 3: Performance comparison for general platooning and individual vehicle models.

Input	Step = 1				Step = 5				Step = 10			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
General	0.88	0.91	0.88	0.89	0.88	0.91	0.88	0.89	0.91	0.93	0.91	0.91
Vehicle 1	0.90	0.92	0.90	0.90	0.90	0.93	0.90	0.91	0.95	0.95	0.95	0.95
Vehicle 2	0.84	0.89	0.84	0.85	0.85	0.90	0.85	0.86	0.91	0.93	0.91	0.91
Vehicle 3	0.92	0.93	0.92	0.92	0.92	0.93	0.92	0.92	0.94	0.95	0.94	0.94
Vehicle 4	0.89	0.90	0.89	0.89	0.90	0.91	0.90	0.90	0.93	0.93	0.93	0.93
Vehicle 5	0.87	0.89	0.87	0.87	0.88	0.89	0.88	0.88	0.90	0.91	0.90	0.90
Vehicle 6	0.92	0.93	0.92	0.92	0.92	0.93	0.92	0.92	0.92	0.93	0.92	0.92

consistently high AUC values validate the transformer architecture capability to capture attack patterns, making it an effective approach for misbehavior detection in vehicle platooning scenarios.

Fig. 4 presents the confusion matrices for three different models utilizing a step 5 input, i.e., 500ms worth of new data. The platoon-level model, in Fig. 4a has 1,704,274 TPs and 5,575,937 True Negatives (TNs), indicating effective identification of non-attack instances across the entire formation. However, the model does not avoid misclassifications, with 829,575 FPs and 132,989 False Negatives (FNs); affecting the precision of the attack detection. This translates to an accuracy of 0.88, with a weighted precision of 0.91, recall of 0.88, resulting in a F1-score of 0.89 as detailed in Table 3.

In Fig. 4b and Fig. 4c, we show the confusion matrix for vehicles 3 and 6 respectively (i.e., in each case a vehicle-specific model). These correspond to the vehicle traveling behind the follower-attacker and the vehicle entering the formation just behind it, respectively, a prime target to cause harm to both the platoon downstream and vehicles on the next lane. For vehicle 3, the confusion matrix contains a smaller number of benign and attack samples; however, the classifier improves substantially, with an increase of 0.2-0.4 across all metrics, even during maneuvering. For vehicle 6, the model demonstrates similar improvements (from the general model), illustrating the effectiveness of our approach, even for the joining vehicle.

5.3 MDS Deployment

An essential part of any vehicular MDS, particularly in platooning where vehicles travel closely together, is to swiftly and reliably determine any abnormal mobility. In Tables 3 and 4, we observe that with smaller step sizes, the MDS trades off worse performance across all evaluation metrics, for faster reaction to any detected misbehavior. A step size of 10 can potentially be catastrophic for the platoon, making a faster but less precise decision preferable when coupled with an adequate mitigation response [11].

In Table 3, we evaluate the performance of the different models during inference, using weighted averages. The general platooning model confirms the result in Fig. 3b), by performing slightly better than the individual models of vehicles 2 and 5. Nonetheless, the rest of the individual models show an increase for all metrics; specifically, the F1-score improves by 0.1-0.3 for the step equal to 1; 0.2-0.3 for the step equal to 5; and 0.1-0.4, reaching a total of 0.95 for vehicle 1. This can be anticipated as individual models were trained specifically for each vehicle. Comparing the individual vehicle performance, vehicles closest to the attackers achieve the best detection rates, even compared to the general model.

For the general platooning model (Table 4), individual vehicle inputs result in varying performance, depending on the position of the vehicles in the formation. Vehicles 1 and 2 are unaffected by all follower attacks (performed by Vehicle 2 itself), as they are upstream and can only detect malicious behavior of a leader attacker (Vehicle 0). Resulting in degraded MDS performance, highlighted in red (with a downward trend for smaller step size), as the model is trained on the full dataset and not just their behavior. Both vehicles show a high number of FPs, affecting the attack-class precision, and FNs, affecting the recall. Notably, a vehicle that decides to enter the platoon formation (Vehicle 6 in our simulations) performs well according to all the metrics (lowest being 0.90), for a step equal to 1. This means that the joiner can immediately distinguish misbehavior, avoid joining the formation and getting affected, thus avoiding causing hazardous conditions to the platoon itself, or other non-platooning vehicles on the road.

