PAMPOS: Causal Transformer-based Trajectory Prediction for Attack-Agnostic Misbehavior Detection in V2X Networks

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

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

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

Abstract

Misbehavior detection in Vehicle-to-Everything (V2X) networks is a second line of defense against insider falsification attacks that cryptographic mechanisms alone cannot address. Existing learning-based Misbehavior Detection Schemes (MDSs) are supervised, requiring labeled attack samples at training time, thus failing to counter unseen falsification attacks. We present PAMPOS, a causal transformer-decoder trained on benign VeReMi++ trajectories to learn normal mobility patterns. At inference time, misbehavior is identified as a deviation from the model's next-step kinematic predictions using a top-K normalized anomaly scoring mechanism that localizes falsification to specific kinematic features, without requiring attack-labeled training data. We evaluate PAMPOS across all 19 attack types in VeReMi++ under rush-hour and afternoon scenarios, achieving Area Under the Curve (AUC) values of up to 0.98 and F1-scores of up to 0.95 for most attack categories.

CCS Concepts

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

Keywords

Transformer Decoder, Anomaly Detection, Vehicular Network, V2X

ACM Reference Format:
Konstantinos Kalogiannis, Ahmed Mohamed Hussain, and Panos Papadimitratos. 2026. PAMPOS: Causal Transformer-based Trajectory Prediction for Attack-Agnostic Misbehavior Detection in V2X Networks. In *Proceedings of the 2026 ACM Workshop on Wireless Security and Machine Learning (WiseML '26), June 30-July 03, 2026, Saarbrücken, Germany*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3811880.3815104

1 Introduction

Vehicular Ad-hoc Networks (VANETs) and Cooperative Intelligent Transport Systems (C-ITSs) have emerged as key enablers of road safety, traffic efficiency, and situational awareness, relying on the continuous exchange of Cooperative Awareness Messages (CAMs) among V2X-enabled vehicles. These messages contain kinematic information, including position, speed, acceleration, and heading, and serve as the foundation of cooperative perception in modern vehicular environments. The integrity of this shared traffic picture is therefore paramount: a vehicle acting on falsified CAMs can make incorrect and potentially dangerous driving decisions, such as emergency braking or unsafe lane changes.

Standards, such as the IEEE 1609.2 WG [1] and European Telecommunications Standards Institute (ETSI) [6], secure Vehicular Communication (VC) systems. Security architectures for VC systems [24] provide security and privacy protection based on vehicular Public-Key Infrastructure (PKI), while preventing misuse of credentials [19]. However, they offer no protection against insider threats: compromised vehicles, provisioned with credentials, that deliberately transmit falsified kinematic data. Such falsification attacks can be particularly dangerous until the attacker is revoked [20], as their effects on surrounding vehicles can be immediate and severe, especially during dynamic scenarios such as lane changes or maneuvers [14, 15]. This motivates the need for MDSs that detect falsified data and trigger revocation [24, 25].

Traditional data-centric MDSs rely on plausibility checks and consistency thresholds, while recent Machine Learning (ML) solutions, including Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and transformer-based architectures, have demonstrated promising results on the VeReMi and VeReMi++ datasets [3, 5, 10, 11] and platooning benchmarks [16, 21]. However, these approaches operate in a supervised manner: a detector trained on a fixed set of known attacks cannot reliably detect falsification strategies that deviate from its training distribution.

On the other hand, (semi-)unsupervised solutions [2, 4, 12, 23] reduce this dependency on labeled attack data by learning the statistical structure of normal behavior and flagging deviations at inference time. However, SVMDformer [23] still requires a small number of labeled samples for the hardest attack class. Campos et al. [4] operate in a federated setting restricted to binary detection on the original VeReMi dataset, and all four approaches are limited to binary detection, leaving multi-class unsupervised misbehavior detection across heterogeneous attack taxonomies an open problem.

In this paper, we propose PAMPOS, a causal transformer-based framework that reframes misbehavior detection as an unsupervised anomaly detection problem. It is trained exclusively on benign vehicular trajectories from VeReMi++ [18], learning the predictive structure of normal mobility without exposure to any attack data. Deployed on a vehicle, at inference time, PAMPOS identifies misbehavior as a statistically significant deviation from the model's next-step kinematic prediction, requiring no attack labels and generalizing naturally to unseen attack variants.

*Equally contributing authors.

