Anomaly Detection in Automated Vehicles Using Multistage Attention-Based Convolutional Neural Network

Abdul Rehman Javed, Muhammad Usman, Saif Ur Rehman, Mohib Ullah Khan, and Mohammad Sayad Haghighi, *Senior Member, IEEE*

***Abstract*— Connected and Automated Vehicles (CAVs), owing to their characteristics such as seamless and real-time transfer of data, are imperative infrastructural advancements to realize the emerging smart world. The sensor-generated data are, however, vulnerable to anomalies caused due to faults, errors, and/or cyberattacks, which may cause accidents resulting in fatal casualties. To help in avoiding such situations by timely detecting anomalies, this study proposes an anomaly detection method that incorporates a combination of a multi-stage attention mechanism with a Long Short-Term Memory (LSTM)-based Convolutional Neural Network (CNN), namely, MSALSTM-CNN. The data streams, in the proposed method, are converted into vectors and then processed for anomaly detection. We also designed a method, namely, weight-adjusted fine-tuned ensemble: WAVED, which works on the principle of average predicted probability of multiple classifiers to detect anomalies in CAVs and benchmark the performance of the MSALSTM-CNN method. The MSALSTM-CNN method effectively enhances the anomaly detection rate in both low and high magnitude cases of anomalous instances in the dataset with the gain of up to 2.54% in F-score for detecting different single anomaly types. The method achieves the gain of up to 3.24% in F-score in the case of detecting mixed anomaly types. The experiment results show that the MSALSTM-CNN method achieves promising performance gain for both single and mixed multi-source anomaly types as compared to the state-of-the-art and benchmark methods.**

***Index Terms*— Anomaly detection, connected and automated vehicles (CAVs), convolutional neural network (CNN), intelligent transportation system (ITS), multi-source anomaly detection.**

I. Introduction

CONNECTED and Automated vehicles (CAVs) technology is among the most focused research areas because of its potential for decreasing the likelihood of accidents, enhancing human satisfaction, maintaining a sustainable environment, and improving the effectiveness of Intelligent Transportation System (ITS) [1]. The ITS ecosystem is an amalgamation of several communication channels such as Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Cloud (V2C) [2]. According to a study on connected vehicles and cybersecurity, it is expected that CAV technology will reduce vehicle crashes by 80%, as well as reduces the 6.9 billion hours that Americans spend in traffic annually [3]. However, this depends upon the provision of appropriately designed data-driven and associated services offered by CAVs [4]–[6]. The correct functioning of these services is heavily dependent on the accuracy and quality of data [7], [8]. The erroneous and anomalous readings, generated through errors or attacks, can disrupt key functionalities such as acceleration, current speed, current position, braking, and adaptive controls. Moreover, some difficult situations such as multiple sorts of traffic, transfer stations, and some harsh weather conditions can happen [9]. Therefore, designing and analyzing an appropriate anomaly detection method is critical for CAVs.

Several anomaly detection and classification methods have been studied in the literature, primarily employing the premise of artificial intelligence [1], [4], [10]. A study carried out on a critical examination of the probability of the crash occurrence due to cyberattacks [11]. The methodologies also support IoT frameworks that rely on cyber-physical systems [12]. These methods, however, have limitations such as (i) poor detection performance over lower magnitudes of anomalous data and for single as well as mixed anomaly types using deep learning methods in CAVs and (ii) absence of a suitable ensemble machine learning approach to substantiate the performance of the method designed to detect anomalies addressing limitation (i). To effectively detect various types of anomalies in CAVs by addressing the above-cited limitations, maintain a strategic check for detecting a dynamic behavioral change, and prevent the effect of irregular readings, this article makes the following contributions:

- We design a method that comprises of multi-stage attention mechanism with Long Short-Term Memory (LSTM)-based fine-tuned Convolutional Neural Network (CNN), namely, MSALSTM-CNN to classify multi-source sensor readings as anomalous or normal. The MSALSTM-CNN method focuses on different parts of streaming input data to learn paying attention to only the significant parts.

Manuscript received July 3, 2020; revised September 1, 2020; accepted September 16, 2020. The Associate Editor for this article was A. Jolfaei. *(Corresponding author: Muhammad Usman.)*

Abdul Rehman Javed is with the National Center for Cyber Security, Air University, Islamabad 44000, Pakistan (e-mail: abdurrahman.j74@gmail.com).