Discussion. Examining both model training approaches (general platooning and vehicle-specific), the individual models generally perform better (except for vehicles 2 and 5). However, this requires vehicles having different models depending on their position in a platoon, effectively limiting them to platoons of size equal to their number of models. With a general platooning model, individual

Table 4: Performance comparison for the general platooning model with vehicle-specific input.

Input	Step = 1				Step = 5				Step = 10			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
General	0.88	0.91	0.88	0.89	0.88	0.91	0.88	0.89	0.91	0.93	0.91	0.91
Vehicle 1	0.83	0.90	0.83	0.85	0.84	0.91	0.84	0.85	0.89	0.93	0.89	0.90
Vehicle 2	0.80	0.89	0.80	0.82	0.80	0.90	0.80	0.82	0.85	0.91	0.85	0.86
Vehicle 3	0.89	0.91	0.89	0.90	0.89	0.91	0.89	0.90	0.92	0.93	0.92	0.92
Vehicle 4	0.89	0.90	0.89	0.89	0.90	0.90	0.90	0.90	0.92	0.92	0.92	0.92
Vehicle 5	0.88	0.89	0.88	0.88	0.88	0.89	0.88	0.88	0.90	0.91	0.90	0.90
Vehicle 6	0.90	0.92	0.90	0.91	0.91	0.92	0.91	0.91	0.91	0.92	0.91	0.92

inference inputs provide comparable results to the entire platoon input, except for vehicles 1 and 2. This implies that training and inference on all the platoons' data can effectively safeguard a platoon by discerning platoon misbehavior. This allows for a more flexible deployment of MDSs, e.g., on an RSU or the platoon leader, enabling it to detect misbehavior within the platoon. Further, given higher step sizes, when timely decisions may not be critical, such an MDS could act as a forensic tool for post-attack analysis.

However, deploying *AttentionGuard* on a vehicle or an edge device (representing an RSU) presents several challenges. Notably, the limited computational power and memory constraints of such platforms. Towards addressing these challenges, we propose using optimization methods provided by the TensorFlow library. Namely, model conversion to TensorFlow Lite (TFLite) and quantization. Such optimizations can significantly reduce the model size and improve inference time, enabling deployment without accuracy degradation [10].

6 Conclusion

We presented *AttentionGuard*, a transformer-based MDS framework to safeguard vehicle platoon formations, operating under different controllers, speeds, and maneuvering states. Our evaluation shows that our approach can provide high detection rates and fast reaction, allowing vehicles to detect attacks even after 100ms. Further, we presented results that support the training of models on entire platoon data, while still guaranteeing the detection performance of individual cars when deployed locally. This enables diverse configurations depending on the preferred outcome. As part of our future work, we will expand our detection scheme to cover a plethora of network attacks in platooning and apply model optimizations for deployment on resource-constrained devices while analyzing their overall performance.

Acknowledgments

This work is supported in parts by the Swedish Research Council (VR) and the Knut and Alice Wallenberg (KAW) Foundation.