WiseML '26, Saarbrücken, Germany

ACM ISBN 979-8-4007-2705-4/2026/06
https://doi.org/10.1145/3811880.3815104

Contributions. (i) An unsupervised MDS framework that requires no attack-labeled training data, enabling detection of novel attacks not seen during training, evaluated across all 19 attack types in VeReMi++. (ii) A causal transformer-decoder architecture trained on benign-only trajectories (i.e., attack-agnostic) with a top-K normalized scoring mechanism that localizes anomalous behavior to specific kinematic features, informing on the performed attack family. (iii) An analysis of the limits of prediction-error-based detection, identifying constant position offset (A2) and eventual stop (A9) as open challenges due to their similarity to benign behavior.

Paper Organization. Sec. 2 reviews related work on misbehavior detection datasets, ML and Deep Learning (DL) approaches for VANETs, and transformer-based modeling. Sec. 3 presents the system and threat model. Sec. 4 describes the PAMPOS framework, including data preprocessing, model architecture, and anomaly scoring. Sec. 5 presents the performance evaluation, discusses cross-scenario generalization, and inference efficiency. Finally, Sec. 6 concludes with key findings.

2 Related Work

Misbehavior Detection Datasets. Van Der Heijden et al. [26] introduced VeReMi, the first publicly available dataset for the comparable evaluation of MDSs in VANETs, providing labeled CAMs across five position-falsification attack types and 3 attacker-density ratios. Kamel et al. [17] proposed F^{2}MD, an open-source C-ITS simulation framework built on OMNeT++ and VEINS for implementing attacks, evaluating detectors, and generating labeled datasets. Building on this, Kamel et al. [18] generated VeReMi++, extending the original dataset with a realistic sensor-error model and approximately twenty attack variations, encompassing Denial of Service (DoS), data replay, Sybil, and falsification attacks across 24-hour simulation periods at diverse traffic densities. VeReMi++ serves as the primary evaluation dataset in this work.

Misbehavior Detection Approaches. Classical ML techniques have demonstrated considerable efficacy in detecting anomalous vehicular behavior [3]. F^{2}MD [17] benchmarked Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), and LSTM classifiers, with LSTM achieving the highest accuracy, confirming the advantage of sequential modeling for kinematic misbehavior detection. Gyawali et al. [10] combined ML with Dempster-Shafer theory for reputation-based detection, while [5] used Angle of Arrival (AoA) and signal-strength features with k-Nearest Neighbor (k-NN) and Random Forest (RF) to detect position falsification in C-ITS.

DL methods have further improved generalization: Liu et al. [22] reduced False Positive (FP) rates using LSTM; Hsu et al. [11] achieved 0.95 accuracy with a hybrid CNN-LSTM on VeReMi++; Alladi et al. [2] reported 0.98 accuracy with an F1-score of 0.97 across all VeReMi++ attack categories; and Youness et al. [28] achieved 0.99 accuracy with an F1-score of 0.99 using a Bidirectional LSTM (BiLSTM) with domain-informed spatiotemporal features. All of these, however, require labeled attack samples during training, failing to generalize to novel attacks absent from the training distribution.

Giuliari et al. [8] studied the viability of transformer encoders for mobility modeling, outperforming LSTM baselines on pedestrian forecasting. In VANETs, Li et al. [21] proposed AttentionGuard, a multi-head transformer-encoder for platoon misbehavior detection, achieving an F1 of 0.95 and an AUC of 0.96 with 100ms decision latency. Kalogiannis et al. [16] extended this with AIMformer, incorporating vehicle-specific temporal positional encoding and a precision-focused loss function to mitigate FPs under class imbalance, with further optimization for edge inference via TFLite, ONNX, and TensorRT. Both systems operate as supervised binary classifiers evaluated on platooning-specific datasets.

Comparison with Existing Work. PAMPOS departs from the supervised paradigm adopted by the works above in two ways. Rather than training on labeled attack examples, it learns the predictive structure of normal mobility from benign-only trajectories, identifying misbehavior at inference time as a deviation from this learned prior without requiring any attack-labeled data. The closest works in this direction are DeepADV [2], SVMDformer [23], UltraADV [12], and Campos et al. [4], yet a direct numerical comparison with these methods is not feasible given their incompatible experimental setup. SVMDformer deploys to Roadside Units (RSUs) rather than on-vehicle, requiring a full 200-message sequence before scoring, roughly 20 seconds at a typical CAM rate, whereas PAMPOS scores after just 10 messages.

UltraADV similarly deploys to RSUs and selects its anomaly threshold using a test set containing artificially introduced attacks, making it semi-supervised rather than fully unsupervised; it also prioritizes deployment efficiency through knowledge distillation but does not address the interpretability of attack types. Campos et al. combine Variational Autoencoders (VAEs) and Gaussian Mixture Models (GMMs) within a Federated Learning (FL) framework on VeReMi under non-Independent and Identically Distributed (IID) conditions, in a setting that is incomparable to centralized detection.