Muhammad Usman is with the Faculty of Computing, Engineering and Science, University of South Wales, Pontypridd CF37 1DL, U.K. (e-mail: muhammad.usman@southwales.ac.uk).

Saif Ur Rehman is with the Faculty of Computing and AI, Air University, Islamabad 44000, Pakistan (e-mail: 181065@students.au.edu.pk).

Mohib Ullah Khan is with the Department of Computer Science, University of Wah, Wah Cantonment 47000, Pakistan (e-mail: mohib_khn@outlook.com).

Mohammad Sayad Haghighi is with the School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran 1417466191, Iran (e-mail: sayad@ut.ac.ir).

Digital Object Identifier 10.1109/TITS.2020.3025875

- We also design a method, namely, WAVED, that comprises a weight-adjusted ensemble of distinct classifiers. The method is designed to detect anomalies in multi-sensor streams of data by using the optimal weight vector of classifiers to assign unique voting weight to the prediction of each classifier to detect anomalous behavior.
- The results of the experiments demonstrate that the MSALSTM-CNN method effectively enhances the anomaly detection rate, both generally and in the case when there is lower magnitude of anomalous data instances in the input streams. The MSALSTM-CNN method achieves the gains of 2.54%, 1.85%, 1.36%, and 0.37% in F-score for detecting instant, constant, gradual-drift, and Bias anomalies in the case of single anomaly types as compared to the baseline approach [1]. The MSALSTM-CNN method achieves the gains of 3.24%, 1.79%, 1.63%, and 1.15% in F-score for detecting instant, constant, gradual-drift, and bias anomalies in the case of mixed anomaly types. This makes the detection process robust; hence, decreasing the likelihood of fatal accidents that could originate due to anomalous data.

The rest of the paper is organized as follows. Section II offers a brief overview of the state-of-the-art related work. Section III provides the details of the selected network model, dataset, and other preliminaries. Section IV formulates the problem and elucidates our proposed anomaly detection methods. The evaluation criteria and results of the proposed methods are presented in Section V. Finally, Section VI concludes the work of this article and highlights the future direction.

II. Related Work

Wyk *et al.* studied a method of anomaly detection in CAVS by employing CNN along with Kalman-Filter with an $X2$ detector [1]. Lee *et al.* presented an algorithm to detect attacks on vehicular networks using Kalman-filter [10]. Basiri *et al.* proposed two algorithms, namely, Rolling Window Detector (RWD) and Novel Residual Detector (NRD), based on Kalman-filter for the detection of real-time attacks and attack-free states in CAVs [13]. A novel observer-based method was proposed by Y. Wang *et al.* to improve the security of CAVs by utilizing an adjustable extended Kalman filter [4]. Y. Wang *et al.* proposed to use an augmented extended Kalman filter to stabilize sensor readings in a nonlinear car-following motion model through time delay [2]. The anomaly detection in vehicles is also studied [12]. Ruoying *et al.* studied effectiveness of deep learning techniques for anomaly detection [14]. The path of the next vehicle is utilized by the subject vehicle to distinguish sensor irregularities. A standard $X2$ fault detector is formulated in aggregation with Adaptive Extended Kalman filter (AEKF) for anomaly discovery [2]. B. Du *et al.* proposed a deep convolutional residual LSTM network model for city traffic passenger classification [9].

Kordestani *et al.* proposed a sensor-based security management system (SMS) for an industrial steam turbine using deep learning techniques [15]. Mozaffari *et al.* studied the efficiency of cyber-physical system (CPS) using machine and deep learning models for the detection of anomaly readings [16]. The review of the related work shows that deep learning methods with LSTM and Kalman filter have been explored. However, to our knowledge, the multistage attention-based CNN has not been studied in CAVs for anomaly detection. Moreover, The dataset we used in this study is examined for anomaly detection by Wyk *et al.* [1]. The dataset contains different types of values with low and high magnitudes. Unlike other studies, we test our methods on different configurations of dataset anomalies for different anomaly types by comparing it with the baseline study. Yet our method clarifies that normalization of data along-with model tuning yields better performance. The results, unlike previous studies [1], [2], [11], show that our proposed method is robust in detecting anomalies even when magnitude of anomalous instances is less in the training dataset. Another major limitation is lack of research-based evidence that examines the utilization of machine learning-based ensemble classifier to substantiate the performance of CNN-based anomaly detection methods [2], [4], [11].

