IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 24 April 2024, accepted 17 June 2024, date of publication 27 June 2024, date of current version 8 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3420090

RESEARCH ARTICLE

ARAD: Automated and Real-Time Anomaly Detection in Sensors of Autonomous Vehicles Through a Lightweight Supervised Learning Approach

ATHENA ABDI AND ARASH GHASEMI-TABAR
Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran 16317-14191, Iran
Corresponding author: Athena Abdi (a_abdi@kntu.ac.ir)

ABSTRACT In this paper, an automated and real-time anomaly detection approach for sensors of autonomous vehicles called ARAD is presented. Automated vehicles gather environmental information through their diverse built-in sensors thus the correctness of this data affects the system's reliability, directly. Accordingly, anomaly detection schemes are employed to guarantee the correctness of the sensors' data. Moreover, due to the necessity of real-time operation in automated vehicles, the response time of the anomaly detection unit is important along with its precision. To this aim, in our proposed ARAD a lightweight and hierarchical architecture to detect and classify the anomalies based on their types is employed. Moreover, to enhance the detection capability, ARAD utilizes the data diversity property based on the sequence prediction scheme. After anomaly detection, ARAD mitigates and removes them from the system's input by its rule-based engine. To meet the precision and real-time requirements of the anomaly detection unit in autonomous vehicles, ARAD has a lightweight sequence prediction structure based on statistical and data-driven methods. To evaluate the effectiveness of our proposed ARAD, several experiments are performed and a performance measurement metric called FoC is proposed to study the contradicted effects of precision and real-time operation in terms of computation overhead, simultaneously. Based on these experiments, ARAD is capable of detecting anomalies efficiently with precision and recall of 84.6 % and 87%, respectively in real-time while applying low overhead to the system. It also shows 75.6% improvement in terms of computation cost over related methods.

INDEX TERMS Anomaly detection, autonomous vehicles, data-driven learning, fault tolerance, sensors.

I. INTRODUCTION

Along with technology advances, cyber-physical systems play important roles in various aspects such as transportation, medicine, smart homes, industrial systems and so on [1]. Automated vehicles (AVs) are introduced with technology progress in transportation and consider various communication technologies in addition to some aspects of automated control. AVs are widely used due to improving safety and decreasing accidents, human satisfaction, and their time saving along with the realization of the sustainable environment [2], [3]. Due to the extent of this concept, multiple levels of automation are presented by the society of automotive engineering and followed by researchers [17].

The mentioned advantages of AVs are tightly dependent on their precision in data gathering and processing. These data are mainly collected by various sensors that are considered for temporal, spatial, and other required information [2], [4], [23]. Thus AVs are controlled and connected by using this information and any corruption and error on them leads to unwanted catastrophic actions [5], [6]. Since these errors challenge the normal operation of the systems, they are

The associate editor coordinating the review of this manuscript and approving it for publication was Mehrdad Saif.

called "anomalies" and categorized into various classes based on their duration effect [6], [7]. Transient anomalies are momentary and have a temporary effect on the part of data, while permanent anomalies remain in the system for a long time and stop its operation. Intermittent anomalies have a short duration but occur frequently and should be predicted and managed to avoid extensive effects [7], [8], [23]. In addition to the anomalies, these systems are vulnerable to malicious cyber-attacks that should be handled [26].

Due to the importance of anomalies and their effect on operations, it is important to detect and isolate them as soon as possible. In real-time applications, the abnormal operation of the sensors propagates very fast and affects the system output. Thus their early detection and mitigation are required and affect the system performance directly [3], [8]. Traditional detection approaches are based on analyzing the statistical behavior of sensors. Since these methods model the system mathematically, provide high reliability but are widely dependent on applications and incapable of handling large data instantaneously [5], [7]. Thus statistical-based anomaly detection approaches are not appropriate candidates for modern real-time applications with a high volume of data and limited built-in computational resources. Recent research on anomaly detection is mainly based on data-driven methods that mainly utilize machine learning schemes. These approaches extract the relations and normal behavioral patterns of sensors' data to detect occurred anomalies more fast and precisely [5], [9], [21].

Deep learning-based approaches are capable and appropriate candidates that have been utilized recently to detect the anomalies of cyber-physical systems that mainly rely on sensors. Due to the unknown, heterogeneous, and diverse natures of anomalies employing deep learning to detect and classify them via neural networks seems thoroughly applicable [5], [22]. The main concerns of these methods in real-time applications such as AVs are their anomaly explanation and appliance speed. Anomalies exhibit different behaviors across various applications, making their detection and classification highly complex. Deep learning-based approaches require application-dependent feature extractions in a short time that should be precisely set in various applications. Thus real-time anomaly detection approaches are widely focused in modern applications such as AVs. These approaches mainly rely on a static threshold to classify the anomalies' type based on their duration [8], [9], [22].

In this paper, an automated data-driven and real-time anomaly detection (ARAD) is proposed that utilizes the advantages of statistical and data-driven methods, simultaneously. In this approach, first, the sensors' data are pre-processed and normalized to prepare for the learning engine. The learning engine is designed in two phases based on a lightweight deep feature extraction and analysis of the data depending on its temporal and spatial characteristics. First, the raw sensors' data is diverse and fed to the detection unit in normal and differential forms. These units determine the existence of anomalies and their results are augmented and fed to the next learning stage. The second learning module classifies the anomalies based on their behavioral patterns and duration in the system, which are reflected in the adaptive and learned thresholds, in three categories transient, intermittent, and permanent. Afterward, appropriate handling actions are applied to them based on the considered rule-based engine. This engine filters the anomalies and prevents them from the system's input by correcting or blocking them. To demonstrate the efficiency of our proposed method, several experiments on real-life benchmarks of AVs are considered. The results of these experiments show that our proposed method is capable of detecting and classifying anomalies well while forcing low overhead and complication to the target system. The rest of the paper is organized as follows. The literature review and related studies are summarized in Section II, and the details of the proposed method and experimental results and evaluations are presented in sections III and IV. Finally, section V draws the conclusion remarks and suggested trends.