All four are further limited to binary detection, whereas PAMPOS addresses multi-class unsupervised detection across the full 19-attack-type VeReMi++ benchmark without federated overhead. Its top-K feature attribution mechanism also localizes anomalies to specific kinematic dimensions, giving Misbehavior Authority (MA) actionable attack-type information beyond a binary flag, and PAMPOS demonstrates cross-scenario generalization between rush-hour and afternoon traffic, a property none of these works evaluate. Furthermore, unlike AttentionGuard [21] and AIMformer [16], which target small homogeneous platooning formations, PAMPOS operates across the broader, more heterogeneous V2X setting of VeReMi++, encompassing diverse sender populations and 19 attack types.

3 System and Threat Model

We consider a V2X-enabled urban vehicular network in which vehicles periodically broadcast CAMs containing their reported kinematic state, namely position, speed, acceleration, and heading, to their neighborhood. Vehicles receiving these messages, from surrounding senders, evaluate each sender's trajectory over time. We assume a threat model applicable to VCs [24], with vehicles possessing valid credentials that allow them to broadcast CAMs [19]. PAMPOS serves as a second line of defense, running on each vehicle and detecting insider threats posed by such compromised vehicles.

Concretely, attackers traveling on the road can manipulate CAMs at multiple levels, spanning three categories [18]: *position and speed falsification*, where reported values are manipulated via constant offsets or random perturbations (A1–A8); *behavioral attacks*, where

the reported trajectory follows a physically plausible but malicious pattern, such as eventual stop or disruptive mobility (A9–A10); and *protocol-level attacks*, including data replay, delayed message injection, and DoS flooding, with Sybil variants of each (A11–A19). We make no assumption about the attacker's knowledge of the detection mechanism, including attackers whose falsified trajectories remain physically consistent with normal driving—such as constant position offset (A2) and eventual stop (A9)—which represent a known open challenge discussed in Sec. 5.

4 Proposed Framework – PAMPOS

Data pre-processing. Our experiments are conducted on the VeReMi++ dataset [18], which comprises 38 scenario subsets spanning two distinct 2-hour traffic periods: 19 morning and 19 afternoon subsets. The scenarios cover a broad behavioral spectrum, from benign vehicular mobility to common kinematic falsification attacks targeting position and speed, as well as DoS scenarios.

Each scenario contains two message types: type 2 messages originating from the ego vehicle (i.e., where PAMPOS is deployed) and type 3 messages corresponding to CAMs received from surrounding senders. Each received CAM is aligned to the most recent ego state; messages whose temporal gap exceeds $2s$ are discarded to exclude delayed messages. The retained messages carry position, speed, acceleration, and heading, transformed into eight relative kinematic features as the delta between each CAM field and the ego state, grouped by sender identity and sorted by receive time.

All messages from a given sender are treated as a single continuous sequence. Since vehicles periodically leave and re-enter the ego's communication range, the resulting sequences contain temporal gaps; we split sequences at these gaps to prevent abrupt feature discontinuities from contaminating training and inflating benign prediction errors. Feature-wise mean and standard deviation are computed from all benign sequences across the full 38-scenario corpus and applied for z-score normalization, ensuring a consistent input scale throughout training and evaluation.

Training. To prevent data leakage, sequences are partitioned into training, validation, and test sets by sender identity, ensuring that all windows belonging to a given sender reside exclusively within one split. Each sequence is then segmented using a sliding window of length 10 and stride 1, with the model trained to predict the 11th step. A practical challenge arises from the acceleration features (Δacl_x, Δacl_y), which exhibit occasional large-magnitude spikes in benign data due to sudden braking or sharp turning maneuvers. Under Mean Squared Error (MSE), a single outlier with a residual of 10 incurs a squared penalty of 100, disproportionately dominating the gradient signal and biasing the model toward over-predicting acceleration magnitude.

To mitigate this, we adopt the Huber loss [13], which interpolates between MSE and Mean Absolute Error (MAE): it preserves the differentiable quadratic regime for well-behaved predictions while bounding the gradient contribution of outliers through a linear tail. Formally, the per-sample, per-feature loss is defined as:

$$L_\delta(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{if } |y - \hat{y}| \le \delta \\ \delta\,|y - \hat{y}| - \frac{1}{2}\delta^2 & \text{otherwise,} \end{cases} \tag{1}$$

where y is the true feature value, $\hat{y}$ is the model's prediction, and δ controls the transition between the quadratic and linear regimes. Setting $\delta = 1.0$ yields:

$$L(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{if } |y - \hat{y}| \le 1 \\ |y - \hat{y}| - \frac{1}{2} & \text{otherwise.} \end{cases} \tag{2}$$

The total training objective is the mean Huber loss over all N samples and $F = 8$ kinematic features:

$$\mathcal{L}_{\text{total}} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{F}\sum_{f=1}^{F} L\left(y_n^{(f)},\ \hat{y}_n^{(f)}\right). \tag{3}$$

Table 1: Transformer decoder architecture.

