# Multi-Head Attention Machine Learning for Fault Classification in Mixed Autonomous and Human-Driven Vehicle Platoons

Theodore Wu Satvick Acharya Abdelrahman Khalil Ahmad F. Aljanaideh
Mohammad Al Janaideh Deepa Kundur

***Abstract*— Connected Autonomous Vehicle (CAV) platoons have been extensively studied to protect against cyber and physical vulnerabilities. Faults can occur in all layers of the platoon system or could be introduced by impaired human drivers. Since different types of faults may require different fault resolution methods, identifying the fault class facilitates the selection of the best mitigation strategy. This paper introduces a Multi-Head Attention Machine Learning (MHA-ML) approach to classify a set of five different faults and abnormalities in mixed autonomous and human-driven vehicle platoons. Autonomous vehicles can face actuator faults, False Data Injection (FDI) attacks, and Denial-of-Service (DoS) attacks, while abnormalities such as drunk or distracted human drivers could occur. MHA-ML is developed to identify faulty vehicle behavior over long sequences of sensor measurements. MHA-ML is trained on a mixed platoon simulation model and then tested on mobile laboratory robots. The experiment classifies the five fault categories with 90% accuracy and outperforms a baseline recurrent neural network approach.**

## I. Introduction

Considered the new generation of autonomous vehicles, the Connected Autonomous Vehicle (CAV) platoon is an emerging cyber-physical technology that integrates self-driving vehicles with wireless communication networks. This integration allows vehicles to communicate with their surroundings, which allows for a higher level of vehicle awareness and supervision. However, CAV platoons are open for any vehicle on different levels of autonomy to join [1], [2] and possess a wide range of physical and cyber-communication components, as illustrated in Figure 1. This renders CAV platoons highly vulnerable to a wide range of physical and cyber faults, as well as malicious cyber-attacks. Moreover, current autonomous vehicles and human-driven vehicles are not supplied with adequate technology to communicate with CAVs [3], [4]. As CAV platoons enter the market, platoons will be forced to coexist alongside such vehicles, which presents an additional challenge since cooperation with noncommunicating vehicles is impossible and unpredictable human driver behavior can resemble fault-like sensor signals [5]. The fault and risk range in CAV platoons becomes wider and more severe when impaired drivers, such as intoxicated or distracted drivers, are introduced within the platoon. With many scenarios in which faults can occur, the real-world viability of CAV platoons hinges on whether safety and stability within the platoon can be ensured.

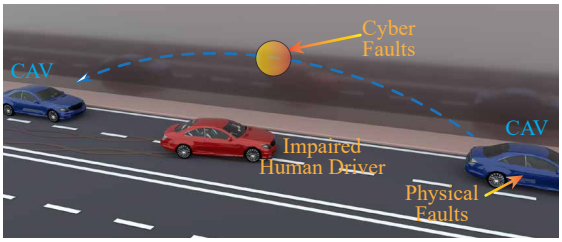



Fig. 1: An illustration of an impaired driver joining a faulty CAV platoon.

Different risk mitigation techniques were introduced in the literature to enhance CAV safety and security, as will be further discussed in Section II. The literature survey showed that mitigating these faults, attacks, and abnormalities due to either CAV issues or uncooperative vehicles (i.e., human-driven) is faster and more reliable if the issue class is known [6], [7]. For example, if a communication channel is jammed or attacked, the risk can be mitigated by switching to a backup communication channel. While extensive research has been conducted to detect such faults and abnormalities, it remains challenging to distinguish the specific fault class that occurred. Further, most fault and risk mitigation techniques are only designed to mitigate a specific fault or abnormality class. Thus, there is an apparent need for a fault classification mechanism to bridge the gap between fault detection and the selection of the appropriate risk mitigation technique.

In this paper, we propose a method for classifying the fault and abnormalities class using Multi-Head Attention Machine Learning (MHA-ML). MHA-ML is able to augment the awareness of the machine learning network by recognizing long-range dependencies in the input sequence and learning to pay more attention to relevant information [8]. The proposed technique only uses a sequence of platoon velocities as inputs to the classifier and is thus insensitive to the platoon model dynamics. We investigate five major faults and abnormalities classes in CAV platoons: (i) Actuator fault, (ii) False Data Injection (FDI), (iii) Denial-of-Service (DoS), (iv) Drunken drivers, and (v) Distracted drivers. These were selected for their full coverage of physical faults, cyber-communication attacks, and human driver abnormalities.