To bridge the above-cited research gaps, we first present the MSALSTM-CNN method that can effectively detect anomalies with low magnitude to assure the high reliability of the fused data. In addition to this, a machine learning-based ensemble method, WAVED, is designed and evaluated to demonstrate the effectiveness of the MSALSTM-CNN method to more accurately detect anomalies.

III. Network Model, Dataset, and Preliminaries

A vehicular sensor network, namely, $SN = \{S_1, S_2, \ldots S_n\}$, contains n number of sensors to gather sensor readings to connect and control the components of the vehicle. Each S_i in SN provides a tuple of readings, $R =< R_1, R_2, \ldots R_n >$ in the form of $M \times N$, where M represents the number of rows and N represents the number of columns. Let T be the set of target classes (i.e., normal, instant, constant, gradual drift and bias anomalies) to be identified using predictive model, i.e., CNN in this case, against each test instance. The model returns a label from the target class T_i reporting the current test instance as normal or anomalous. Hence the feature matrix can be represented using (1).

$$F = \{f_{m_i}^{n_i}\}_{n=1}^{N} \tag{1}$$

where F represents a feature matrix, m_i denotes instances of the feature n_i. Since the range of values of raw sensor data varies, it is important to scale and normalize dataset, wherever applicable. The scaled feature matrix is described using (2)

$$F' = \frac{f_{m_i}^{n_i} - \min(f_{m_i}^{n_i})}{\max(f_{m_i}^{n_i}) - \min(f_{m_i}^{n_i})} \tag{2}$$

where F' is the normalized feature matrix, $f_{m_i}^{n_i}$ denotes the numerical value of a feature that is subtracted from the minimum value of a feature $f_{m_i}^{n_i}$. In the denominator, the maximum value of the feature $f_{m_i}^{n_i}$ is subtracted from the minimum value of the feature $f_{m_i}^{n_i}$. The scaled data is used in the WAVED method. The data set used in this study was initially explored by Wyk *et al.* for anomaly detection [1]. The dataset consists of three features, namely, (i) *speed*: calculated by vehicle speedometer, denoted by sensor 1; (ii) *GPS speed*: denoted

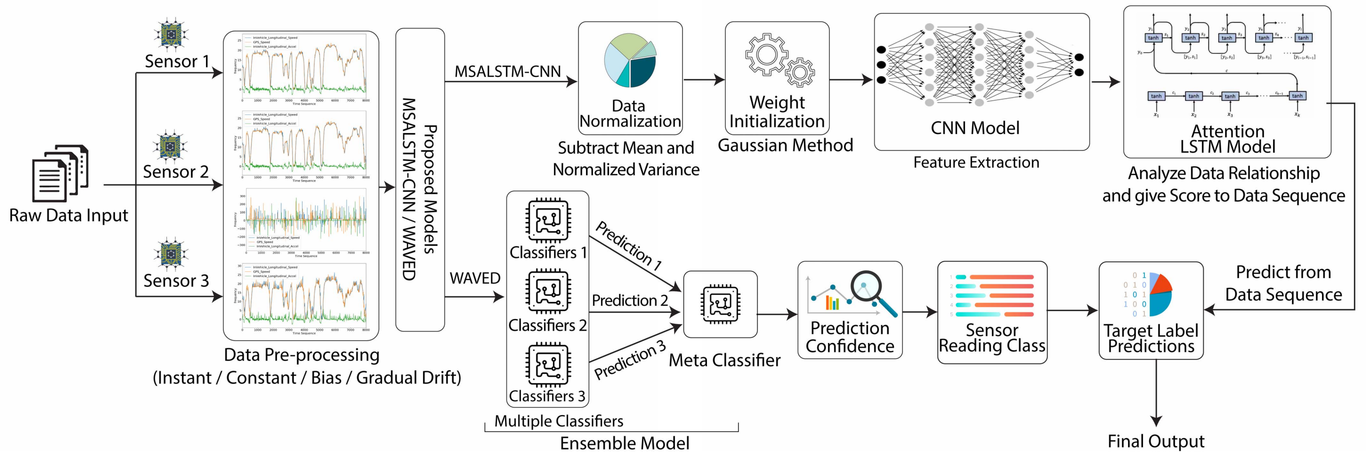


Fig. 1. An overview of the proposed framework.

by sensor 2 and (iii) *acceleration*: calculated on the basis of in-vehicle speed, denoted by sensor 3. Since the original Safety Pilot Model Deployment (SPMD) dataset [17] did not include anomalies, Wyk *et al.* [1] used simulations to add 4 types of anomalies, i.e., (i) instant, (ii) constant, (iii) gradual drift, and (iv) bias to the SPMD dataset. State-of-the-art studies mention that these anomalies can cause malfunctions in the vehicular services such as speed and acceleration due to faults and cyberattacks [18], [19]. We, therefore, focus on four key types of anomalous readings consistent with the existing research works [1], [20], [21].

