

Paper Type: Original Article

A Hybrid Approach for Anomaly Detection in Autonomous Vehicles Using Genetic Algorithm and GRU Model

Zongke Bao* 

Department of Accounting, Zhejiang University, China; zongkebao@mail.cnas.ac.cn.

Citation:

Received: ----

Revised: ----

Accepted:---

Bao. (2025). A hybrid approach for anomaly detection in autonomous vehicles using genetic algorithm and GRU model. *Computational Engineering and Technology Innovations*, 2(4), 122-133.

Abstract

Autonomous Vehicle (AV) and Connected Autonomous Vehicle (CAV) technologies for transport systems to become safer, less congested and cleaner is a technological breakthrough. Nevertheless, these are complex systems and the interconnection between them presents a number of challenges, more notably in the form of cyber security and Anomaly Detection (AD) from sensor data. Errors due to sensor malfunctions, cyber-attacks and environmental disturbances can disrupt vehicle performance and safety. In this work, we propose the Hybrid model which combines Genetic Algorithm (GA) for feature selection process and Gated Recurrent Unit (GRU) based models for anomaly classification. The GA is used to do so such that the results would be more accurate of detection. where the GRU simulates the processing of time-series sensor data to detect anomalous behaviors. We evaluate this model on the MMITSS-Multi-Modal Intelligent Transportation Signal Systems dataset which is a large-scale real-world dataset with millions of records, and three different feature channel configurations, one feature channel Mean Squared Error (MSE), Two feature channels (MSE, Mean Absolute Percentage Error (MAPE)), three feature channels (MSE, MAPE, original data). The proposed model achieves better performance in terms of accuracy, precision, recall and specificity as compared to existing techniques, especially for detecting anomalies. This hybrid approach is an excellent and scalable means of detecting online anomalies in CAVs, thus contributing to enhance the safety and dependability of self-driving systems.

Keywords: Connected autonomous vehicle, Cyber security, Anomaly detection, Genetic algorithm and gated recurrent unit.

1 | Introduction

The concept of the Autonomous Vehicle (AV) arose due to advances in connectivity and automation. This technology is supposed to make transportation safer, more energy-efficient, quieter and less crowded. An AV, or an SDV can operate and accomplish necessary tasks without human intervention because of its capacity to understand the environment [1]. A self-driving vehicle employs a fully automated driving system to process situations in its surroundings as a human driver would normally do. People have talked about self-

 Corresponding Author: zongkebao@mail.cnas.ac.cn

 <https://doi.org/10.48314/ceti.vi.72>



Licensee System Analytics. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

driving cars for a long time, but they're expensive to build and it's been difficult to get enough of them on the road. One such technology is AV powered by alternative fuel sources rather than oil. This means less greenhouse gases are produced by normal fossil fuel-converting-vehicles [2]. Similar tools - capable of helping with autonomous and safe Connected Autonomous Vehicle (CAV) evolution (without the driver) on very massive amounts of information-Machine Learning, including but not limited to Deep Learning methods. This kind of procedures can include, real-time sensor Anomaly Detection (AD) [3], problems in sections of the car as e.g. battery [4], and strange behaviors in the driver [5]. There are six levels of autonomy, where level 0 refers to a human driver with no support and level 5 is an automaton without any human intervention [6]. The four companies that made statements to us Waymo (formerly the google self-driving car project), Cruise, TuSimple and Aurora are just a few of those working on level 5 self-driving cars. Breaking, Advances in technology, particularly the development of deep learning, are enabling cars to drive themselves. This is possible now because of the low-cost but powerful data analytics that cars can exploit to process and understand a huge amount of complex data all from different sources that they have to understand in order to do their work. AD of CAV sensor is challenging but essential task, because it has a lot of privacy and security considerations.

When vehicles are tethered together you might use other sources of data to spot anomalies, such as GPS signals, readings from the engine or any other sensors reading off the vehicle [7]. You can apply machine learning algorithms such as clustering and classification to these data to examine whether there are any weird patterns [8]. A clustering algorithm, for example, may identify patterns in the data and points that are much dissimilar from those patterns. Investigation of anomalies in Connected Vehicles (CVs) also requires detection to be performed in real-time. Connected cars is a useful of data that anomalies should be detected and reported in real time [9]. You also have to look at anomalies to figure out why the issue occurred, as they could be a signal of a larger problem that needs fixing immediately. The data from the numerous systems in a connected and automated car like sensors, cameras and GPS equipment is constantly being monitored to detect problems and prevent them from happening. The system itself identifies regular behaviors and detects any changes in those patterns, which could indicate misuse, through the use of sophisticated machine learning algorithms. This prevents accidents from occurring and makes the car operate more effectively overall. Discovering the weirdness of the world is a central issue for self-driving cars [10]. Anomalies may stem from different reasons, such as sensor failures, corrupted data, natural environment disturbances or malicious attacks (e.g., GPS spoofing and False Data Injection (FDI) [11]. It is essential to detect such anomalies in order to operate AVs safely and avoid the accidents. Conventional condition-based maintenance and AD can be inadequate in the ever changing, complex landscape where AV operates because these techniques struggle to scale, learn and respond quickly to new emerging threats [12].