T. Wu, S. Acharya, and D. Kundur are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada theodore.wu@mail.utoronto.ca, satvick.acharya@mail.utoronto.ca, dkundur@ece.utoronto.ca

A. Khalil and M. Al Janaideh are with the Department of Mechanical Engineering, Memorial University, St. John's, NL, Canada amkhalil@mun.ca, maljanaideh@mun.ca

M. Al Janaideh is with the School of Engineering, University of Guelph, Guelph, ON N1G 2W1, Canada maljanai@uoguelph.ca

A. Aljanaideh is with Bentley University, Waltham, MA, USA. aaljanaideh@bentley.edu

## II. Related Work

The pairing of fault detection with fault mitigation techniques is a recent trend in improving CAV safety [9]. Typical methods are analytical or software-based and aim to restore normal system behavior after abnormal behavior is induced by a specific target fault class. In [10], a reliable observer-based detector was used to detect and mitigate actuator and sensor faults. However, no algorithm was given to distinguish the fault characteristics nor whether the sensor attack severity lies within the bendable range. In [11], a novel distributed observer is proposed to achieve consensus among estimated agent states when a disturbance occurs, although the specific type of disturbance is not identified. The authors of [12] combined machine learning techniques with a Kalman filter to detect faults in CAV platoons but are unable to distinguish which subsystem causes the fault. In [13], DoS attacks were detected and estimated using sliding mode observers, although the identification of the attack as a DoS was known a priori. The authors of [14] similarly assumed a known fault class and designed an adaptive synchronization-based control algorithm for communication time delays and FDI. For actuator faults, [15], [16] and [17] follow similar trends.

Many of the existing fault mitigation techniques are developed for specific fault classes but may not generalize to additional fault types. On the other hand, fault detection methods can flag many types of abnormal system behavior but do not often localize the source of the fault [9]. This motivates the need for a fault classification mechanism – a critical task in the fault management process for cyber-physical systems [18] – that allows the detection system to select appropriate mitigation methods. However, there is a gap in the CAV literature on such classification methods.

Fault classification has been explored in other systems with limited fault range and severity. In [19], [20], CNN and LSTM approaches were used to classify faulty transmissions and cyberattacks in autonomous and internet-connected vehicles. However, CNNs showed poor feature localization and LSTMs showed difficulties representing long sequence data as a single vector. In [21], multinomial logistic regression was used to categorize bearing faults. In [22], an SVM was used to distinguish between normal and six faulty sensor behavior types, although signal data needed to be carefully preprocessed to extract useful features for the classifier. Other methods such as random forest classifiers and XGBoost were used in [23], [24], but the quantity of decision trees slows down real-time predictions in practice. A fault classification algorithm that can quickly process large amounts of sensor data and distinguish a wide range of faults and abnormalities has yet to be developed for the CAV domain.

Machine learning is commonly used in fault classification for its ability to extract patterns from large amounts of historical data. The concept of attention is particularly promising, as it allows neural networks to prioritize specific timesteps of sequential data, such as that emitted by CAV sensors, in their decision-making. A CNN with multi-head attention is used in [25] to perform fault classification in industrial systems. The network uses convolutional layer features as the attention mechanism inputs and uses a softmax function to generate fault probabilities. Self-attention is used in [26]–[28] and is shown to consistently improve accuracy while enabling parallel processing of sequence data. The multi-head attention CNN technique in [29] improved accuracy, feature extraction and feature selection for human activity recognition. In the context of autonomous vehicles, [30]–[32] have examined the use of attention for autonomous vehicle motion forecasting and trajectory prediction.

## III. Healthy Platoon Model

Although the proposed technique does not require knowledge of platoon dynamics, a simulation model is required to generate training data. This paper considers the platoon longitudinal drive only. The lateral drive is kept for future work. This section considers the healthy conditions of the platoon, whereas Section IV modifies the healthy model with the five fault classes considered in this work.

### A. CAV Model

We model the longitudinal drive of CAVs as brushless electric vehicles with an internal PI controller representing the cruise control law. Following [33], the CAV model is given by the following transfer function

$$\frac{V_i(s)}{V_i^*(s)} = \frac{\delta_i s + \epsilon_i}{s^3 + \alpha_i s^2 + \beta_i s + \gamma_i}, \tag{1}$$

where for both first and third vehicles $i \in \{1, 3\}$, $V_i$ is the velocity of the $i$th vehicle in the frequency domain, and $V_i^*$ is the desired velocity of the $i$th vehicle in the frequency domain. This model was constructed in [33] using the bond graph approach to create more realistic characteristics. The model parameters are configured for the first and third vehicles $i \in \{1, 3\}$ as $\alpha_i = 72.01, \beta_i = 117.9, \gamma_i = \epsilon_i = 46.72$, and $\delta_i = 28.03$.

### B. Human-driven

For the second vehicle, we adopt the human Intelligent Driver Model (IDM) in [34] as an example of healthy driver behavior. The healthy IDM model is given by

$$a_2(t, v_2) = a_{\max}\left[1 - \left(\frac{v_2(t)}{v_2^*(t)}\right)^{\lambda} - \left(\frac{s_2^*(t, v_2)}{s_2(t)}\right)^2\right], \tag{2}$$

where $a_2$ is the acceleration of the human-driven vehicle (second vehicle in the platoon), $a_{\max} = 1m/s^2$ is the

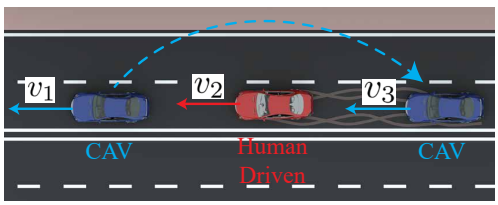



Fig. 2: Three vehicles mixed CAV and human-driven vehicle platoon. The first and third vehicles are CAVs, while the second vehicle is human-driven.