II. RELATED STUDIES

Anomalies in sensor data are the main source of errors and failures in cyber-physical systems. As a result of anomaly occurrence in sensors' data, the system operates on corrupted information or misses some parts of the input. Anomalies are categorized into three main classes based on their shapes: point, contextual, and pattern [6]. Moreover, their duration effect in the system is transient, intermittent, or permanent. Independent of the anomalies' type, comparing the sensors' data to its threshold in normal operation is the main idea of anomaly detection approaches [6], [5], [8]. These approaches are classified into statistical and data-driven methods [1], [6], [7].

Statistical-based anomaly detection approaches analyze the system behavior through mathematical theories. To this aim, the collected information forms a probability distribution, and statistical operators such as mean or variance are employed to detect abnormal behaviors [10], [11]. The likelihood of the sensors' data to the derived statistical model is the measure of anomaly detection in these approaches. Along with statistical schemes, piece-wise linear models, construction Markov chain, signal processing methods, time series analysis, and employing information theory are the most effective related approaches of this field [11]. Employing these approaches provides several advantages such as high accuracy in anomaly detection due to the certainty in the derived system model. However, along with the complexity of applications in terms of their behavior and input patterns, deriving the appropriate statistical models is very complicated and requires high processing capabilities that are not available in many cases. This limitation leads the designers to the second class of methods that are based on employing learning schemes on systems' data [10], [12].

Data-driven anomaly detection approaches utilize machine learning to derive the pattern and behavior of the application's data based on its historical information. The model's reliability in these approaches is less than the previous class but is more used due to their appropriateness in modern complicated applications with limited resources [1], [5], [6]. In this class, several approaches based on supervised, non-supervised, reinforced, and deep learning are employed related to the characteristics of the target application [11], [21], [22]. Supervised data-driven anomaly detection employs a set of labeled data that determines the target data based on an expert classification. In this context, support vector machines, decision trees, and rule-based classifiers demonstrated appropriate efficiency [1], [5], [11], [21]. Since in many applications, the labeled predefined data is not available, semi and unsupervised learning schemes are introduced. Deep learning techniques like auto-encoders, Boltzmann, and Bayesian network-based schemes are examples of these class of anomaly detection approaches. Massive data demand of modern applications leads to anomaly detection to deep learning-based approaches that are supervised or unsupervised according to the collected data. Convolutions neural networks, auto-encoders, sequential networks, Generative adversarial networks, and hybrid models are the most successive related methods in various applications [1], [4], [9], [11], [20], [27], [29].

Online and real-time anomaly detection is a major requirement of designers with the growth of cyber-physical systems that are tightly reliant on sensors' data. The mentioned data-driven anomaly detection approaches mostly suffer from proper location and data processing time. Thus, proposing a real-time and lightweight anomaly detection approach is a trend in modern cyber-physical systems. To this aim, windowing, updating dynamic models, and employing short-term memories on incoming sensors' data are known as appropriate solutions! [2], [8], [11], [28]. Nowadays, proposing lightweight and efficient approaches in this field is still a requirement and should be considered in various applications based on their characteristics.

III. PRELIMINARIES

A. AUTONOMOUS VEHICLES

Autonomous vehicles (AVs) aim to replace the human driver with electro-mechanical devices at specific levels. In this context, six levels based on the level of human intervention are introduced by the Society of Automotive Engineers (SAE). The first three levels are dependent on the support of the driver alongside the automotive feature and the others completely rely on the automotive driving properties. Figure 1 shows these levels and their specifications as presented in SAE standard j3016 [15], [16], [17].

As this figure shows, at levels 0-4 the driver's action is required constantly or on-demand. However, levels 5 and 6 are completely driverless. AVs combine technology advancements in terms of processing and sensor developments to accurately sense the roadway, other vehicles, and objects on and along the roadway and generate the appropriate reaction in real-time. In this context, various research fields are presented to deal with the requirements and limitations of this area. In all described automation levels, the vehicles gather environmental data from their various built-in sensors. Thus the reliability of the sensors and the accuracy of their corresponding data is very important.

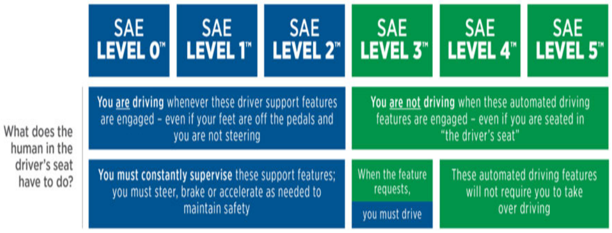

SAE LEVEL 0™
SAE LEVEL 1™
SAE LEVEL 2™
SAE LEVEL 3™
SAE LEVEL 4™
SAE LEVEL 5™
What does the human in the driver's seat have to do?
You are driving whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering
You are not driving when these automated driving features are engaged – even if you are seated in "the driver's seat"
You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety
When the feature requests,
you must drive
These automated driving features will not require you to take over driving

FIGURE 1. Levels of driving automation based on SAE standard [15].

B. SENSOR'S ANOMALY

The sensor's data are generated serially in time. An anomaly is a point or set of points with different behavior over prediction. There are several types of anomalies based on their behavior and duration in systems. Anomalies are classified into three categories point, conceptual, and pattern related to their behaviors [6], [13]. The point anomalies are unusual compared to the data but in the same pattern and occurred rarely. The conceptual anomaly affects the variation frequency of the data but is in the normal range which does not lead to data inconsistency. Lastly, the pattern anomaly is the most complicated one and is detectable based on its inconsistency with data behavior in time. This kind of anomaly can't be detected before extracting the normal behavior of the system [6], [23].

Moreover, the sensors' anomalies have various time durations. Transient anomalies are very short in time and their effect diminishes soon. However, the permanent anomalies are persistent and have long-term effects on the system. These anomalies lead to the system crash in time and their effect should be eliminated. Intermittent faults occur repeatedly in the system in several instances and their behavior in each sample is similar to the transient faults [2], [13]. Based on the application, all anomaly types could occur during the system execution. The output of a GPS sensor injected by the mentioned anomalies is presented in Fig. 2.