2 | Literature Survey

Yang et al. [13] introduced an innovative technique for identifying GPS spoofing attacks aimed at Connected Vehicles (CVs) and AVs. They employed a learning from demonstration methodology, which derives a computationally efficient driving model using past vehicle trajectories to forecast typical driving behaviors. Then, a statistical method was used to find the differences between the observed trajectory and the expected normal trajectory in order to find anomalies. Two threat models were used to test the method, one for the multi-sensor fusion system of AVs and the other for the intersection movement assist application of CVs. The validation was conducted using two actual datasets, specifically the KAIST and Michigan roundabout datasets. The results indicated that the suggested model could find virtually all of the attacks with few false positives and false negatives. This shows that it works to make CVs and AVs safer and more reliable. Mansourian et al. [14] put out a deep learning-based AD system for CAVs, employing spatiotemporal information to identify anomalies in sensor data. The model, validated on real-world datasets, outperformed existing methods, achieving improved accuracy and detection capability. This highlights its potential to enhance CAV safety by reducing false positives and improving AD. Alsulami et al. [15] put forward a symmetrical AD technique for self-driving cars, concentrating on identifying FDI assaults that use hacked

sensors to sneak into the control system. The study utilized a simulation-based system created by Math Works Inc. to implement FDI attacks and gather statistics for analysis. The authors used a Long Short-Term Memory (LSTM) deep learning model to sort the data into normal and abnormal groups, and they got an amazing accuracy rate of 99.95%. The experimental results demonstrated that their proposed solution surpassed existing leading models in identifying cyberattacks on AVs, hence assuring improved security and operational stability.

Abdi and Ghasemi-Tabar [16] introduced ARAD, an automatic real time AD system on the sensors of AVs using a lightweight supervised learning approach. ARAD identifies and categorizes anomalies based on a hierarchical structure and sequence prediction approach, and it removes them using a rule-based engine. The system attained 84.6% precision and 87% recall at a reduction in computational cost as high as 75.6%, which indicates its real-time AD effectiveness. Nazat et al. [17] introduced a general framework to evaluate black-box Explainable Artificial Intelligence (XAI) approach for enhancing AD in autonomous driving systems. Six metrics are chosen as the yardstick to quantify how well SHAP and LIME perform, descriptive accuracy, sparsity, stability, ease of computation, robustness (where applicable), and completeness. The research utilized these XAI techniques to determine the importance of features involved in anomaly classification, testing out two popular datasets, VeReMi and sensor. They provide useful insights into the pros and cons of current black-box XAI technologies to be more utilized for real-time AD and cyber security in self-driving cars.

Min et al. [18] presented an interpretable AD method on AVs using a Denoising Variational Transformer (DVT). They have developed a variational attention mechanism which applies the Gaussian prior and improves the accuracy of AD. The model reconstructs its input data to discover anomalies. It also has a residual interpreter for interpreting the results of detection by seeing how each feature fared overall. Experiments show that DVT performs superior to conventional unsupervised methods, improving the F1-score by 8%, 50% and AUC score by 1%, 20%. Moreover, the residual interpreter was more accurate by more than 50% than kernel Shapley additive explanations while was 99% faster. Rajendar and Kaliappan propose a M-Convolutional Neural Networks (CNN) model to detect AV anomalies using SPMD dataset, which is applied to safety measurement tasks. M-CNN model consists of layers with ReLU activation functions that have been well-trained for feature extraction and AD. We then refine the model using Adam optimizer. The proposed model was compared with the Isolation Forest (IF) algorithm and Support Vector Machine (SVM) method, and it is founded that it performs better than IF and SVM in AV anomalies detection, which means an accuracy up to 99.40% for detecting such kind of anomalies [19].

Tsiktsiris et al. [20] proposed a semi-supervised AD in the local maxima of MD responses with when capturing all signals from overhead fisheye cameras in autonomous shuttles to improve safety. The architecture of it is based on spatio-temporal autoencoders with convolutional and recurrent layers. The model was trained on unlabeled video data, optimizing for Center-Weighted Loss (CWL) to capture the middle of the cabin. The proposed method had better AUC performance than those of the CETH-AV dataset and demonstrated stability in comparison experiments on the UCF-Crime and Shanghai Tech datasets. When applied on top of self-driving minibuses integrated with NVIDIA Jetson systems for real-time passenger safety monitoring, it functioned well [20].

Wang and Zhang [21] have presented a new AD model, CWT-CNN for connected and self-driven automobiles or cars, which would provide the high-level secure within the transportation networks by integrating Continuous Wavelet Transform (CWT) with CNN. The CWT is employed in the model to convert signals from car inner sensors (i.e., accelerations of roof, menuum, and floors) into wavelet coefficients. This results into vehicle state images which capture both the temporal and frequency characteristics. This method reveals the latent structures in vehicle events for better event detection. We employ the model to produce a two-dimensional CWTS, and input it to the optimized CNN to extract abnormal pattern associated with malicious activities. Experiments on the capability of detecting various problems in vehicles demonstrate the

effectiveness of our model with a 2.51% higher accuracy and a 2.46% higher F1 score than those obtained using other systems.