References

[1] Amoozadeh et al. 2015. Security Vulnerabilities of Connected Vehicle Streams and Their Impact on Cooperative Driving. *IEEE Comm. Mag.* 53, 6 (Jun. 2015).
[2] Cobilean et al. 2023. Anomaly Detection for In-Vehicle Communication Using Transformers. In *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*. 1–6. doi:10.1109/IECON51785.2023.10311788
[3] Secil Ercan, Marwane Ayaida, and Nadhir Messai. 2022. Misbehavior Detection for Position Falsification Attacks in VANETs Using Machine Learning. *IEEE Access* 10 (2022), 1893–1904. doi:10.1109/ACCESS.2021.3136706
[4] François Chollet. 2025. *Keras*. https://keras.io/ Accessed: Feb 2025.
[5] Google. 2025. *TensorFlow*. https://www.tensorflow.org Accessed: Feb 2025.
[6] Grover et al. 2011. Machine Learning Approach for Multiple Misbehavior Detection in VANET. In *Advances in Computing and Communications*. Springer Berlin Heidelberg, Berlin, Heidelberg, 644–653.
[7] Dayanand Gurjar, Jyoti Grover, Vanisha Kheterpal, and Athanasios Vasilakos. 2025. Federated learning-based misbehavior classification system for VANET intrusion detection. *Journal of Intelligent Information Systems* (16 Jan 2025). doi:10.1007/s10844-025-00920-0
[8] Sohan Gyawali, Yi Qian, and Rose Qingyang Hu. 2020. Machine Learning and Reputation Based Misbehavior Detection in Vehicular Communication Networks. *IEEE Transactions on Vehicular Technology* 69, 8 (2020), 8871–8885. doi:10.1109/TVT.2020.2996620
[9] Hsiao-Yuan Hsu, Nai-Hsin Cheng, and Chun-Wei Tsai. 2022. A Deep Learning-Based Integrated Algorithm for Misbehavior Detection System in VANETs. In *Proceedings of the 2021 ACM International Conference on Intelligent Computing and Its Emerging Applications* (Jinan, China) *(ACM ICEA '21)*. Association for Computing Machinery, New York, NY, USA, 53–58. doi:10.1145/3491396.3506509
[10] Ahmed Mohamed Hussain, Nada Abughanam, and Panos Papadimitratos. 2024. Edge AI-based Radio Frequency Fingerprinting for IoT Networks. *arXiv preprint arXiv:2412.10553* (2024).
[11] Konstantinos Kalogiannis, Michael Hartmann, and Panos Papadimitratos. 2024. PRIME: Platoon Restructuring for Incident Mitigation and Exclusion. In *2024 20th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 172–177. doi:10.1109/WiMob61911.2024.10770319
[12] Konstantinos Kalogiannis, Mohammad Khodaei, Weaam Mostafa Nemr Mohamed Bayaa, and Panos Papadimitratos. 2022. Attack impact and misbehavior detection in vehicular platoons. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 45–59.
[13] Kamel et al. 2019. CaTch: A Confidence Range Tolerant Misbehavior Detection Approach. In *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. 1–8. doi:10.1109/WCNC.2019.8885740
[14] Kamel et al. 2020. Simulation framework for misbehavior detection in vehicular networks. *IEEE Transactions on Vehicular Technology* 69, 6 (2020), 6631–6643.
[15] M. Khodaei et al. 2018. SECMACE: Scalable and Robust Identity and Credential Management Infrastructure in Vehicular Communication Systems. *IEEE TITS* 19, 5 (May 2018), 1430–1444.
[16] Xiangyu Liu. 2022. Misbehavior Detection based on Deep Learning for VANETs. In *2022 International Conference on Networks, Communications and Information Technology (CNCIT)*. 122–128. doi:10.1109/CNCIT56797.2022.00027
[17] Cherry Mangla, Shalli Rani, and Norbert Herencsar. 2023. A misbehavior detection framework for cooperative intelligent transport systems. *ISA Transactions* 132 (2023), 52–60. doi:10.1016/j.isatra.2022.08.029
[18] Minki Nam, Seungyoung Park, and Duk Soo Kim. 2021. Intrusion Detection Method Using Bi-Directional GPT for in-Vehicle Controller Area Networks. *IEEE Access* 9 (2021), 124931–124944. doi:10.1109/ACCESS.2021.3110524
[19] Trieu Phong Nguyen, Heungwoo Nam, and Daehee Kim. 2023. Transformer-Based Attention Network for In-Vehicle Intrusion Detection. *IEEE Access* 11 (2023), 55389–55403. doi:10.1109/ACCESS.2023.3282110
[20] P. Papadimitratos et al. 2008. Secure Vehicular Communication Systems: Design and Architecture. *IEEE Comm. Mag.* 46, 11 (Nov. 2008), 100–109.
[21] Ploeg et al. 2011. Design and Experimental Evaluation of Cooperative Adaptive Cruise Control. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. 260–265. doi:10.1109/ITSC.2011.6082981
[22] Santini et al. 2017. A Consensus-Based Approach for Platooning with Intervehicular Communications and Its Validation in Realistic Scenarios. *IEEE Transactions on Vehicular Technology* 66, 3 (2017), 1985–1999. doi:10.1109/TVT.2016.2585018
[23] R. van der Heijden et al. 2017. Analyzing Attacks on Cooperative Adaptive Cruise Control (CACC). In *IEEE Vehicular Networking Conference (VNC)*. Torino, Italy.
[24] Rens W Van Der Heijden, Thomas Lukaseder, and Frank Kargl. 2018. Veremi: A dataset for comparable evaluation of misbehavior detection in vanets. In *14th International Conference, SecureComm 2018, Singapore, Singapore, August 8-10, 2018*. 318–337.
[25] Vaswani et al. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).