C. LONG SHORT-TERM MEMORY NETWORKS (LSTM)

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network that can learn sequences deeply for their upcoming prediction. This capability is provided by employing the feedback connections that focus on the entire sequence of data to process. This feature makes the LSTM networks an appropriate candidate for predicting patterns in sequential data like time series, text, and speech [18], [19].

The architecture of this network consists of three main elements forget, input, and output gates along with the memory cell. The first gate determines the necessity of

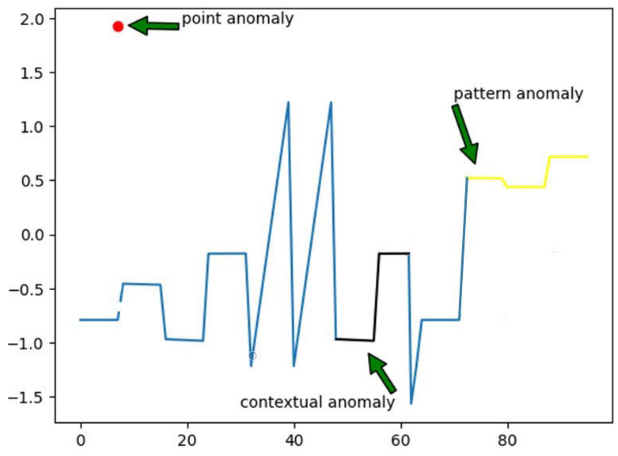


FIGURE 2. Sample output of a GPS sensor with various types of anomalies in time.

saving the corresponding data of the previous time step (h(t-1)). To this aim, a sigmoid activation function is employed to decide whether to keep or discard this information. Equation 1 shows the details of computation in this gate.

$$f_t = \sigma(x_t \times U_f + H_{t-1} \times W_f) \tag{1}$$

where, X_t is the input of the current time step, U_f is the corresponding weight of input, H_{t-1} is the hidden state of the previous step and W_f represents its corresponding weight. After applying the sigmoid function, the f_t maps to zero or one and determines keeping or discarding the data of the previous time step. The second gate is input and is employed to determine the importance of the current information carried by the input. The equation of this gate is defined as follows:

$$i_t = \sigma(x_t \times U_i + H_{t-1} \times W_i) \tag{2}$$

where U_i is the weight matrix of input and W_i represents the weight matrix of input associated with hidden states. In this gate again, a sigmoid function is applied to map its result in the range [0,1]. Finally, the output gate determines the output of the current step based on its input and previous state. To this aim the following equation is employed:

$$o_t = \sigma(x_t \times U_o + H_{t-1} \times W_o) \tag{3}$$

where U_o is the weight matrix of output and W_i represents the weight matrix of output associated with hidden states. Similarly, a sigmoid function is applied to map the result of this gate in the range [0,1]. Last, the tanh activation function is applied to the result of the output gate to update the cell state. Figure 3 shows the architecture of an LSTM cell considering its gate and working flow.

LSTM is capable of storing long-term information without being affected by the current input or output due to its separate memory cell. Thus, this feature could be employed in sequence prediction while adjusting its required cells based on the problem scale.

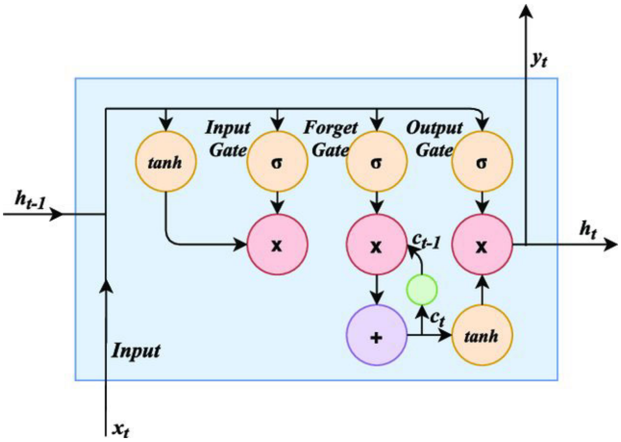


FIGURE 3. The structure of the LSTM cell considering its gate and working flow [18].

IV. PROPOSED METHOD

Our proposed approach aims to detect and mitigate the anomalies of sensors in automated vehicles during their operation. To this goal, various types of anomalies based on their time duration and behavior in the system are considered. Since the proposed scheme is considered for real-time operation it should be light weighed and react fast. These features are not compatible with deep learning-based schemes that are very suitable for anomaly detection in large data volumes such as the outputs of sensors in automated cars. To handle this, we have proposed a data-driven approach that employs a lightweight anomaly detection engine based on an LSTM auto encoder along with a dynamic rule engine. Moreover, the temporal and spatial aspects of data are considered hierarchically. Our proposed learning engine applies to the system in two steps, first, the anomalies are identified and then they are classified based on their impact duration in the system. This data-driven detection engine performs based on dynamic threshold values that are derived and adjusted dynamically during system operation depending on the effects of anomalies on data. Afterward, the designed rule-based engine mitigates the detected anomalies based on their types and anomaly-free data are sent to the processing core of the system. Fig 4 shows the block diagram of our proposed ARAD.

As this figure shows, first the raw sensors' data are fed to our proposed anomaly detection architecture. These data include various types of anomalies and should be pre-processed to remove ambiguous parts and convert them to an understandable format. In this phase, the raw data are filtered, cleaned, and ordered to be prepared for the learning phase, and to reduce the data scattering it standardizes in the range [0,1]. Since the target data is extracted from sensors in the form of a time series, it should be framed appropriately. Small frames disturb the data dependency feature and large frames enforce complicated analysis. To this aim, we have performed several experiments and divided the data into a frame size of eight.