Lian et al. [22] proposes a framework for seeking and repairing failures in optimal autonomous systems based on inverse Reinforcement Learning (RL). The approach identifies and protects normal from anomalous agents in objective function optimizing systems. It involves three sequential stages: training (it learns the normal objective), testing (behaviors of entities is compared online with what is nominal for a system at any particular time) and repairing (the anomalous behaviors is modified to bring it closer to the detected anomaly). Tests on a quadrotor UAV showed that the technique was effective for locating and resolving problems.

3 | Proposed Methodology

The primary objective of this study is to create an innovative methodology capable of distinguishing between heavy and light automobiles from the input image. *Fig. 1* shows the flow chart. We will use Python tools to design and test the proposed model. Below is a thorough explanation of the suggested method:

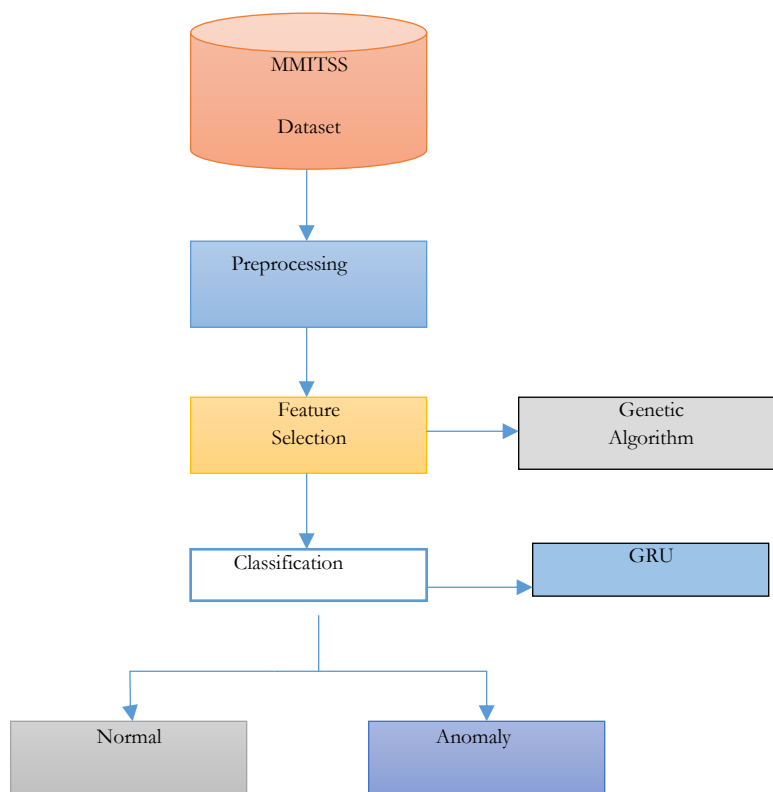


Fig. 1. flow chart.

3.1 | Dataset Description

The multi-modal intelligent transportation signal systems dataset is used to determine how well the model worked. The Dynamic Mobility Applications (DMA) initiative created MMITSS, a traffic signal system that aims to provide a foundation for multi-modal transportation traffic information. This driving dataset is a multivariate time series, which implies that it is a series of data points for more than one variable that are sorted by time. There are more than 4.7 million records in it, and it has 17 properties. The most essential ones are Index (the sequence number since the vehicle started), LAP ID (trip ID), MS (time in milliseconds), speed, acceleration, latitude, longitude, heading, lane number, and more. During the day (from 6:00 AM to 12:00 AM), CAV gets sensor signals every 100 ms. The data that was collected needs to be cleaned up and then put into a 3D array (samples, timesteps, features) so that it fits into the LSTM auto encoder input. Here, samples are the number of samples in our dataset, timesteps is the window size, and features is the number of features.

3.2 | Preprocessing

The first step in prepping data for AD in CAVs is cleaning the data. That is filtering away useless, noisy or distorted information from raw sensor readings to keep only useful data for further processing. We then finally standardize the data using normalization, shifting each feature here so that the mean value is 0 and center to unit variance. This phase ensures that all the features have the same scale which is very important for training a model effectively. For time series data, a sliding window approach is used, breaking down the continuous sensor signals into smaller segments (samples) so that the model can more readily consume sequential information. After the windows are generated, shuffling is applied to remove any potential bias or overfitting caused by data ordering. Then the data is reshaped as 3D array with dimension; (features, time stamp, samples), for being passed into deep learning models as input such as LSTM or Transformer. For the better performance of next models, there are also some operations such as expanding the sample width and shuffling the dimensions for them, which can contribute to better quality of feature extraction and avoid overfitting by means of a more diversified training process. These pre-processing steps allow the data from the sensors to be trained well and assure effective training and AD. In this way, model is able to learn the pattern and detect the change in progressive.

3.3 | Feature Selection

Once the data preprocessing step is completed, feature selection becomes another key step in the CAVs AD pipeline. The purpose of this step is to identify the most relevant features out of raw sensor data, such that the model becomes better and takes less computation power. Feature selection is capable of removing irrelevant, redundant or noisy features, thereby accelerating training and reducing overfitting potential. Genetic Algorithm (GA) is used in this research as part of the feature selection technique. GA gets to cover the feature space step by step in seeking the best subset of features which will help it effectively detect anomalies. Here are the next stages as to how you do it.