Instant anomaly type, among these four anomaly types, represents a sudden and unknown change that is noticed between the readings of two successive CAV sensors. Constant anomaly types represents an uncorrelated behavior that connotes the irregularities in the values of sensors. The gradual drift anomaly represents a small change during a specified period. The bias anomaly represents a constant offset for a particular time in a sensor. The dataset contains anomalies generated at different frequencies (i.e., 1%, 5%) and added to the ground truth value of the normal readings. The dataset contains two types of data: (i) single anomaly type and (ii) mixed anomaly type. Single anomaly type represents specific types of anomalies (i.e., instant, constant, gradual drift, and bias), whereas the mixed anomaly type includes various types of anomalies. These categories of anomalies are from the list of malicious attacks that could cause malfunctions in CAVs and may lead to severe consequences [22], [23]. Key notations of this study are listed given in Table I.

IV. Methods

This section elucidates the concepts of CNN, LSTM, Random Forest, AdaBoost and Support Vector Machines (SVM), as they are the building blocks of our proposed methods. Fig. 1 depicts the functionality of our presented framework. The classifiers are selected based on their capability to cover different aspects such as small, large, and noisy datasets, and detection rate improvement of weak learning classifiers.

A. Multistage Attention With LSTM-Based CNN

The CNN takes input data instances, processes them through numerous hidden layers, and then classifies them [24].

TABLE I

List of Key Notations

Notation	Description
CNN	Convolution Neural Network
CV	Context Vector
IC	Class of Each Instance
ICC	Instance Class Count
KF	Kalman-Filter
LSTM	Long short-Term Memory
NTL	Total Target Classes
PC	Predicted Confidence
SPMD	Safety Pilot Model Deployment
SV	Sequence Vector
SVM	Support Vector Machine
TC	Targeted Confidence
TL	Target Lable
WAVED	Weight-Adjusted Fine-Tuned Ensemble

The requirement of pre-processing in CNN is relatively lower in comparison to other classification algorithms. The attention-based CNN has slightly different architecture as compared to the normal CNN architecture. Below we define the structure used in this study.

$$CV = \sum_{k=1}^{T} at_{jk} h_k \tag{3}$$

where

$$at_{jk} = softmax(fn(h_k, s_{j-1})) \tag{4}$$

$$fn(h_k, s_{j-1}) = tanh(W[h_k, s_{j-1}]) \tag{5}$$

The context vector, (3), denotes the significant data that is used to predict target labels [25]. The notation T denotes the context vector size, at_{jk} represents the attention weight score given to the sequence, and h_k represents the current state of the sequence. The attention score given to the vectors is shown in (4), where s_j denotes the previous hidden states and the attention alignment function is given in (5) [25].

The process begins by converting the target labels into one hot encoding using one-hot-transformer, where labels are converted into label vectors of 1's and 0's. Sensor data is next converted into scaled data using the standard scaler function. Next the data labels are transformed into a sequence. The data are reshaped into 3 dimensional sequences to be fed to the CNN model. The 3D-sequence data are passed to the CNN model, where CNN layers extract the significant information from the data and also tune the weights accordingly. Extracted features are converted into vectors from the obtained matrix, which are fed to the LSTM layer where aspect relation in sequence data points is checked and attention mechanism is mounted on the LSTM layer. Attention mechanism applies weight scoring in those vectors and then context vector is created by multiplying the vectors by the weight score. Then sum of those values is computed. The context vector is used to predict the target labels. The process is shown in (6) to (11).