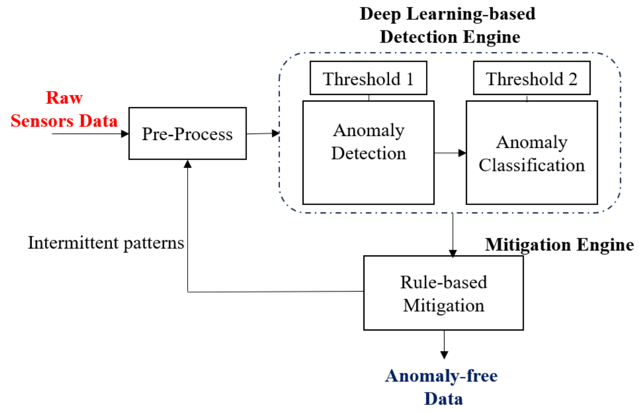


FIGURE 4. Block diagram of our proposed ARAD.

Afterward, the pre-processed data is passed to the proposed detection engine. This engine has two parts: the first detects abnormal data, and the second classifies the anomalies based on their duration in the system. Due to the hierarchical architecture of the proposed detection scheme, its complexity and timing overhead have been managed to be utilized in embedded real-time applications such as self-driving cars. The first stage of our proposed detection engine consists of a lightweight LSTM due to the time series form of the sensor's data. This scheme makes it possible to predict future data from the history and detect abnormal behaviors that are not matched to the expected pattern. This phase classifies the sensors' data into normal and abnormal categories. Due to the criticality of this detection and its effect on the performance of the ahead steps, we have combined it with a redundant structure. This unit gets the sensors' data in the differential format and extracts their existing abnormal behaviors. Thus, by considering data diversity in detection units their performance has been improved. These detection units that are fed with original and differential data of sensors are performed in parallel and their results are compared to determine the abnormal data of the system. The second unit is based on the LSTM structure the same as the primal one but summarized and light-weigh to manage the system's overhead and cost. Since the input data of the detection unit are extracted from sensors sequentially its changes over time are limited and smooth which makes the range of their differential values very narrow. Thus, considering a lightweight LSTM structure for the second redundant unit that processes the differential data with limited variation to detect abnormal behaviors is appropriate. Figure 5 shows the architecture of our proposed detection engine.

As this figure shows the original and differential sensors' data are passed to the detection unit to operate the primal and redundant detection modules. The results of these modules are compared to predefined thresholds and then connected to derive the final output. To determine the final output, equal weights are considered for the primal and redundant modules but this could be learned and adjusted. The thresholds are determined based on a learning approach and are fine-tuned during the system execution. It should be noted that the initial value of the thresholds are derived by intersecting the precision and recall and then they are adjusted through learning.

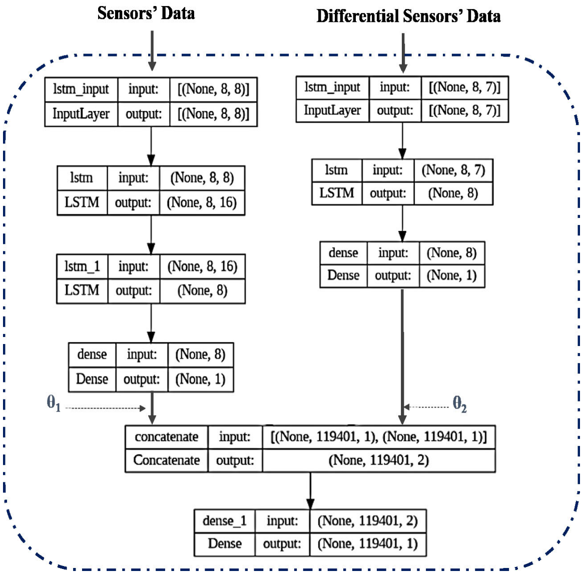

Sensors' Data
Differential Sensors' Data
lstm_input | input: | [(None, 8, 8)]
InputLayer | output: | [(None, 8, 8)]
lstm | input: | (None, 8, 8)
LSTM | output: | (None, 8, 16)
lstm_1 | input: | (None, 8, 16)
LSTM | output: | (None, 8)
dense | input: | (None, 8)
Dense | output: | (None, 1)
lstm_input | input: | [(None, 8, 7)]
InputLayer | output: | [(None, 8, 7)]
lstm | input: | (None, 8, 7)
LSTM | output: | (None, 8)
dense | input: | (None, 8)
Dense | output: | (None, 1)
θ_1
θ_2
concatenate | input: | [(None, 119401, 1), (None, 119401, 1)]
Concatenate | output: | (None, 119401, 2)
dense_1 | input: | (None, 119401, 2)
Dense | output: | (None, 119401, 1)

FIGURE 5. Architecture of ARAD's detection unit ($\theta_1 and \theta_2$ are the threshold values of the primal and redundant detection modules).

In the third step, the abnormal data are classified into transient, intermittent, and permanent errors based on their behavior and duration length. To this aim, the detected abnormal data are passed from the explained detection unit to the anomaly classifier module. The structure of this module is based on the LSTM due to our data format which is time series. Due to the previous classification of the data and limiting the process to abnormal data, the considered LSTM architecture is lightweight with three layers of 16-8-3 neurons. The final layer of this unit classifies the anomalies based on their repetition and duration length by probabilistic selection.