3.3.1 | Chromosome representation

Every potential solution (chromosome) in the population embodies a subset of characteristics, where each feature is encoded as a binary value 1 and 0 (selected feature and excluded feature respectively).

For instance, if the dataset consists of 17 features, the chromosome would be a binary vector of length 17. A vector such as [1, 0, 1, 0, 1, ..., 0] means that features 1, 3, and 5 are selected, while the others are excluded.

3.3.2 | Fitness function

Using the chosen features, an LSTM auto-encoder model is trained to find out how fit each person in the population is. The fitness function is designed to maximize the model's accuracy or classification performance by assessing how well the selected feature subset contributes to AD. To get the fitness score, the trained model is run on a validation set. A higher fitness score means the model is doing a better job. The fitness function can be defined as follows:

- I. Select the features indicated by the binary vector (chromosome).
- II. Train an LSTM auto-encoder for AD using only the selected features.
- III. Apply a CNN classifier to further classify data as either normal or anomalous.
- IV. Calculate the model's accuracy or F1-score on a validation set.
- V. The fitness score is the negative of the classification error (since GA minimizes the fitness function).

3.3.3 | Population initialization

The initial population of binary vectors (chromosomes) is randomly generated. Each individual represents a different subset of features, creating diversity in the population. A typical population size could be set to 50-100 individuals.

3.3.4 | Selection process

The selection process identifies the best performing individuals based on their fitness scores. Tournament selection or roulette-wheel selection are commonly used methods to select parents. The selected parents are then used to create offspring via crossover and mutation, guiding the algorithm towards the best feature subset.

3.3.5 | Crossover and mutation

- I. Crossover: a one-point crossover operation is used to mix the genetic material (feature subsets) of two parent solutions. This step produces offspring with a combination of features from both parents, allowing the exploration of new solutions.
- II. Mutation: during the mutation step, random bits in the binary vector are flipped (changing a 0 to 1 or vice versa), introducing variation into the population. This stage makes sure that the search space is well-explored and keeps the algorithm from getting stuck in local optima.

3.3.6 | Iterative evolution

The GA iteratively evolves the population over multiple generations (typically 50-100 generations), refining the feature subsets toward the optimal solution. In each generation, the best-performing individuals are selected for crossover and mutation. This iterative process gradually improves the quality of the selected feature subsets.

3.3.7 | Final feature subset

After several generations, the best individual in the final population is chosen. This person is the best set of features that makes the AD model work best. The final model for finding anomalies is then trained using these chosen features.

4 | Classification

4.1 | Gated Recurrent Unit

The last stage of classification uses a Gated Recurrent Unit (GRU) network GRU, which is capable to process sequential sensor data provided by AV systems. GRU is adopted, because of its capability to capture the temporal dependency and dynamic pattern as well as having fewer parameters than LSTM and enabling better generalization for real-time AD. The selected GA features in this work are form the time series inputs and each sample is passed to a GRU-based architecture. There are two GRU layers in the classification model (64 units and 32 units, respectively). After that, there is a fully connected layer with 16 neurons with ReLU activation, and finally, a sigmoid output node for binary classification. To avoid overfitting, a dropout rate of 0.3 is used. The model is trained with the Adam optimizer (learning rate = 0.001) and early halting based on validation loss. When GA has completed the process of feature selection, in next stage we have to utilize the features that have been selected for AD by training a classifier.

In this phase, we use a GRU model of an Recurrent Neural Network (RNN) and that is designed to process sequence data such as time series sensor data collected from CAVs. The GRU model is meant to find the time-based relationships in the data, meaning it can learn patterns over time (e.g., changes in speed, acceleration, and heading over consecutive time steps) that help distinguish normal driving behaviors from anomalous patterns. The selected features from the GA step are fed into the GRU model, which processes these features step-by-step (or time step by time step) to recognize important patterns in the sequence of data. Three separate feature channels are used for assessing how well the model works; the original data used to train the model, Mean Squared Error (MSE), and Mean Absolute Percentage Error (MAPE). In the first setup, the only statistic utilized to evaluate is MSE. The MSE is the average of the squared discrepancies between the anticipated probabilities (the output from the GRU model) and the actual binary labels (0 for normal and 1 for anomalous). A lower MSE means greater performance because it means that the model's

projected probabilities are closer to the actual class labels. Then, in the second case MAPE is included so as to guide what model should be considered as the best. First off, the MAPE measure demonstrates by how much the model's prediction is wrong in percentage terms, it does so by taking a percent difference between the predicted probability and the true ones.

As MSE and MAPE are being combined, we can show the absolute error as well as percent-based error which provide a more comprehensive and thorough understanding of how well our model is able to sort data accurately. Lastly, in the third setting, one combines MSE, MAPE and original data (features applied to train the model). This makes it possible for us to assess how well the GRU model uses these selected features to construct its internal states during learning and prediction. The addition of raw data ensures that features input to the model have a strong degree of impact on classification as it assesses just how well the model can pull out anomalies. To make sure the evaluation is strong, the dataset is split into training, validation, and testing sets in a 70:15:15 ratio. The order of the data is kept the same to prevent data leaking. Accuracy, precision, recall, specificity, and F1-score were used to measure how well the GRU classifier works. Additional evaluation metrics were also provided such as AUC-ROC and a confusion matrix, which confirm that the GRU effectively distinguishes normal and anomalous vehicle behaviors. Table 1 demonstrates how well the model performed in the three different situations.