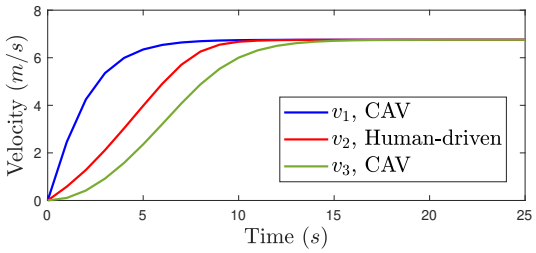



Fig. 3: Healthy response of the three vehicles platoon in Figure 2, where the first and third vehicles are CAVs and the second is human-driven. Note that human drivers have a slower response than CAVs.

maximum acceleration, $v_2$ and $v_2^*$ are the measured and desired velocities of the second vehicle, respectively. $\lambda$ is a constant that controls the human response speed. For a healthy human driver, the response constant is set to $\lambda = 8$. $s_2$ is the actual spacing distance between the first and second vehicles, and $s_2^*$ is the desired spacing distance given by

$$s_2^*(t, v_2) = s_0 + \max\left(0, v_2(t)T + \frac{v_2(t)\Delta v_2(t)}{2\sqrt{b^* a^*}}\right), \quad (3)$$

where $s_0 = 2m$ is the minimum spacing distance, $T = 1.5s$ is the desired time headway to the next vehicle, $\Delta v_2(t) = v_2(t) - v_1(t)$, $a^* = 1m/s^2$ is the maximum acceleration, and $b^* = 3m/s^2$ is the comfortable braking deceleration.

Figure 3 shows the healthy response of the platoon in Figure 2. The platoon was constructed such that the first and third vehicles follow the model in (1), and the second vehicle follows the model (2)-(3). The second vehicle follows the velocity of the first vehicle, while the third vehicle follows the velocity average of the first two vehicles as a simple approach to establish platoon velocity consensus when faults are possible [35]. Note that the human driver has a slower response by nature compared with CAVs.

## IV. Fault Models

This section alters the healthy platoon model introduced in Section III with the five fault classes investigated in this paper. Only one fault is assumed to occur at the same time. We leave the case of multiple faults for future work.

### A. CAV Faults

*1) Actuator Fault:* We consider a loss of effectiveness fault in the third vehicle's motor. This fault occurs via severe operating conditions of the vehicle's brushless motor, including high magnetic force and different weather conditions [36]. The loss of effectiveness fault results in a lower response amplitude. Therefore, we drop the transfer function amplitude by altering the parameter $\epsilon_3 = 41$ in the CAV model in (1), which lowers the actuator effectiveness to almost $80\%$.

*2) FDI Attack:* FDI attacks refer to falsifying the information transmitted through communication channels [14]. We model this fault as noise injected in the first vehicle's velocity that is received by the third vehicle. That is, the first vehicle's velocity remains healthy, but the velocity used in controlling the third vehicle is altered as $\tilde{v}_1(t) = v_1(t) + \eta_{\text{FDI}}(t)$, where $\tilde{v}_1$ is the velocity of the first vehicle received by the third vehicle, and $\eta_{\text{FDI}}$ is injected bounded white noise.

*3) DoS Attack:* DoS attacks occur when the communication channel is kept busy, which results in the information being transmitted late [14]. We model this fault as a time-variant communication delay in the velocity of the first vehicle that is received by the third vehicle under a no packet loss assumption. We alter the velocity of the first vehicle as $\tilde{v}_1(t) = v_1(t - \tau_{\text{delay}}(t))$, where $\tau_{\text{delay}}$ is a normally distributed variable time delay that captures the communication latency.

### B. Impaired Drivers

*1) Distracted Drivers:* The main feature of distracted drivers is their delayed response to stimuli [37]. Distracted drivers are less severe than drunk drivers but are much more common on real-world roadways. We model this abnormality as a delay in both the response $\lambda = 5$ as well as in tracking the velocity of the front vehicle $v_2^*(t) = v_2^*(t - \tau_{\text{distracted}}(t))$, where $\tau_{\text{distracted}}(t)$ is a normally distributed variable time delay that captures the delay in tracking the front vehicle.

*2) Drunk Drivers:* Following the report published in [38], a moderately drunk driver has the effects of (i) decline in visual perception, (ii) reduced coordination, (iii) increased latency in tracking moving objects, and (iv) decline in ability to multitask and respond to emergencies. We model these effects as (i) $\tilde{s}_2^*(t) = s_2^*(t) + \eta_{s^*}(t)$, (ii) $\tilde{s}_2(t) = s_2(t) + \eta_s(t)$, (iii) $\tilde{v}_1(t) = v_1(t - \tau_{\text{drunk}})$, and (iv) $\lambda = 3$, where $\tilde{s}_2^*, \tilde{s}_2, \tilde{v}_1$ refer to the corrupted desired spacing distance, actual spacing distance, and first vehicle's velocity, respectively. $\eta_{s^*}, \eta_s$ are bounded white noise and $\tau_{\text{drunk}} = 2s$ captures the constant time delay in tracking the front vehicle.