Parameter	Value
Architecture style	Pre-norm (GPT-2)
Hidden dimension (d_{model})	128
Attention heads	8 (key dim = 16)
Feed-forward dimension	256
Decoder blocks	3
Activation	GELU
Dropout	0.1
Positional encoding	Sinusoidal [27]
Total parameters	399,880

We use a causal decoder-only design (Fig. 1) with 3 pre-norm blocks, 8 attention heads, and a hidden dimension of 128 (~400K parameters), trained to predict the next trajectory step from a window of 10 consecutive relative-feature vectors. Each block follows the pre-norm layout, where layer normalization is applied before attention and feed-forward sublayers rather than after, providing more stable gradients during training. For positional encoding, we use the fixed sinusoidal functions of [27].

Model Evaluation. Our evaluation is twofold: first, we assess the ability to predict future values based on the sender's history, as smaller prediction errors enable better anomaly detection; second, we aggregate the top-K sender features with large errors to predict an attack using a threshold τ. Specifically, we compute the per-feature absolute errors as:

$$e_f = \left|y^{(f)} - \hat{y}^{(f)}\right| \quad f = 1, \ldots, 8 \tag{4}$$

Then, we normalize by the benign MAE of each feature, bringing all features to the same scale, enabling us to take the mean of the top-K largest errors as:

$$s = \frac{1}{K}\sum_{k=1}^{K}\tilde{e}_{(k)} \qquad \text{where } \tilde{e}_{(1)} \ge \tilde{e}_{(2)} \ge \cdots \ge \tilde{e}_{(8)} \tag{5}$$

Selecting from the top subset of features enables us to detect attacks that affect only some of them. For sender-level detection, all window scores from a sender are averaged:

$$\bar{s}_{\text{sender}} = \frac{1}{|\mathcal{W}|}\sum_{w\in\mathcal{W}} s_w, \tag{6}$$

and a sender is flagged as an attacker if $\bar{s}_{\text{sender}} > \tau$.

5 Performance Evaluation

5.1 Setup

The transformer-decoder was implemented using TensorFlow [9] and Keras [7]. We performed our analysis on an Ubuntu machine with an AMD Ryzen Threadripper PRO 5965WX with 24 physical

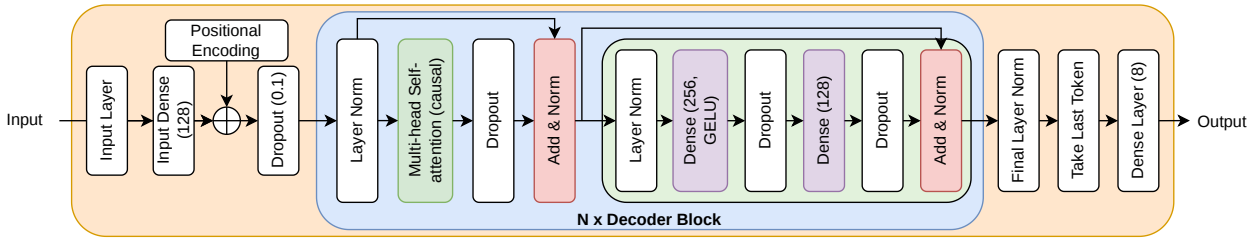


Figure 1: PAMPOS causal transformer-decoder architecture and parameters.

cores and 48 logical cores, 128 GB of available Random Access Memory (RAM), and an NVIDIA GeForce RTX 4090 with 24 GB of DDR5 memory. For training the model, we use a batch size of 512, an initial Learning Rate (LR) of 3×10^{-4} for the Adam optimizer with gradient clipping at $\|\nabla\|_2 \leq 1.0$, and Huber loss ($\delta = 1.0$) to reduce sensitivity to acceleration outliers.

The LR is halved after 4 epochs without improvement in validation loss (ReduceLROnPlateau), and training stops early after 8 stagnant epochs. The model is trained on the benign data from all 38 VeReMi++ scenarios (19 afternoon and 19 morning rush), yielding ~13M training windows from 2,735 unique benign senders split 70/15/15 by sender identity to prevent data leakage. Table 2 outlines all parameters used in the training and data setup.