5 | Result and Discussion

The evaluation of the GRU-based AD model was conducted using three different feature channel configurations: One feature channel MSE, Two feature channels (MSE + MAPE), and Three feature channels (MSE + MAPE + Original Data). Accuracy, precision, recall, and specificity were used to judge how well the model worked. The table below shows the findings. These measures were chosen to fully test the model's capacity to find strange things in CAVs sensor data, making sure that it correctly identifies both typical and strange cases.

Table 1. Evaluation of the channel boosting use.

Number of GRU Input Channels	Accuracy	Precision	Recall	Specificity
One feature channel (MSE)	94.2%	93.1%	92.3%	94.9%
Two feature channels (MSE + MAPE)	96.3%	95.4%	94.5%	96.1%
Three feature channels (MSE + MAPE + original data)	96.2%	95.0%	94.8%	96.0%

5.1 | Accuracy

The accuracy statistic allows see how many predictions a model got right as Equation 1. It is quite clear that the GRU based model outperforms dramatically over the CNN model for various settings in our observations. On OFC MSE, GRU obtains 94.2% accuracy, while CNN catches up to 93.4%. Under MSE + MAPE, GRU's accuracy is 96.3% and CNN's is 95.6%. With the incorporation of original data, the performance is improved slightly to GRU=96.2%>CNN=95.5%. The above results prove that the GRU model, which is customized for CAV sensors data is more capable of detecting precisely the anomalous patrols from other models.

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

Table 2. GRU based model outperforms.

Accuracy	CNN	GRU (Proposed)
One feature channel (MSE)	93.4%	94.2%
Two feature channels (MSE + MAPE)	95.6%	96.3%
Three feature channels (MSE + MAPE + original data)	95.5%	96.2%

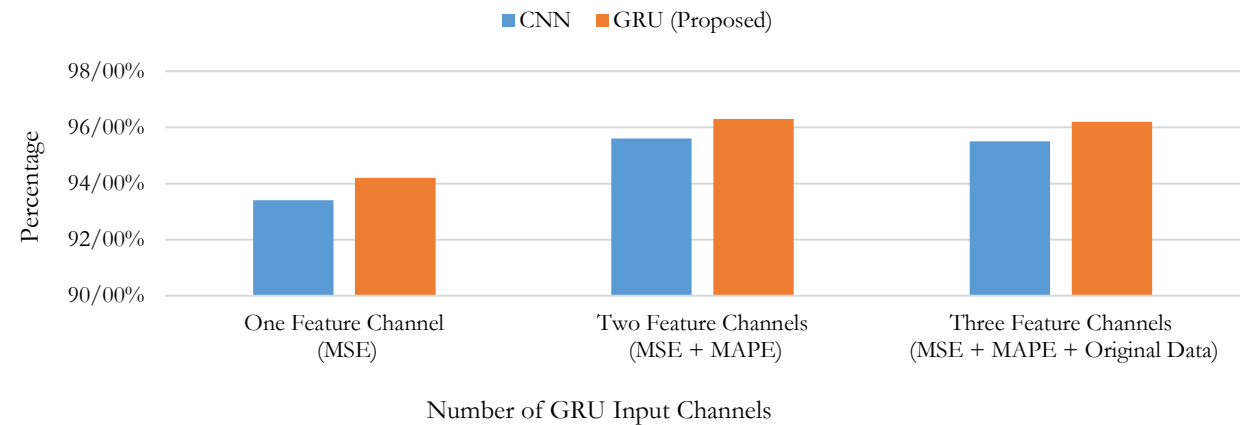


Fig. 2. Accuracy.

5.2 | Precision

The Precision score represents all the predictions that got right of all positive ones as *Eq. (2)*. This is a nice way to see if the model predicts true positives and not too many false ones. In the one feature channel MSE, GRU obtains 93.1% precision, as compared to CNN’s 92.7%. GRU (MSE+MAPE) with two feature channels: 95.4% surpassing CNN’s 94.5%. Incorporating the original data in the three feature channels (MSE + MAPE + original data) configuration achieves 95.0% precision for GRU, whereas it is still at 94.2% for CNN. These findings as shown in *Fig. 3* and *Table 3*, indicate that GRU model is better in reducing false alarms and well recognizing outliers.