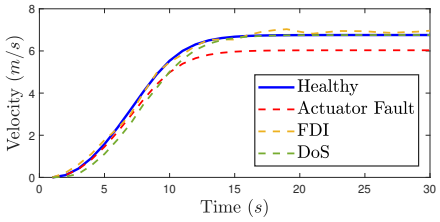
Figure 4 shows the healthy platoon responses from Section III after altering them with the faults and abnormalities introduced in Section IV. Figure 4a shows the third vehicle's response after altering it with the three CAV faults, and Figure 4b shows the human driver vehicle's velocity after altering it with the two abnormalities.

## V. Multi-head Attention

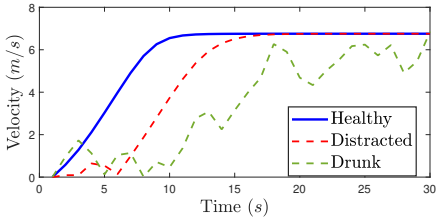
In this paper, we implement Multi-Head Attention Machine Learning (MHA-ML) [8] given its established success for long sequence processing. At its core, MHA-ML consists of a stack of parallel computations of scaled dot-product attention. Each attention computation results in an independent output encoding that is then aggregated across stacks.

For the discrete-time sample $k \in \{0, \ldots, n\}$, where $n$ is the total number of samples, let the platoon velocities be collected in the vector $\mathbf{v}(k) = \begin{bmatrix} v_1(k) & v_2(k) & v_3(k) \end{bmatrix}$. The goal is to identify the fault class using platoon velocities only. MHA-ML uses the concept of scaled-dot-product attention, with the function given by

$$\mathcal{A}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (4)$$




(a)




(b)

Fig. 4: Visualization of faulty vehicle responses under each fault class. (a) Third vehicle response $v_3$ in Figure 3 after emulating the CAV faults introduced in Section IV-A, and (b) Second vehicle velocity $v_2$ in Figure 3 after emulating the impaired driver effects introduced in Section IV-B. Non-faulting vehicles are omitted for visualization clarity.

where $Q \in \mathbb{R}^{n \times d_k}$ represents an arbitrary query, $K \in \mathbb{R}^{n \times d_k}$ represents an arbitrary key, and $V \in \mathbb{R}^{n \times d_v}$ represents an arbitrary value. $Q, K, V$ are matrices that correspond to different projections of a common input matrix $X = \begin{bmatrix} \mathcal{T}(\mathbf{v}(0)) & \dots & \mathcal{T}(\mathbf{v}(n)) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{n \times d_{\text{model}}}$, with $\mathcal{T}(\mathbf{v}(i)) = \mathbf{v}(i)W_E + \mathbf{p}(i)$ being the transformation performed on the platoon velocities for time $i$. $W_E \in \mathbb{R}^{3 \times d_{\text{model}}}$ denotes a weight matrix mapping the platoon velocities to the model hidden size and $\mathbf{p}(i) \in \mathbb{R}^{d_{\text{model}}}$ is a sinusoidal signal with element $j \in \{1 \dots d_{\text{model}}\}$ of the vector defined as

$$\mathbf{p}(i)_j = \begin{cases} \sin\left(\frac{i}{10000}^{2j/d_{\text{model}}}\right) & \text{when } j \text{ is even,} \\ \cos\left(\frac{i}{10000}^{2j/d_{\text{model}}}\right) & \text{when } j \text{ is odd.} \end{cases} \quad (5)$$

Since $Q, K, V$ are projections of $X$, we have $Q = XW_Q$, $K = XW_K$, and $V = XW_V$, where $W_Q \in \mathbb{R}^{d_{\text{model}} \times d_k}, W_K \in \mathbb{R}^{d_{\text{model}} \times d_k}, W_V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ are different weight matrices. The scalars $d_k$ and $d_v$ denote the key and value projection sizes, respectively. $d_{\text{model}}$ denotes the hidden size of the MHA-ML model. We consider $d_{\text{model}}$, $d_k$ and $d_v$ to be architectural hyperparameters that can be tuned to control the size of the MHA-ML model.

We can also interpret $Q, K, V$ as a stack of query, key, and value vectors for different sequence steps $k \in \{0 \dots n\}$. Under this paradigm, the softmax function used in (4) computes a set of attention probabilities over the entire time step sequence by comparing the dot-product similarity between each pair of query and key vectors in $Q$ and $K$. These probabilities are used to assign weights to value vectors at corresponding time steps in $V$, where the vectors at time steps with higher probability are more significant to the MHA-ML model.