The anomaly detection parameters used in this work are summarized in Table 3. We compute the anomaly score for each incoming message based on the 3 largest per-feature normalized absolute errors (Eq. 5), and then average the window scores per sender (Eq. 6) for vehicle-level classification. The threshold τ is set as a fixed percentile of the benign sender score distribution (Eq. 6) using only benign validation data, requiring no attack labels; the chosen percentile directly controls the False Positive Rate (FPR). During deployment, the operator selects the percentile based on their tolerable FPR. For evaluation, we report results at the 98th percentile ($\tau_{98} = 4.43$), which corresponds to a 2% FPR; the test set is completely unseen by the model.

Table 2: Training and data parameters.

Training		Data	
Parameter	**Value**	**Parameter**	**Value**
Batch size	512	Scenarios	38 (19 per period)
LR	3×10^{-4}	Attack types	19 (A1–A19)
Optimizer	Adam ($\|\nabla\|_2 \leq 1.0$)	Input features	8 relative kinematics
Loss function	Huber ($\delta = 1.0$)	Window size / stride	10 / 1
LR schedule	ReduceLROnPlateau	Training windows	~13M
LR reduce patience	4 epochs	Data split (train/val/test)	70/15/15 (by sender)
Early stopping patience	8 epochs	Min. sequence length	15 messages
		Temporal gap threshold	2*s*

5.2 Trajectory Prediction Analysis

We assess the quality of transformer-decoder trajectory prediction, as the anomaly-scoring mechanism relies on the magnitudes of per-feature prediction residuals to discriminate between falsifying senders and benign ones. Fig. 2 presents scatter plots of predicted vs. true values for each kinematic feature family, where the diagonal $y = x$ denotes perfect prediction. The model achieves alignment

Table 3: Anomaly detection parameters.

Parameter	Value
Anomaly score	Mean-of-top-K
Top-K features	3 (of 8)
Sender aggregation	Mean of window scores
Deployment threshold (τ_{98})	4.43
Threshold percentile	98th
Calibration set	411 validation senders

for position (Fig. 2a), with moderate residuals in speed (Fig. 2b) and heading (Fig. 2d) attributable to abrupt lane changes. Acceleration (Fig. 2c) exhibits the largest errors, as it is the second derivative of position and thus inherently noisier, with sudden driver inputs producing large-magnitude changes that the model cannot anticipate from prior context alone. Overall, prediction fidelity is sufficient to yield meaningful separation between benign and attack residuals, as demonstrated in the following detection analysis.

5.3 Misbehavior Detection Analysis

Table 4 shows the top-3 feature selection frequency across attack types. For each window, the anomaly scorer computes the per-feature normalized absolute error, i.e., the ratio of the prediction error to the benign baseline for that feature. Each cell in the table reports the fraction of windows where that feature appears among the selected top-3. i.e., if the selection were purely random, every feature would appear with frequency $3/8 = 0.375$.

Benign senders (A0) approximate this uniform baseline (0.28 to 0.47), confirming that no single feature consistently dominates under normal conditions. In contrast, attack types show strong concentration on the features they distort. Position attacks (A1, A3, A4) select Δx and Δy in over 86% of windows, while speed attacks (A7, A8) shift selection toward Δspd_x and Δspd_y (up to 0.79). Notably, acceleration features (Δacl) are rarely selected (as low as 0.00 for A14 and A18) because position or speed attacks produce errors on the other features that are larger than acceleration noise.

Specifically, A2 (Constant Position Offset) exhibits near-uniform selection that closely resembles benign traffic. Its anomaly signal is diffuse rather than concentrated in specific features, as the small constant offset keeps the falsified trajectory behaviorally consistent with normal driving. Choosing a different value of K would not improve the detector. On the other hand, A9 (Eventual Stop), A13 (DoS), A16 (Grid Sybil), and A17 (Data Replay Sybil) show moderate position-feature concentration (0.70–0.79 for Δx, Δy), which could suggest K=2; conversely, speed attacks (A7, A8) would suggest

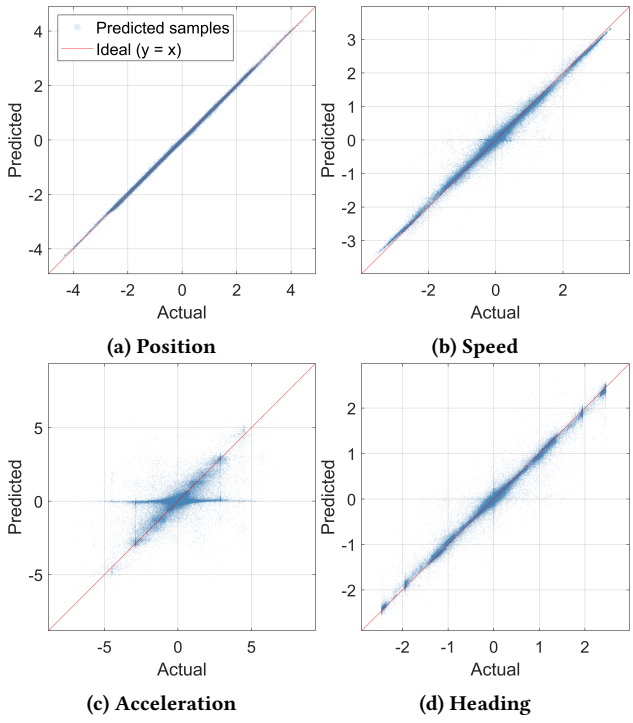


Figure 2: Feature prediction error.