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

(2)

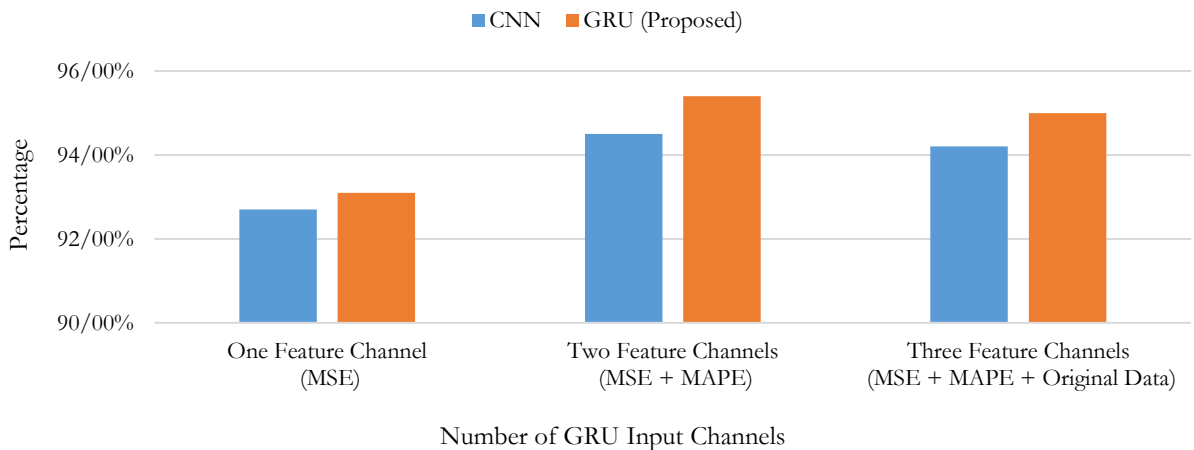


Fig. 3. Precision.

Table 3. GRU based model outperforms.

Precision	CNN	GRU (Proposed)
One feature channel (MSE)	92.7%	93.1%
Two feature channels (MSE + MAPE)	94.5%	95.4%
Three feature channels (MSE + MAPE + original data)	94.2%	95.0%

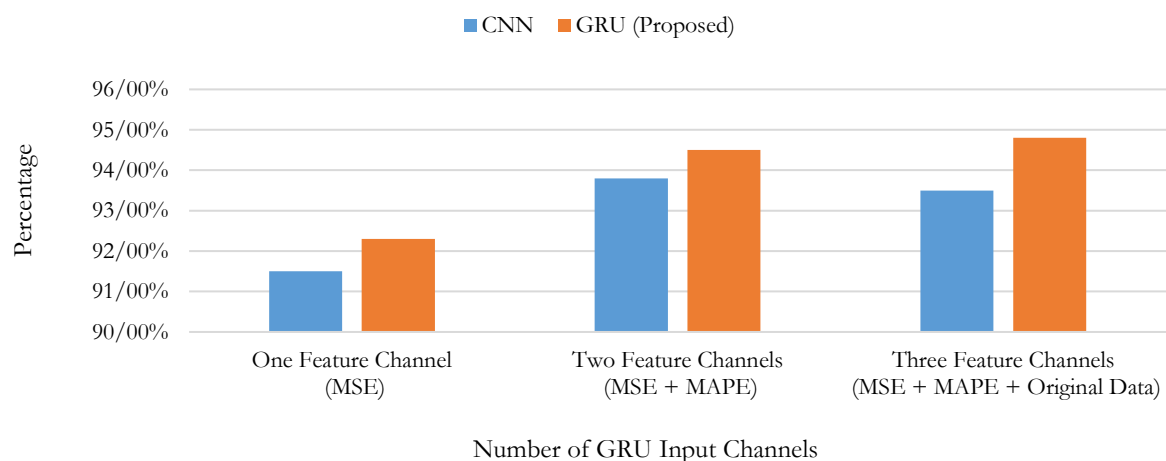
5.3 | Recall

The recall score refers to the percentage of the true positive predictions that were correct as described by Eq. (3). It does not account for false positive and is meant to gauge the positive predictions themselves. The best improvement is achieved with the introduction of the MAPE map in MSE + MAPE localization configuration: The accuracy of GRU reaches a 94.5% recall as opposed to 93.8% for CNN. In the three feature channels (MSE + MAPE + original data) setting, incorporating original data improves GRU's recall to 94.8%, further validating the advantage of using more features for AD. The results are shown in Fig. 4 and Table 4.

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

Table 4. GRU based model outperforms.

Feature Channel Configuration	CNN	GRU (Proposed)
One feature channel (MSE)	91.5%	92.3%
Two feature channels (MSE + MAPE)	93.8%	94.5%
Three feature channels (MSE + MAPE + original data)	93.5%	94.8%

**Fig. 4. Recall.**

5.4 | Specificity

Then we can calculate the number of negative predictions by the model that were indeed negative. It's the same thing as Recall but for negative predictions. For both models, specificity increases when adding MAPE to MSE in the two feature channels (MSE + MAPE) setting; 96.1% for GRU compared to 95.8% for CNN. In the three feature channels (MSE + MAPE + Original Data) configuration, GRU still remains with better performance up to 96.0% specificity and CNN keeps its superiority with 95.4%, this result demonstrates that adding several features can help GRU more significantly in accurate classification.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}.$$

(4)

Table 5. Specificity based model outperforms.