The concept of multi-head attention extends scaled-dot-product attention by computing parallel stacks of scaled dot-product attention with different $Q, K, V$ matrices for each head. Each attention head computes an independent output, where all outputs are then concatenated and transformed linearly to restore the model dimension $d_{\text{model}}$. This feature provides the model with additional flexibility to prioritize different elements of the input sequence depending on the $Q, K, V$ projections learned by each head.

The proposed network architecture is shown in Figure 5. We first map the velocity measurements for each time-step $\mathbf{v}(1) \dots \mathbf{v}(k)$ to a higher-dimensionality representation through a velocity embedding layer. We implement this embedding as a single fully-connected layer. Note that the velocity embedding size is synonymous with the hidden size of the MHA-ML model, $d_{\text{model}}$. Next, the sinusoidal signal in (5) is summed with the velocity embeddings to inject sequential information, as introduced in [8]. This sinusoidal signal allows the model to learn the progression of the input sequence. The inclusion of this sinusoidal is necessary since MHA-ML avoids slow sequential processing by removing recurrent neural networks (RNNs) from the architecture. The sinusoid-augmented velocity embeddings are then passed to a multi-head attention layer and residual block. The attention layer identifies the most important time steps of the input sequence, while the residual connection helps with model stability and preserves information from the attention layer input. A feedforward network further transforms the attention layer's output. Specifics on the residual connections and feedforward network are given in [8].

We then apply average pooling across the time dimension of the feedforward network's output state to extract a sequence length-independent representation. Finally, a fully connected layer with a softmax output function is applied to calculate fault class probabilities. The model proposes the fault class with the highest predicted probability as the fault associated with the input velocity signal.

## VI. Model Training

In this section, we introduce our data generation process, our preprocessing techniques, and our model parameter optimization process.

### A. Data Generation

To train the machine learning network, the platoon model in Section III was constructed and each fault or abnormality in Section IV was implemented individually in a separate run. The desired velocity of the platoon was set to change randomly every 30 seconds. 5000 runs were recorded, with 1000 runs per fault or abnormality class. Each run is 500 seconds long with a sampling time of 1 second. The three vehicles' velocities were recorded in each run.

### B. Data Preprocessing

The data was normalized prior to training to improve stability of the training process. We applied min-max normalization to re-scale each data sample to the range $[-1, 1]$. For

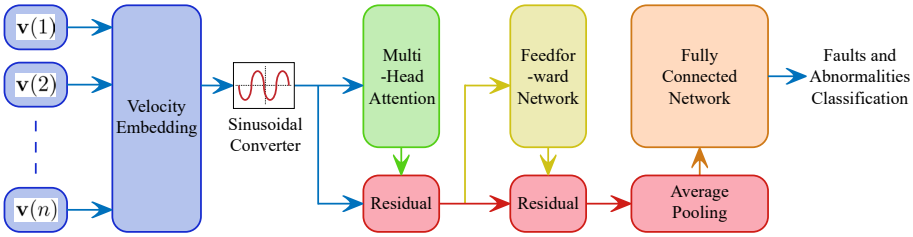



Fig. 5: Proposed multi-head attention network architecture for CAV platoon fault classification.

all $k \in \{0, \ldots, n\}$, the scaled velocity of vehicle $i \in \{1, 2, 3\}$ is given by

$$\bar{v}_i(k) = 2\left(\frac{v_i(k) - |v_i|_{\min}}{|v_i|_{\max} - |v_i|_{\min}}\right) - 1, \tag{6}$$

where $\bar{v}_i$ is the scaled velocity of vehicle $i$, and $|v_i|_{\max}, |v_i|_{\min}$ are the upper and lower bounds of the complete training dataset of the velocity signal $v_i$. The min-max normalization was used rather than zero-mean feature standardization since mapping to a single mean and variance would misrepresent the platoon's dynamics.

### C. Network Optimization

We randomly selected $80\%$ of the simulation data to use as a training dataset and isolated the remaining $20\%$ to use as a validation dataset. The network parameters are tuned using a cross-entropy loss function and Adam optimization [39]. We used a batch size of 100 and maintained default hyperparameter values for the optimizer. The network is trained for 150 epochs with early stopping if the validation loss does not decrease for 10 consecutive epochs. A random search was conducted over 20 combinations of model architecture sizes to optimize the network structure. Values were selected from: $\{8, 16, 32, 64\}$ for $d_{\text{model}}$, the velocity embedding size; $\{16, 32, 64, 128, 256\}$ for $d_k$, the key projection size; 1-8 attention heads; and $\{16, 32, 64\}$ for the feedforward hidden size. We constrained the range of the parameter values to reduce model overfitting and further reduced the search complexity by setting $d_v = d_k$. The combination with the lowest validation loss was taken as the final model, which uses $d_{\text{model}} = 64$, $d_v = d_k = 256$, 3 attention heads, and a feedforward hidden size of 64. The cross-entropy loss and the model's accuracy on the training and validation datasets are shown in Figure 6. The final model achieved a validation loss of 0.978 and a validation accuracy of $92.4\%$.