K=4, though this would dilute the scores for position-concentrated attacks (A1, A3). K=3 provides a conservative, but robust default. Nonetheless, top-K co-selection patterns offer insights into the attack family: speed co-selection identifies speed attacks (A5, A7, A8), while heading co-selection identifies disruptive attacks.

Table 4: Top-3 feature selection frequency per attack type.

		Δx	Δy	Δspd_x	Δspd_y	Δacl_x	Δacl_y	Δhed_x	Δhed_y
A0	Benign	0.43	0.47	0.32	0.41	0.28	0.35	0.34	0.40
A1	Const. Position	0.94	0.93	0.14	0.20	0.13	0.10	0.26	0.30
A2	Const. Pos. Offset	0.47	0.57	0.26	0.45	0.23	0.28	0.26	0.49
A3	Random Position	0.98	0.97	0.17	0.22	0.10	0.11	0.25	0.20
A4	Random Pos. Offset	0.91	0.86	0.17	0.19	0.21	0.25	0.18	0.23
A5	Const. Speed	0.85	0.80	0.53	0.40	0.06	0.04	0.16	0.17
A6	Const. Speed Offset	0.96	0.84	0.29	0.20	0.20	0.14	0.17	0.19
A7	Random Speed	0.79	0.60	0.77	0.64	0.02	0.02	0.08	0.08
A8	Random Speed Offset	0.68	0.46	0.79	0.70	0.09	0.07	0.12	0.09
A9	Eventual Stop	0.70	0.74	0.23	0.30	0.13	0.18	0.35	0.38
A10	Disruptive	0.79	0.80	0.25	0.30	0.02	0.06	0.40	0.39
A11	Data Replay	0.73	0.76	0.26	0.31	0.09	0.15	0.36	0.34
A12	Delayed Messages	0.85	0.91	0.16	0.20	0.07	0.12	0.27	0.41
A13	DoS	0.71	0.76	0.22	0.28	0.28	0.28	0.20	0.25
A14	DoS Random	0.95	0.91	0.53	0.29	0.00	0.00	0.20	0.11
A15	DoS Disruptive	0.79	0.81	0.27	0.28	0.03	0.04	0.41	0.38
A16	Grid Sybil	0.74	0.71	0.32	0.34	0.15	0.15	0.33	0.25
A17	Data Replay Sybil	0.74	0.79	0.24	0.31	0.10	0.13	0.35	0.34
A18	DoS Random Sybil	0.95	0.91	0.49	0.29	0.00	0.00	0.21	0.14
A19	DoS Disruptive Sybil	0.86	0.85	0.24	0.22	0.03	0.03	0.42	0.34

Fig. 3 presents the Receiver Operating Characteristic (ROC) curves for all nineteen attack scenarios, with the legend sorted by AUC. The majority of attacks achieve AUC values between 0.93 and 0.98, with consistently high True Positive Rate (TPR) at low FPR. The notable exceptions are A13, A9, and A2, which exhibit progressively degraded performance. A9 and A2 display qualitatively distinct curve characteristics: A9 maintains a higher TPR than A2 up to a FPR of 0.38, beyond which the performance reverses.

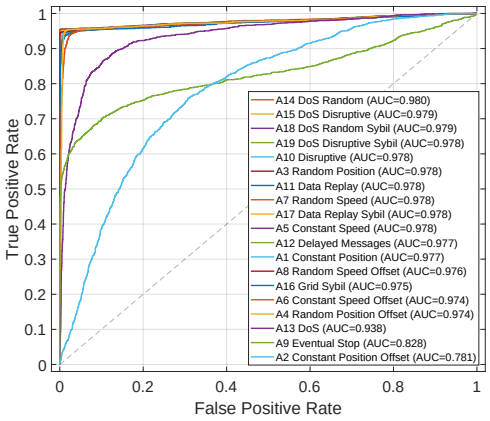


Figure 3: ROC curves for each attack scenario.