Feature Channel Configuration	CNN	GRU (Proposed)
One feature channel (MSE)	94.6%	94.9%
Two feature channels (MSE + MAPE)	95.8%	96.1%
Three feature channels (MSE + MAPE + original data)	95.4%	96.0%

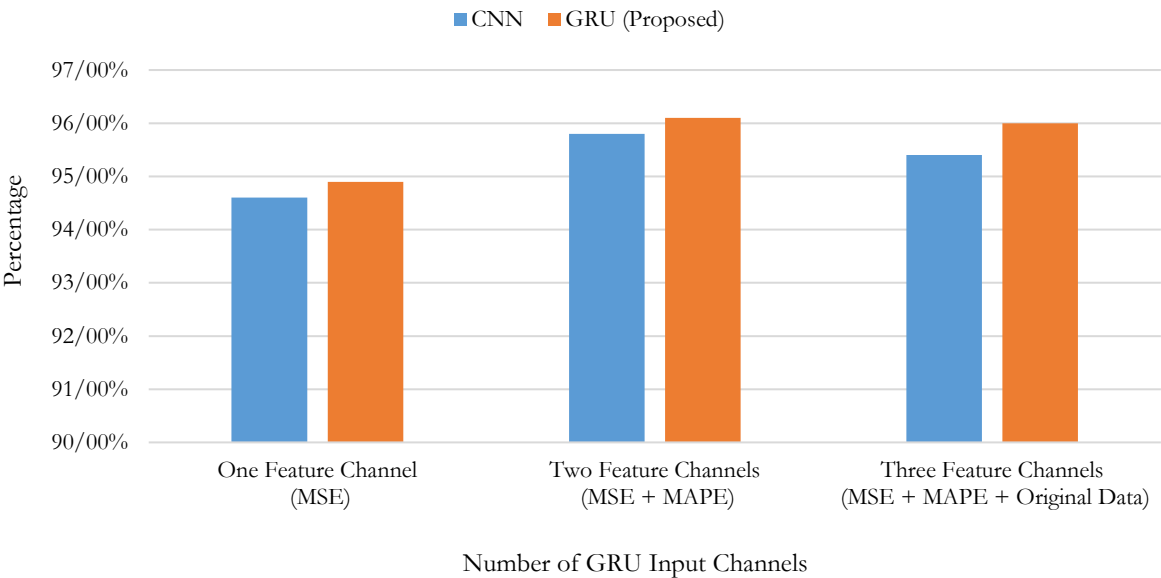


Fig. 5. Recall.

6 | Conclusion

Autonomous driven vehicles are a potential solution for future transportation safety and efficiency. There are still problems to be worked out, however, if this technology is going to work as well as it can. Anomalies are one of the greatest challenges that can make an autonomous car system fail. This paper provides an extensive survey on the anomalies that CAVs may suffer -more specifically, describes the different types, their causes and consequences that can affect both sensors' elements and decision-making system. as well as the many existing detection methods. A novel approach of completely methodology for AD in CAVs proposed in this paper by coupling GA feature selection and GRU classification models together. This pre- processing has the advantage that the data is ready to be trained efficiently and effectively. The GA systematically searches the feature space by evaluating subsets based on their effect in optimizing model effectiveness. The GA enhances the capability of GRU model detecting anomalies by enhancing feature set. feature channels model GRU model assessments were performed on three feature channel options: One Feature Channel MSE, two feature channels (MSE + MAPE), and three feature channels (MSE + MAPE + Original Data). These criteria permitted evaluating AD performance of the proposed model, targeting at the MSE and MAPE. The findings emphasize the necessity of integrating temporal models with fully-automated feature selection and evaluation metrics to achieve high performance in autonomous driving systems. In future we will focus on addressing these issues and enhancing the scalability and robustness of the model in real-world scenarios.

Data Availability

All data are included in the text.

Funding

This research received no external funding.

Conflict of Interest

No potential conflict of interest was reported by the authors.

Consent for Publication

All authors have provided their consent for the publication of this manuscript.

Ethics Approval and Consent to Participate

This article does not involve studies with human participants or animals conducted by any authors.