Finally after classifying the anomalies, they are handled based on their types in the rule-based mitigation engine. This unit consists of confrontational and recursive actions to remove the erroneous data and filter them earlier in their ahead occurrences. These strategies are determined based on the detected anomaly types. In this context, the transient errors are corrected by time series interpolation extracted by the past and predicted future samples. In case of intermittent errors in addition to their correction, the corresponding patterns are added to the pre-processing unit to be filtered and avoided earlier. Moreover, regarding the permanent errors due to their durability in the system and irreparable consequences, the system stops the operation and resets to activate the backup solutions. As a result of this phase, the cleaned data is passed to the system or its operation

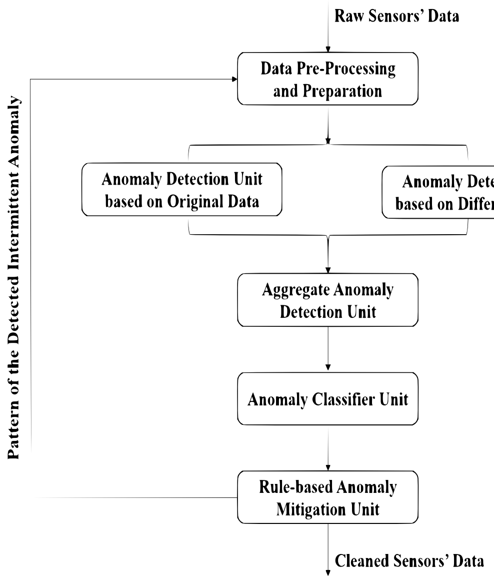

Raw Sensors' Data
Data Pre-Processing and Preparation
Anomaly Detection Unit based on Original Data
Anomaly Detection Unit based on Differential Data
Aggregate Anomaly Detection Unit
Anomaly Classifier Unit
Rule-based Anomaly Mitigation Unit
Cleaned Sensors' Data
Pattern of the Detected Intermittent Anomaly

FIGURE 6. The block diagram of our proposed ARAD based on its operational stages.

enters the pre-defined safe mode. The block diagram of our proposed method based on its operational steps is presented in Fig. 6.

V. EXPERIMENTAL RESULTS

To evaluate our proposed approach to detecting and mitigating anomalies effectively in real-time, in this section the setup description, analysis of the features and capabilities of the proposed ARAD, and its comparison to related studies are presented.

A. SIMULATION SETUP AND DATASET DESCRIPTION

The proposed anomaly detection approach is simulated with Python and in the Google Colaboratory environment. In this context, the Intel(R) Xeon(R) CPU @ 2.20GHz and NVIDIA Tesla K80 with 12GB of VRAM processors are utilized. Since the proposed method targets at sensor data, we have employed the Multi-Modal Intelligent Traffic Signal Systems GPS (MMITSS) dataset. This dataset is collected by the multi-modal intelligent transportation signal systems along with the vehicle's GPS to provide performance and operation details of vehicles over time. This dataset describes the vehicle's position and speed, fidelity measures of GPS-based data elements, and vehicle operation data. This dataset includes more than 4.7 million entries that are collected over time and presents 17 various features for them. Some of the most important features of MMITSS are index, vehicle ID, lap ID, time, date, moving direction, latitude, longitude, height, speed, and source of data collection. These entries are in the form of a multi-modal time series that are updated each 100ms over several days [14].

Since our proposed method aims at detecting the anomalies, our data set should contain errors and abnormal behaviors. The MMITSS dataset is clear and shows the normal behavior of some test vehicles that are collected by GPS sensors. Thus, it is required to inject various types of anomalies into it. In this context, we have injected transient, intermittent, and permanent errors into the data considering their duration and behavior. The anomalies are dispensed based on Gaussian distribution and their duration is set experimentally. We assume that transient and permanent anomalies last about 28 to 56 seconds and 180 to 240 seconds, consequently. Moreover, the period of intermittent errors is assumed to be about 270 to 330 seconds. The distances of these anomalies and normal data are determined by uniform distribution and the portion of the abnormal data in the dataset is set to 64% such that covering all anomaly types. Based on this mentioned procedure, our considered dataset is labeled for the proposed learning procedure. It should be noted that the defined injection process based on statistical behaviors of the anomalies is followed from previous research of this field [2], [8].

Afterward, the data should be prepared and segmented to remove the redundant information and make it ready for the detection learning engine. To prepare the data, we have summarized the features as vehicle ID, time, moving direction, latitude, longitude, height, speed, and source. Then the data is segmented into windows of size eight and reshaped in three dimensions (samples, timestamps, features) to be prepared for the LSTM structure. The window size is derived experimentally to compromise the accuracy and real-time operation. Last, due to the large variation range of the data, they are standardized by their standard deviation and mean. It should be mentioned that the data split 70-30% between the train and test phases of the proposed learning schemes.

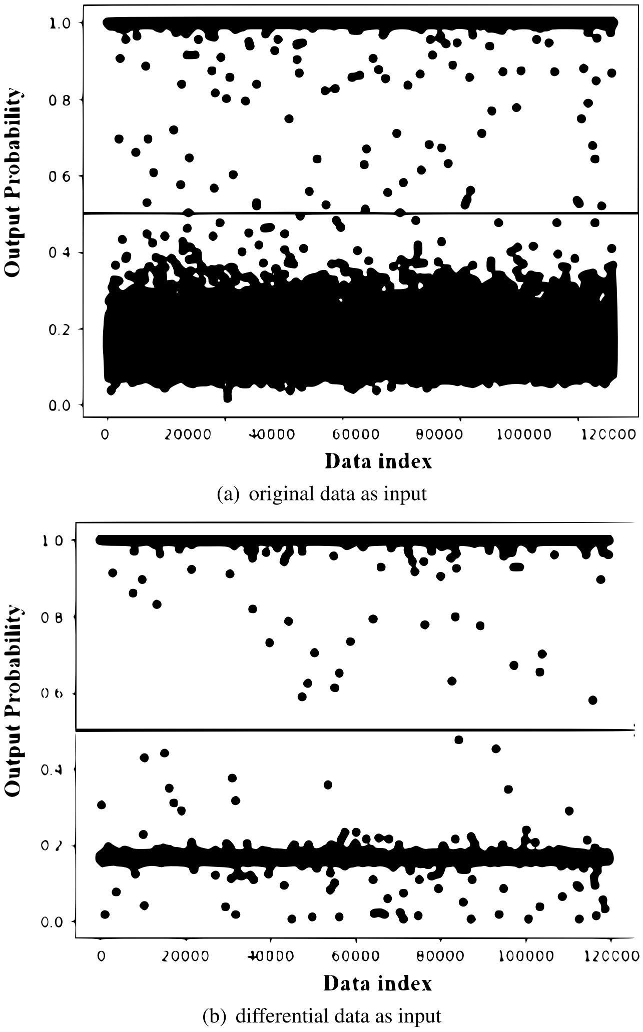
B. EXPERIMENTAL RESULTS

To evaluate our proposed anomaly detection method, the effectiveness of its units in terms of accuracy in the detection, classification, and mitigation of anomalies is first studied. Moreover, since the aim of this method is real-time application its overhead is important so timing analysis is performed on it. Finally, the performance of our proposed method is compared to related research in terms of the detection capability and enforced overhead.