The overall detection metrics are reported in Table 5. DoS attacks A14 to A19 are detected despite the absence of explicit temporal features because their CAMs carry replayed or kinematically altered data, producing large prediction residuals regardless of transmission frequency. The simple DoS (A13) is a partial exception: its messages carry valid kinematics with only the frequency altered, compressing the 10-message window to milliseconds rather than the expected ∼1 s. The resulting near-zero per-step deltas, which the model expects for normal speeds, yield a moderate anomaly signal (F1 = 0.75), confirming that the learned temporal structure implicitly encodes expected transmission intervals.

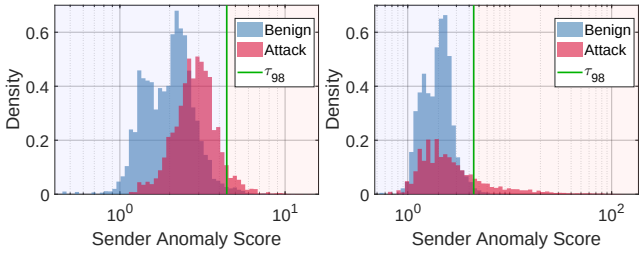
The most challenging cases are A2 (Constant Position Offset, F1 = 0.19) and A9 (Eventual Stop, F1 = 0.73). A2 small constant shift preserves all higher-order kinematics (speed, acceleration, heading) and trajectory shape, keeping the anomaly score distribution overlapping with the benign one (Fig. 4), while the top-3 selection frequencies (0.23 to 0.57) are indistinguishable from the benign baseline (0.28 to 0.47, Table 4), leaving no discriminative feature. A9 performs better, though early deceleration is indistinguishable from normal braking; the anomaly signal emerges only once the vehicle remains stationary. Despite this, our solution has comparable overall F1 scores with supervised solutions that require labeled data during training (recall Sec. 2).

5.4 Cross-Scenario Generalization

To test generalization, we train a model exclusively on afternoon (14:00 to 16:00) benign data and evaluate on morning rush hour (07:00 to 09:00) attacks, featuring different traffic densities and vehicle populations. The threshold τ_{96} (producing the optimal F1 score) is selected with the same method, as described in Sec. 5.1. The afternoon-trained model achieves a mean AUC of 0.95 and a mean F1 of 0.87 across all 19 attacks. 16 of those retain F1 $\geq$ 0.91, with the largest drops confined to A2 (F1 = 0.14), A9 (F1 = 0.70), and A13 (F1 = 0.74). Training in the reverse direction (morning rush

Table 5: Attack detection metrics for each attack category.

Metric	Position				Speed				Eventual Stop	Other			DoS			Sybil			
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
Accuracy	0.97	0.69	0.97	0.97	0.97	0.96	0.97	0.97	0.86	0.97	0.97	0.97	0.86	0.97	0.97	0.96	0.97	0.97	0.97
Precision	0.94	0.63	0.95	0.94	0.94	0.94	0.95	0.95	0.92	0.95	0.95	0.94	0.92	0.94	0.94	0.94	0.95	0.94	0.94
Recall	0.95	0.11	0.95	0.95	0.95	0.94	0.95	0.95	0.61	0.95	0.95	0.95	0.64	0.96	0.96	0.95	0.95	0.96	0.96
F1	0.95	0.19	0.95	0.95	0.95	0.94	0.95	0.95	0.73	0.95	0.95	0.95	0.75	0.95	0.95	0.94	0.95	0.95	0.95



(a) Const. Position Offset **(b) Eventual Stop**

Figure 4: Anomaly scores.

to afternoon) yields nearly identical aggregate performance (mean AUC = 0.95, mean F1 = 0.86), confirming symmetric generalization. This is expected: relative kinematic features are invariant to absolute position, road topology, and traffic density, so patterns learned from afternoon traffic transfer directly to rush hour conditions.

6 Conclusion

We presented PAMPOS, an unsupervised MDS for V2X networks based on a causal transformer-decoder trained only on benign trajectories from VeReMi++. PAMPOS identifies falsifying senders as those whose reported kinematic sequences deviate significantly from a learned mobility prior, without requiring any attack-labeled training data. Evaluated across all 19 attack types in VeReMi++, PAMPOS achieves AUC values of up to 0.98 and F1-scores of up to 0.95 for the majority of attack categories. The top-K normalized scoring mechanism successfully localizes falsification to the specific kinematic features each attack distorts, enabling detection of attacks that corrupt only a subset of the reported state. Bidirectional cross-scenario evaluation confirms symmetric generalization between afternoon and morning rush hour traffic, owing to the scenario-invariant relative features. The constant position offset (A2) and eventual stop (A9) attacks remain open challenges, as their falsified trajectories are behaviorally indistinguishable from benign mobility under the learned model; addressing these cases requires complementary strategies such as cross-sender consistency checks or map-aware validation. Future work will investigate augmented scoring strategies to address these hard cases, an ablation study on all parameters, incorporate temporal frequency features to strengthen DoS detection, and explore model compression for deployment on resource-constrained devices.