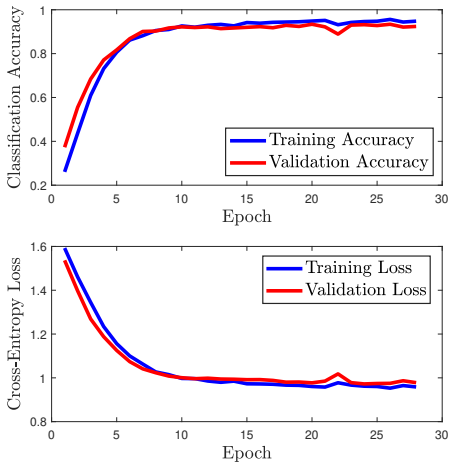



Fig. 6: Cross-entropy loss and accuracy on the training and validation datasets created with simulation data.

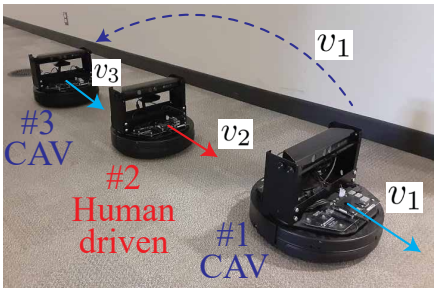



Fig. 7: The experimental setup consists of three differential robots. The first and third robots are autonomous, while the second is keyboard-controlled to emulate a human-driven vehicle.

## VII. Experimental Results

Given the ideal nature of the simulation data, we further validate our approach by leveraging real-world data collected from a physical CAV platoon. In this section, we will introduce our experimental testing setup, outline our data collection process, and discuss our testing results.

### A. Experimental Setup

The proposed approach was tested on a platoon of three laboratory mobile robots, shown in Figure 7. Similar to the platoon formulation in Figure 2, the first and third robots are autonomous with a communication link between them and the second robot is human-driven through the computer keyboard. The robots used are Quanser Qbot 2e robots, which are differential mobile robots with the ability to communicate with each other. Each Qbot consists of two wheels, each driven by a DC motor. Both wheels were set to the same velocity, so the platoon only had longitudinal motion. The platoon's desired velocity was sent to the first robot, and an internal PI controller (cruise control) was used to track it. The second robot is driven by the human to follow the first robot, and the third robot has another internal PI

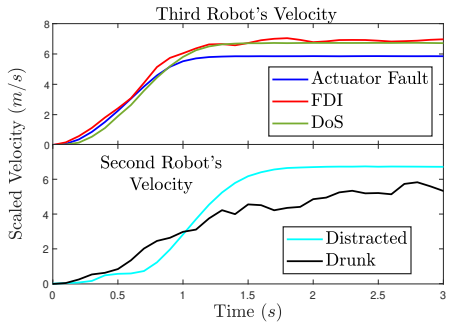



Fig. 8: An example of the emulated faults on the experimental setup.

controller to track the average velocity of the first two robots.

### B. Data Collection

The internal PI controllers were tuned such that the robots have a scaled settling time of 0.1 of the simulated CAVs. Thus, each run was 50 seconds long with a 0.1 second sampling time. That is, the velocity sequence length of both the simulations and experiments match. On the same scale of 0.1, the desired velocity of the robots was set to change randomly every 3 seconds. A testing dataset of 50 runs was recorded, where every 10 runs correspond to a different fault or abnormality class. A low pass filter was used to filter the velocity measurements. Figure 8 shows an example of the filtered laboratory robot responses with the five fault and abnormality classes. The CAV faults introduced in Section IV were modelled directly in the robot's interface, while the human driver faults were approximated by introducing input delays as specified by the human driver abnormality models.

### C. Testing Results

The trained MHA-ML network was tested on the collected data from the experimental setup. The model achieved a success rate of 90%, with the predictions shown in Figure 9a. From the misclassified samples, we observed that the most significant sources of error were the misprediction of the FDI and DoS fault classes. It can be observed from Figure 8 that these two faults result in very similar responses.

### D. Results Comparison

For sequence classification problems, RNNs are frequently leveraged as a powerful deep learning technique. An example on sensor data is given in [40]. We trained a deep RNN as a baseline model to compare with the proposed MHA-ML approach. The model leverages an equivalent velocity embedding layer but replaces the attention network with a bidirectional Long Short-Term Memory (LSTM) cell. The final outputs of the forward and backward LSTM are concatenated to form a feature vector and mapped to fault class probabilities via a fully-connected layer and softmax output function. The trained RNN baseline was tested with the same experimental test dataset and achieved 78% classification accuracy. Figure 9b shows the distribution of the baseline model's predictions. We conclude that the model easily