1) EVALUATING THE EFFICIENCY OF THE PROPOSED DETECTION UNIT

Our proposed ARAD passes the pre-processed sensors' data to the detection unit to determine the existence of anomalies. This unit consists of two parallel learning units fed with original and differential data their results are aggregated and produce the final decision. The probabilistic output of the detection units with original and differential data is shown in Fig. 7.

As this figure shows, the probabilistic output of the detection unit in its two modules that are fed with original



(a) original data as input

(b) differential data as input

FIGURE 7. The probabilistic output of the detection units.

and differential data are totally different. Thus, considering this data variation improves the accuracy of this unit and makes its classification more precise. It should be noted that the sparseness of the probabilistic output of the detection unit based on differential data is due to the limited range of variation in this case. The illustrated probabilistic output of these modules should be mapped to binary classes of abnormal or normal behaviors. In this context, a threshold value is considered that we have determined experimentally in this step.

Afterward, the results of the mentioned modules of the detection unit are aggregated to a non-sequential structure that compares them and performs weighted voting between them based on leaned and dynamic threshold values. To determine the appropriate initial threshold value of this unit, the precision and recall parameters are derived for various thresholds, and the value corresponding to their intersection point is selected. Then a simple neural network consisting of a single neuron is employed to adjust the selected threshold value based on the false negative and false positive points.

TABLE 1. Evaluated parameters of the anomaly detection unit and its modules.

		Precision	Recall	F1-score
Primal module (original data)	Normal Data	84%	89%	88%
	Abnormal Data	96%	71%	76%
Redundant module (differential data)	Normal Data	88%	82%	90%
	Abnormal Data	92%	77%	74%
Detection Unit (Aggregated model)	Normal Data	90%	95%	94%
	Abnormal Data	95%	86%	87%

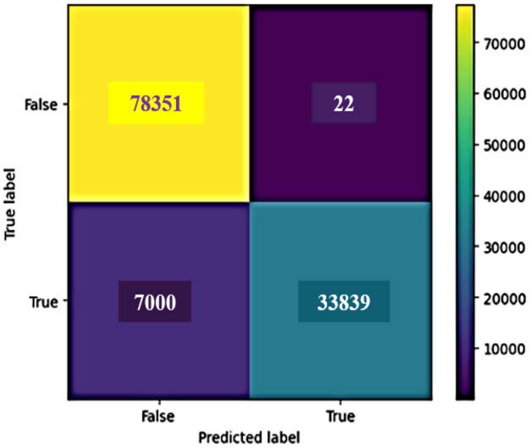


FIGURE 8. Confusion matrix of our proposed anomaly detection unit.

TABLE 2. Evaluated parameters of the anomaly classification unit.

	Precision	Recall	F1-score
Transient Error	91%	84%	89%
Intermittent Error	90%	84%	89%
Permanent Error	73%	93%	79%

The evaluated parameters of the detection unit considering its modules are summarized in Table 1.

Based on these results, augmenting the results of two detection modules enhances the output in terms of detection capability. The detection unit performs well with precision 95% in finding the abnormal data and separating it from normal and healthy ones. Moreover, to demonstrate the effectiveness of the detection unit its confusion matrix is presented in Fig. 8.

As this figure shows, the detection capability of our proposed unit is very good and its miss-classification rate is about 5%. Moreover, the false positive is smaller than the false negative limits the useless anomaly handling, and saves the system's operation time.

After detecting the anomalies, they should be classified based on their types. To this aim the anomalies are classified into three classes transient, intermittent, and permanent based on their pattern and duration. The evaluated parameters of this unit are presented in table 2.

As this table shows, our proposed classification unit is capable of finding anomaly types with acceptable precision. The precision of detecting permanent anomalies is less than the other types due to its large duration and the limitation

TABLE 3. **The detailed overhead analysis of our proposed ARAD in terms of FoC parameter.**

Model	Number of neurons	F1-score	FoC
Anomaly detection using original data	3 layers 16-8-1	85%	0.034
Anomaly detection using differential data	2 layers 8-1	86%	0.095
Augmented Anomaly detection unit	3 layers 1-2-1	92%	0.23
Anomaly Classification unit	3 layers 16-8-3	83%	0.03
Total	56	85	1.51

of the prediction window. However, the miss-classification rate of permanent anomaly is about 15% and its majority (14.2%) is caused by the false positive predictions that are less harmful but may waste time. Moreover, this parameter for transient and intermittent faults is estimated at 2% and 3%, respectively.

The mitigation rule engine performs the appropriate action on the detected anomalies based on their types. In this context, due to the lower detection rate of permanent anomalies, more rules are considered in this engine to reduce the false positive rate during its mitigation and save the system's time.

2) OVERHEAD ANALYSIS OF OUR PROPOSED ARAD

Since the proposed anomaly detection aims at real-time applications such as autonomous vehicles, its complexity, and overhead should be very limited. Moreover, this scheme should be precise and reliable to avoid anomalies in the system. These factors are contradicted and usually, precise computations enforce a heavy load on the system that is not acceptable in real-time applications. To this aim, we have introduced a parameter that considers the precision, recall, and computation cost of the detection method simultaneously. This parameter is called FoC for F1-score over computation and its large value is wanted during the detection process. To estimate the computation cost, the number of learning neurons of the detection methods is considered.