References

- [1] Bogdoll, D., Nitsche, M., & Zöllner, J. M. (2022). Anomaly detection in autonomous driving: A survey. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4488–4499). CVPR 2022.
https://openaccess.thecvf.com/content/CVPR2022W/WAD/papers/Bogdoll_Anomaly_Detection_in_Autonomous_Driving_A_Survey_CVPRW_2022_paper.pdf
- [2] Jang, C. K., Lee, J., & Yi, O. (2017). Encryption scheme in portable electric vehicle charging infrastructure: encryption scheme using symmetric key. *2017 4th international conference on computer applications and information processing technology (CAIPT)* (pp. 1–5). IEEE. <https://doi.org/10.1109/CAIPT.2017.8320674>
- [3] Wang, J., Zhang, L., Huang, Y., & Zhao, J. (2020). Safety of autonomous vehicles. *Journal of advanced transportation*, 2020(1), 8867757. <https://doi.org/10.1155/2020/8867757>
- [4] Li, X., Li, J., Abdollahi, A., & Jones, T. (2021). Data-driven thermal anomaly detection for batteries using unsupervised shape clustering. *2021 IEEE 30th international symposium on industrial electronics (ISIE)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ISIE45552.2021.9576348>
- [5] Quader, F., & Janeja, V. (2019). Anomaly detection: under the [data] hood in smart cars. *2019 IEEE international conference on smart computing (Smartcomp)* (pp. 126–131). IEEE.
<https://doi.org/10.1109/SMARTCOMP.2019.00041>
- [6] Spratlin Jr, A. D. (2020). Autonomous and electric cars and trucks: Did they survive COVID? *Brief*, 50, 50.
<https://www.butlersnow.com/news-and-events/autonomous-and-electric-cars-and-trucks-did-they-survive-covid>
- [7] Yun, K., Yun, H., Lee, S., Oh, J., Kim, M., Lim, M., ... & Choi, J. (2024). A study on machine learning-enhanced roadside unit-based detection of abnormal driving in autonomous vehicles. *Electronics*, 13(2), 288. <https://doi.org/10.3390/electronics13020288>
- [8] Yang, T., & Lv, C. (2021). A secure sensor fusion framework for connected and automated vehicles under sensor attacks. *IEEE internet of things journal*, 9(22), 22357–22365. <https://doi.org/10.1109/JIOT.2021.3101502>
- [9] Javed, A. R., Usman, M., Rehman, S. U., Khan, M. U., & Haghighi, M. S. (2020). Anomaly detection in automated vehicles using multistage attention-based convolutional neural network. *IEEE transactions on intelligent transportation systems*, 22(7), 4291–4300. <https://doi.org/10.1109/TITS.2020.3025875>
- [10] El-Shafeiy, E., Alsabaan, M., Ibrahim, M. I., & Elwahsh, H. (2023). Real-time anomaly detection for water quality sensor monitoring based on multivariate deep learning technique. *Sensors*, 23(20), 8613. <https://doi.org/10.3390/s23208613>
- [11] Abrar, M. M., Youssef, A., Islam, R., Satam, S., Latibari, B. S., Hariri, S., ... & Satam, P. (2024). *GPS-IDS: An anomaly-based GPS spoofing attack detection framework for autonomous vehicles*.
<https://doi.org/10.48550/ARXIV.2405.08359>
- [12] Oucheikh, R., Fri, M., Fedouaki, F., & Hain, M. (2020). Deep real-time anomaly detection for connected autonomous vehicles. *Procedia computer science*, 177, 456–461. <https://doi.org/10.1016/j.procs.2020.10.062>

- [13] Yang, Z., Ying, J., Shen, J., Feng, Y., Chen, Q. A., Mao, Z. M., & Liu, H. (2023). Anomaly detection against GPS spoofing attacks on connected and autonomous vehicles using learning from demonstration. *IEEE transactions on intelligent transportation systems*, 24(9), 9462–9475. <https://doi.org/10.1109/TITS.2023.3269029>
- [14] Mansourian, P., Zhang, N., Jaekel, A., & Kneppers, M. (2023). Deep learning-based anomaly detection for connected autonomous vehicles using spatiotemporal information. *IEEE transactions on intelligent transportation systems*, 24(12), 16006–16017. <https://doi.org/10.1109/TITS.2023.3286611>
- [15] Alsulami, A. A., Abu Al-Haija, Q., Alqahtani, A., & Alsini, R. (2022). Symmetrical simulation scheme for anomaly detection in autonomous vehicles based on LSTM model. *Symmetry*, 14(7), 1450. <https://doi.org/10.3390/sym14071450>
- [16] Abdi, A., & Ghasemi-Tabar, A. (2024). Arad: Automated and real-time anomaly detection in sensors of autonomous vehicles through a lightweight supervised learning approach. *IEEE access*, 12, 90432–90441. <https://doi.org/10.1109/ACCESS.2024.3420090>
- [17] Nazat, S., Arreche, O., & Abdallah, M. (2024). On evaluating black-box explainable AI methods for enhancing anomaly detection in autonomous driving systems. *Sensors*, 24(11), 3515. <https://doi.org/10.3390/s24113515>
- [18] Min, H., Lei, X., Wu, X., Fang, Y., Chen, S., Wang, W., & Zhao, X. (2024). Toward interpretable anomaly detection for autonomous vehicles with denoising variational transformer. *Engineering applications of artificial intelligence*, 129, 107601. <https://doi.org/10.1016/j.engappai.2023.107601>
- [19] Rajendar, S., & Kaliappan, V. K. (2022). Sensor Data based anomaly detection in autonomous vehicles using modified convolutional neural network. *Intelligent automation & soft computing*, 32(2). <https://doi.org/10.32604/iasc.2022.020936>
- [20] Tsiktsiris, D., Lalas, A., Dasygenis, M., & Votis, K. (2024). Enhancing the safety of autonomous vehicles: Semi-supervised anomaly detection with overhead fisheye perspective. *IEEE access*, 12, 68905–68915. <https://doi.org/10.1109/ACCESS.2024.3399841>
- [21] Wang, L., & Zhang, X. (2023). Anomaly detection for automated vehicles integrating continuous wavelet transform and convolutional neural network. *Applied sciences*, 13(9), 5525. <https://doi.org/10.3390/app13095525>
- [22] Lian, B., Kartal, Y., Lewis, F. L., Mikulski, D. G., Hudas, G. R., Wan, Y., & Davoudi, A. (2022). Anomaly detection and correction of optimizing autonomous systems with inverse reinforcement learning. *IEEE transactions on cybernetics*, 53(7), 4555–4566. <https://doi.org/10.1109/TCYB.2022.3213526>