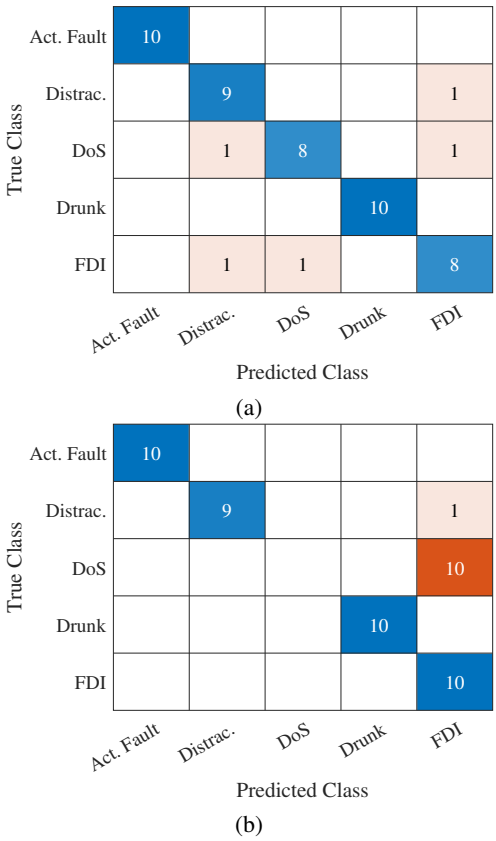



Fig. 9: Comparison between MHA-ML and RNN: (a) Confusion matrix visualization of test data predictions by the MHA-ML and (b) Confusion matrix visualization of test data predictions by the RNN baseline model.

predicts the actuator and drunk classes, but overpredicts the FDI class. We note that the baseline struggles with the DoS class especially, misclassifying all samples as FDI faults.

## VIII. Conclusion

This paper proposes multi-head attention machine learning to classify faults and abnormalities in mixed autonomous and human-driven vehicle platoons. Three CAV fault classes – (i) Actuator fault, (ii) FDI, and (iii) DoS – and two human driver abnormalities – (i) Distracted drivers, and (ii) Drunk drivers – are considered. We constructed a simulation model of a healthy platoon and altered it with the five fault and abnormality classes. The simulation model was used to generate the training dataset, on which the model achieved an accuracy of 92.4%. We conducted a laboratory experiment on three differential mobile robots, on which the trained model achieved an accuracy of 90%. For comparison, a baseline deep RNN model was trained and tested on the same datasets. The baseline RNN achieved only a 78% success rate on the same experimental dataset. The experiment showed that our approach, which classifies faults based on velocity measurements alone, generalizes to scenarios where the model dynamics are unknown and thus shows applicability to real-world CAV platoon environments.

## REFERENCES


[1] P. Wang, X. Wu, and X. He, "Modeling and analyzing cyberattack effects on connected automated vehicular platoons," *Transportation research part C: emerging technologies*, vol. 115, p. 102625, 2020.

[2] F. Zhao, Y. Liu, J. Wang, and L. Wang, "Distributed model predictive longitudinal control for a connected autonomous vehicle platoon with dynamic information flow topology," vol. 10, no. 9, pp. 204–211, 2021.

[3] M. Amirgholy, M. Shahabi, and H. O. Gao, "Traffic automation and lane management for communicant, autonomous, and human-driven vehicles," *Transportation research part C: emerging technologies*, vol. 111, pp. 477–495, 2020.

[4] C. Chen, J. Wang, Q. Xu, J. Wang, and K. Li, "Mixed platoon control of automated and human-driven vehicles at a signalized intersection: dynamical analysis and optimal control," *Transportation Research Part C: Emerging Technologies*, vol. 127, p. 103138, 2021.

[5] L. Zhang and E. Tseng, "Motion prediction of human-driven vehicles in mixed traffic with connected autonomous vehicles," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 398–403.

[6] A. Khalil, M. Al Janaideh, K. F. Aljanaideh, and D. Kundur, "Transmissibility-based health monitoring of the future connected autonomous vehicles networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 3633–3647, 2022.

[7] A. Khalil, M. Al Janaideh, and D. Kundur, "Online fault classification in connected autonomous vehicles using output-only measurements," *Mechanical Systems and Signal Processing*, vol. 190, pp. 1–15, 2023.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[9] V. K. Kukkala, S. V. Thiruloga, and S. Pasricha, "Roadmap for cybersecurity in autonomous vehicles," *IEEE Consumer Electronics Magazine*, 2022.

[10] X. Huang and J. Dong, "Reliable control policy of cyber-physical systems against a class of frequency-constrained sensor and actuator attacks," *IEEE Transactions on Cybernetics*, vol. 48, no. 12, pp. 3432–3439, 2018.

[11] D. Ding, Z. Wang, D. W. Ho, and G. Wei, "Observer-based event-triggering consensus control for multiagent systems with lossy sensors and cyber-attacks," *IEEE transactions on cybernetics*, vol. 47, no. 8, pp. 1936–1947, 2016.

[12] Y. Fang, H. Min, W. Wang, Z. Xu, and X. Zhao, "A fault detection and diagnosis system for autonomous vehicles based on hybrid approaches," *IEEE Sensors Journal*, vol. 20, no. 16, pp. 9359–9371, 2020.

[13] Z. A. Biron, S. Dey, and P. Pisu, "Real-time detection and estimation of denial of service attack in connected vehicle systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3893–3902, 2018.