The computation overhead of our proposed method based on its learning neurons is presented in Table 3. Based on this table, our proposed method consists of four LSTM models for the detection and classification of anomalies. The third first ones have 39 neurons considering the one that is added to learn the proper threshold value for anomaly detection and the last one has 27 neurons of their classification.

3) COMPARISON TO RELATED RESEARCH

To evaluate the effectiveness of our proposed method, we have compared it to related research in terms of the detection capability and enforced overhead. The anomaly detection approach of [8] is proposed based on LSTM and CNN structures to have real-time operation. The considered LSTM of this approach consists of three layers with 256-128-64 neurons and its output is passed to a CNN. The overhead and time complexity of this method is high due to its architecture but in cost of more precision in anomaly detection. The comparison of our proposed method to the real-time anomaly detection method of [8] in terms of precision, recall, and accuracy is presented in Fig. 9. In this figure, the results are averaged for various types of anomalies.

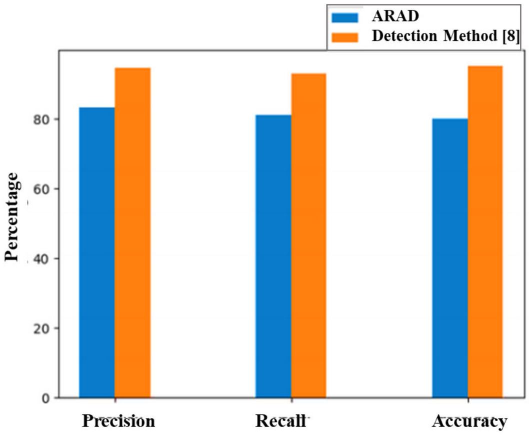


FIGURE 9. **The detection capability of our proposed ARAD over the proposed method of [8] in terms of precision, recall, and accuracy.**

TABLE 4. **Comparison of our proposed anomaly detection to related methods in terms of the detection capability, computation cost, and FoC.**

	F1-score	Computation Cost	FoC
Detection method of [8]	91.4%	2950	0.003
LSTM-based method inspired of [24], [25]	78.06%	160	0.48
CNN-based method inspired of [2]	84.4%	1124	0.007
ARAD	81%	56	**1.44**

As this figure shows, the capability of the proposed method of [8] in anomaly detection is about 82% and 7% less than the detection method of [8]. However, in real-time applications, accuracy is important along with the response time. If the target system provides the best accuracy but in cost of a long time it is not efficient. Thus, we have compared our proposed ARAD to the detection method of [8] in terms of its accuracy and computation cost simultaneously by employing the presented FoC measure. Table 4 shows the results of this comparison.

As this table shows, our proposed ARAD outperforms the real-time detection methods of [8], LSTM-based inspired of [24] and [25], and CNN-based inspired of [2] in terms of the FoC parameter. In this table, the computation cost is considered as the number of neurons in the considered methods. The number of neurons in our proposed model is computed in detail in Table 3. The method of [8] consists of an LSTM with 256-128-64-64-128-256 neurons and a CNN architecture with about 28675 neurons. The considered auto encoder-based model inspired of [24] and [25] is considered of four 32 hidden layers along with input and output layers. Moreover, the assumed CNN-based model inspired of [2] consists of 11243 neurons based on its input shapes. Our proposed method meets the trade-off between the detection capability and computation overhead better due to its higher value of FoC and is appropriate for employment in real-time applications such as autonomous vehicles.

VI. CONCLUSION REMARKS

Autonomous vehicles are introduced along with technology advances in cyber-physical systems to facilitate driving. These vehicles are sensing the environment and operating without human involvement. Thus the sensing data that are provided by their built-in sensors must be diverse and reliable. In this context, various sensors are developed to gather all required information from the environment. Moreover, the accuracy of this data is analyzed precisely during the anomaly detection process. The anomaly detection aims at finding the abnormal behavior of the sensor's data by comparing them with the predicted pattern continuously. In this paper, a real-time sensor anomaly detection approach called ARAD is presented. This method is capable of detecting anomalies during the system execution and classifying them into three categories transient, permanent, and intermittent depending on their duration. In addition, in ARAD a rule-based mitigation engine is considered to handle the detected anomalies and clean the input data of the system. Since our target application is critical, its instantaneous and reliable operation despite the huge volume of data provided by the various sensors is required. Thus, our proposed method is designed based on a lightweight and hierarchical architecture. In ARAD the LSTM network is employed to predict the data sequence and detect the abnormal behavior of sensors' data. To avoid the computation overhead of this structure, we have employed the hierarchy and data diversity properties to have a lightweight LSTM architecture capable of detecting abnormal data in real time. Afterward, the detected anomalies are classified based on their duration to apply appropriate handling action to them. Finally, the rule-based mitigation engine employs confrontational actions to clean the anomalies and correct or block them before the system's input. Based on the experimental results, our proposed method is capable of detecting 83% of the anomalies at low computation cost and timing overhead. Moreover, we have introduced a performance evaluation measure called FoC that combines the detection capability and computation cost in terms of the model's neurons count as two important parameters in real-time anomaly detection methods. Based on FoC our proposed ARAD strongly outperforms the detection method of [8] that is most related. As the future trend, improving the feature extraction process to dynamically determine the anomaly threshold of the data more precisely along with employing attention mechanisms to refine the detection capability of intermittent errors are considered.

REFERENCES

[1] Y. Luo, Y. Xiao, L. Cheng, G. Peng, and D. Yao, "Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities," *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1–36, Jun. 2022.

[2] F. van Wyk, Y. Wang, A. Khojandi, and N. Masoud, "Real-time sensor anomaly detection and identification in automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1264–1276, Mar. 2020.

[3] S. E. Shladover, "Connected and automated vehicle systems: Introduction and overview," *J. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 190–200, May 2018.

[4] A. Sarker, H. Shen, M. Rahman, M. Chowdhury, K. Dey, F. Li, Y. Wang, and H. S. Narman, "A review of sensing and communication, human factors, and controller aspects for information-aware connected and automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 7–29, Jan. 2020.

[5] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, 2021.