Acknowledgments

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

References

[1] 1609_WG - V2X Communications Working Group . 2022. IEEE Standard for Wireless Access in Vehicular Environments–Security Services for Application and Management Messages. *IEEE Vehicular Technology Society* (Jan. 2022).
[2] Alladi et al. 2021. DeepADV: A Deep Neural Network Framework for Anomaly Detection in VANETs. *IEEE TVT* (2021).
[3] Abdelwahab Boualouache and Thomas Engel. 2023. A Survey on Machine Learning-Based Misbehavior Detection Systems for 5G and Beyond Vehicular Networks. *IEEE COMST* (2023).
[4] Campos et al. 2025. Federated Learning for Misbehaviour Detection with Variational Autoencoders and Gaussian Mixture Models. *International Journal of Information Security* (2025).
[5] Secil Ercan, Marwane Ayaida, and Nadhir Messai. 2022. Misbehavior Detection for Position Falsification Attacks in VANETs Using Machine Learning. *IEEE Access* 10 (2022).
[6] ETSI. 2016. Intelligent Transport Systems (ITS); Security; ITS Communications Security Architecture and Security Management.
[7] François Chollet. 2026. *Keras*. https://keras.io/ Accessed: Feb 2026.
[8] Giuliari et al. 2021. Transformer Networks for Trajectory Forecasting. In *ICPR*.
[9] Google. 2026. *TensorFlow*. https://www.tensorflow.org Accessed: Feb 2026.
[10] Sohan Gyawali, Yi Qian, and Rose Qingyang Hu. 2020. Machine Learning and Reputation Based Misbehavior Detection in Vehicular Communication Networks. *IEEE TVT* (2020).
[11] Hsu et al. 2022. A Deep Learning-Based Integrated Algorithm for Misbehavior Detection System in VANETs. In *ACM ICEA*.
[12] Huang et al. 2025. UltraADV: An Unsupervised Deep Learning Lightweight Framework for Anomaly Detection in V2X. *IEEE IoT-J* (2025).
[13] Peter J. Huber. 1964. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics* (1964).
[14] Konstantinos Kalogiannis, Andreas Henriksson, and Panos Papadimitratos. 2023. Vulnerability analysis of vehicular coordinated maneuvers. In *IEEE EuroS&PW*.
[15] Kalogiannis et al. 2022. Attack impact and misbehavior detection in vehicular platoons. In *ACM WiSec*.
[16] Kalogiannis et al. 2025. Attention in Motion: Secure Platooning via Transformer-based Misbehavior Detection. *arXiv preprint arXiv:2512.15503* (2025).
[17] Kamel et al. 2020. Simulation Framework for Misbehavior Detection in Vehicular Networks. *IEEE TVT* (2020).
[18] Kamel et al. 2020. Veremi extension: A dataset for comparable evaluation of misbehavior detection in vanets. In *IEEE ICC*.
[19] M. Khodaei et al. 2018. SECMACE: Scalable and Robust Identity and Credential Management Infrastructure in Vehicular Communication Systems. *IEEE T-ITS* (2018).
[20] Mohammad Khodaei and Panos Papadimitratos. 2021. Scalable & Resilient Vehicle-Centric Certificate Revocation List Distribution in Vehicular Communication Systems. *IEEE TMC* (July 2021).
[21] Li et al. 2025. AttentionGuard: Transformer-based Misbehavior Detection for Secure Vehicular Platoons. In *ACM WiseML*.
[22] Xiangyu Liu. 2022. Misbehavior Detection based on Deep Learning for VANETs. In *CNCIT*.
[23] Liu et al. 2023. SVMDformer: A semi-supervised vehicular misbehavior detection framework based on transformer in iov. In *IEEE ICDCS*.
[24] Papadimitratos et al. 2008. Secure Vehicular Communication Systems: Design and Architecture. *IEEE Comm. Mag.* (2008).
[25] Raya et al. 2008. On Data-Centric Trust Establishment in Ephemeral Ad hoc Networks. In *IEEE INFOCOM*.
[26] Rens W Van Der Heijden, Thomas Lukaseder, and Frank Kargl. 2018. Veremi: A dataset for comparable evaluation of misbehavior detection in vanets. In *EAI SecureComm*.
[27] Vaswani et al. 2017. Attention is all you need. *NIPS* (2017).
[28] Youness et al. 2025. VeMisNet: Enhanced Feature Engineering for Deep Learning-Based Misbehavior Detection in Vehicular Ad Hoc Networks. *JSAN* (2025).