[14] A. Petrillo, A. Pescape, and S. Santini, "A secure adaptive control for cooperative driving of autonomous connected vehicles in the presence of heterogeneous communication delays and cyberattacks," *IEEE transactions on cybernetics*, vol. 51, no. 3, pp. 1134–1149, 2020.

[15] G. Guo, P. Li, and L.-Y. Hao, "Adaptive fault-tolerant control of platoons with guaranteed traffic flow stability," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 6916–6927, 2020.

[16] A. Lopes and R. E. Araújo, "Active fault diagnosis method for vehicles in platoon formation," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3590–3603, 2020.

[17] J. Lunze, "Adaptive cruise control with guaranteed collision avoidance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1897–1907, 2018.

[18] B. Dowdeswell, R. Sinha, and S. G. MacDonell, "Finding faults: A scoping study of fault diagnostics for industrial cyber-physical systems," *Journal of systems and software*, vol. 168, p. 110638, 2020.

[19] T. Alladi, V. Kohli, V. Chamola, and F. R. Yu, "Securing the internet of vehicles: A deep learning-based classification framework," *IEEE networking letters*, vol. 3, no. 2, pp. 94–97, 2021.

[20] F. Van Wyk, Y. Wang, A. Khojandi, and N. Masoud, "Real-time sensor anomaly detection and identification in automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1264–1276, 2019.

[21] H. Ahmed and A. K. Nandi, "Compressive sampling and feature ranking framework for bearing fault classification with vibration signals," *IEEE Access*, vol. 6, pp. 44 731–44 746, 2018.

[22] S. U. Jan, Y.-D. Lee, J. Shin, and I. Koo, "Sensor fault classification based on support vector machine and statistical time-domain features," *IEEE Access*, vol. 5, pp. 8682–8690, 2017.

[23] J. Xie, Z. Li, Z. Zhou, and S. Liu, "A novel bearing fault classification method based on xgboost: The fusion of deep learning-based features and empirical features," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2020.

[24] D. Chakraborty, U. Sur, and P. K. Banerjee, "Random forest based fault classification technique for active power system networks," in *IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, 2019, pp. 1–4.

[25] W. Cui, X. Deng, and Z. Zhang, "Improved convolutional neural network based on multi-head attention mechanism for industrial process fault classification," in *IEEE Data Driven Control and Learning Systems Conference (DDCLS)*, 2020, pp. 918–922.

[26] Y. Zhang, B. Xu, and T. Zhao, "Convolutional multi-head self-attention on memory for aspect sentiment classification," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 4, pp. 1038–1044, 2020.

[27] Q. Zhang, R. Lu, Q. Wang, Z. Zhu, and P. Liu, "Interactive multi-head attention networks for aspect-level sentiment classification," *IEEE Access*, vol. 7, pp. 160 017–160 028, 2019.

[28] A. Kumar, V. T. Narapareddy, V. A. Srikanth, A. Malapati, and L. B. M. Neti, "Sarcasm detection using multi-head attention based bidirectional lstm," *IEEE Access*, vol. 8, pp. 6388–6397, 2020.

[29] H. Zhang, Z. Xiao, J. Wang, F. Li, and E. Szczerbicki, "A novel iot-perceptive human activity recognition (har) approach using multihead convolutional attention," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1072–1080, 2019.

[30] J. Mercat, T. Gilles, N. El Zoghby, G. Sandou, D. Beauvois, and G. P. Gil, "Multi-head attention for multi-modal joint vehicle motion forecasting," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9638–9644.

[31] K. Messaoud, N. Deo, M. M. Trivedi, and F. Nashashibi, "Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 165–170.

[32] H. Kim, D. Kim, G. Kim, J. Cho, and K. Huh, "Multi-head attention based probabilistic vehicle trajectory prediction," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1720–1725.

[33] A. Khalil, K. F. Aljanaideh, and M. Al Janaideh, "On detecting drunk drivers in mixed autonomous platoons using vehicles velocity measurements," *IEEE/ASME Transactions on Mechatronics*, Early Access, 2022.

[34] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, pp. 1805–1824, 2000.

[35] M. Pirani, E. Hashemi, A. Khajepour, B. Fidan, B. Litkouhi, S.-K. Chen, and S. Sundaram, "Cooperative vehicle speed fault diagnosis and correction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 783–789, 2018.

[36] G. Zhang, H. Zhang, X. Huang, J. Wang, H. Yu, and R. Graaf, "Active fault-tolerant control for electric vehicles with independently driven rear in-wheel motors against certain actuator faults," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1557–1572, 2015.

[37] M. Emily Parcell, M. Shivani Patel, C. Severin, Y. Cho, and A. Chaparro, "Effect of driver distraction on vehicle speed control," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 65, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2021, pp. 958–962.

[38] N. H. T. S. A. (NHTSA), "The effects of blood alcohol concentration," pp. [Online]. Available: https://www.nhtsa.gov/risky–driving/drunk–driving.

[39] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[40] A. Girma, X. Yan, and A. Homaifar, "Driver identification based on vehicle telematics data using lstm-recurrent neural network," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2019, pp. 894–902.