[6] A. A. Cook, G. Misirli, and Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020.

[7] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009.

[8] R. Oucheikh, M. Fri, F. Fedouaki, and M. Hain, "Deep real-time anomaly detection for connected autonomous vehicles," *Proc. Comput. Sci.*, vol. 177, pp. 456–461, Jan. 2020.

[9] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller, "A unifying review of deep and shallow anomaly detection," *Proc. IEEE*, vol. 109, no. 5, pp. 756–795, May 2021.

[10] H. Mohammadi Rouzbahani, H. Karimipour, A. Rahimnejad, A. Dehghantanha, and G. Srivastava, "Anomaly detection in cyber-physical systems using machine learning," in *Handbook of Big Data Privacy*. New York, NY, USA: Springer, 2020.

[11] L. Erhan, M. Ndubuaku, M. Di Mauro, W. Song, M. Chen, G. Fortino, O. Bagdasar, and A. Liotta, "Smart anomaly detection in sensor systems: A multi-perspective review," *Inf. Fusion*, vol. 67, pp. 64–79, Mar. 2021.

[12] A. Sgueglia, A. Di Sorbo, C. A. Visaggio, and G. Canfora, "A systematic literature review of IoT time series anomaly detection solutions," *Future Gener. Comput. Syst.*, vol. 134, pp. 170–186, Sep. 2022.

[13] A. Chatterjee and B. S. Ahmed, "IoT anomaly detection methods and applications: A survey," *Internet Things*, vol. 19, Aug. 2022, Art. no. 100568.

[14] (2023). *Multi-Modal Intelligent Traffic Signal Systems GPS*. Accessed: Apr. 10, 2024. [Online]. Available: https://datahub.transportation.gov/Automobiles/Multi-Modal-Intelligent-Traffic-Signal-Systems-GPS/2f79-bkh3/data

[15] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, On-Road Automated Driving (ORAD) Committee, SAE Int., Warrendale, PA, USA, 2021.

[16] A. Faisal, T. Yigitcanlar, M. Kamruzzaman, and G. Currie, "Understanding autonomous vehicles," *J. Transp. Land Use*, vol. 12, no. 1, pp. 45–72, Jan. 2019.

[17] "Autonomous vehicles factsheet," Center Sustain. Syst., Univ. Michigan, Tech. Rep. CSS16-18, 2023. [Online]. Available: https://css.umich.edu/publications/factsheets/mobility/autonomous-vehicles-factsheet

[18] I. K. Ihianle, A. O. Nwajana, S. H. Ebenuwa, R. I. Otuka, K. Owa, and M. O. Orisatoki, "A deep learning approach for human activities recognition from multimodal sensing devices," *IEEE Access*, vol. 8, pp. 179028–179038, 2020.

[19] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306.

[20] M. Gutowska, S. Little, and A. Mccarren, "Constructing a meta-learner for unsupervised anomaly detection," *IEEE Access*, vol. 11, pp. 45815–45825, 2023.

[21] A. B. Nassif, M. A. Talib, Q. Nasir, and F. M. Dakalbab, "Machine learning for anomaly detection: A systematic review," *IEEE Access*, vol. 9, pp. 78658–78700, 2021.

[22] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines," *IEEE Access*, vol. 9, pp. 120043–120065, 2021.

[23] S. T. Banafshehvaragh and A. M. Rahmani, "Intrusion, anomaly, and attack detection in smart vehicles," *Microprocessors Microsyst.*, vol. 96, Feb. 2023, Art. no. 104726.

[24] R. Sun, Q. Luo, and Y. Chen, "Online transportation network cyber-attack detection based on stationary sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 149, Apr. 2023, Art. no. 104058.

[25] F. W. Alsaade and M. H. Al-Adhaileh, "Cyber attack detection for self-driving vehicle networks using deep autoencoder algorithms," *Sensors*, vol. 23, no. 8, p. 4086, Apr. 2023.

[26] E. Eziama, F. Awin, S. Ahmed, L. M. Santos-Jaimes, A. Pelumi, and D. Corral-De-Witt, "Detection and identification of malicious cyber-attacks in connected and automated vehicles' real-time sensors," *Appl. Sci.*, vol. 10, no. 21, p. 7833, Nov. 2020.

[27] P. Mansourian, N. Zhang, A. Jaekel, and M. Kneppers, "Deep learning-based anomaly detection for connected autonomous vehicles using spatiotemporal information," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 16006–16017, Dec. 2023.

[28] S. Rezaei, N. Masoud, and A. Khojandi, "GAAD: GAN-enabled autoencoder for real-time sensor anomaly detection and recovery in autonomous driving," *IEEE Sensors J.*, vol. 24, no. 7, pp. 11734–11742, Apr. 2024.

[29] S. Rajendar and V. K. Kaliappan, "Sensor data based anomaly detection in autonomous vehicles using modified convolutional neural network," *Intell. Autom. Soft Comput.*, vol. 32, no. 2, pp. 859–875, 2022.

ATHENA ABDI received the B.Sc., M.Sc., and Ph.D. degrees from the Department of Computer Engineering, Amirkabir University of Technology (AUT), Tehran, Iran, in 2010, 2012, and 2018, respectively. She spent a year as a Visiting Researcher with the SPADES Team, INRIA, Rhone-Alpes, France. She was a Postdoctoral Research Fellow with the Design and Analysis of Dependable Systems (DADS) Laboratory, AUT, from 2019 to 2020. She has been an Assistant Professor with the Faculty of Computer Engineering, K. N. Toosi University of Technology (KNTU), since 2020. Her research interests include embedded and real-time systems, scheduling, optimization, and fault-tolerant design in computer architecture.

ARASH GHASEMI-TABAR received the B.Sc. degree from the Islamic Azad University of Tehran and the M.Sc. degree from the K. N. Toosi University of Technology (KNTU). His research interests include cyber-physical systems and their design challenges, fault-tolerant design, automotive vehicles, sensors' anomalies, and light-weight deep learning schemes.