$$\gamma f = \sigma(W_{fg}[P_{re} - s_{t-1}, h_{t-1}, h_t] + b_{fg}) \tag{6}$$

$$\gamma i = \sigma(W_{ig}[P_{re} - s_{t-1}, h_{t-1}, h_t] + b_{ig}) \tag{7}$$

$$\gamma o = \sigma(W_{og}[P_{re} - s_{t-1}, h_{t-1}, h_t] + b_{og}) \tag{8}$$

$$h_t = \gamma o + tanh(s_t) \tag{9}$$

$$\widetilde{S_t} = tanh(W_{st}[h_{t-1}, h_t] + b_{st}) \tag{10}$$

$$St = \gamma f \times s_{t-1} + \gamma i \times \widetilde{S_t} \tag{11}$$

The notation W, in (6) to (11), represents the weights of the neural network, h_t denotes the current input, h_{t-1} is the information passed from one cell to another, and b is a bias. The forget gate layer is given in (6), which specifies the sigmoid function, (σ), that decides the importance of the information. The sigmoid and tanh layers, in (7) and (10), represent as input gate that is used to decide which values are going to be updated and to create a new candidate vector which is added to the $\widetilde{S_t}$, respectively.

The proposed anomaly detection method consists of 5 hidden layers, three CNN Layers contain 64 neurons in the 1st layer, 32 in the second and third layers. 75% of data instances are used for training, 10% for validation, and 15% for test purposes. These parameters enable a model to achieve highest detection rates. The performance of the proposed method can be affected by various hyper-parameters involved in the network. We tuned those parameters to attain the best performance. To avoid over-fitting, we used L2-regularization weight decay method. The 20% dropout probability was used in the network layer. The non-linear activation function Rectified Linear Unit (Relu) was employed in the network. A Batch size of 200 with 200 epochs was used. For optimization, we used Adam, a method for stochastic optimization, with a decay rate of 0.025. To select the best weights, we introduced a checkpoint to record all the weights in a file whenever the loss was detected. In the last output layer, we used a sigmoid function because of the binary output. The purpose of the attention layer was to create a context vector for input [26].

1) MSALSTM-CNN Algorithm: Let D denote the dataset containing instance $I = \{i_1, i_2, \ldots, i_n\}$ and H represents the one-hot encoding transformer function to convert the labels into vectors, V. To normalize data, the first step is to subtract the mean, μ, from data and then normalize the variance σ. The next step is conversion of data to a 2D matrix using the NumPy function. The third step is the weight initialization, using the Gaussian variable, where the number of layers is represented as L, the number of features as n and weight matrix is represented as W. The dimension of the Weight matrix is represented as $x * y$. In the next step, the 2D matrix, D_2, contains the training dataset that is converted into a 3D matrix, D_3, using the reshape function and is then fed into the CNN. Furthermore, the CNN model produces the feature map F, which is converted further into vectors V after max-pooling. It is given as input to the LSTM model, where aspect and context ac between data sequences are analyzed. This information is used by attention mechanism to assign scores, WS, to each sequence vector, SV. Those vectors are then multiplied with their scores and summed up in the form of the context vector, CV, to predict the target label, TL.

The target labels are converted into one-hot encoding. Next, the dataset is scaled with the standard scaling method. Data points are converted into data sequences and passed to MSALSTM-CNN in 3 dimensions. The features are extracted and than the feature vectors are fed into the LSTM memory units after max-pooling. The sequence patterns of anomalies are then learned and with the help of attention mechanism, scores are assigned to the data based on their significance. This process improves the learning procedures and helps to achieve the higher accuracies in the anomaly detection process.

Algorithm 1 Multi-Stage Attention Mechanism With a Convolutional Neural Network (CNN)

```
Input: data ← CAV Sensor Readings
Output: Normal, Anomalous
$V \leftarrow H(data)$                                # One Hot Encoding
$\mu \leftarrow 1/m * \sum_{i=1} X^{(i)}$             # Data scaled
$X \leftarrow X - \mu$
$\sigma^2 \leftarrow 1/m * \sum_{i=1} X^{(i)2}$
$X/ = \sigma^2$
$D_2 \leftarrow np.array(Df)$                         # Matrix Convergence
for $l$ in $range(1, len(L))$ do                      # Weight Initialization
    $W[l] \leftarrow rand((x \times y)) * \sqrt(2/n[l-1])$
end for
$D_3 \leftarrow$ ReshapeMatrix($D_2$)                  # CNN Model
$F \leftarrow$ STATE($D_3$)
$V \leftarrow$ MaxPooling($F$)                         # Vector Conversion
$ac \leftarrow$ LSTM($V$)                              # Attention LSTM
$ws \leftarrow$ AttentionMechanism($ac$)
$cv \leftarrow \sum_{(ws \times sv)}$
$TL \leftarrow$ PredictClass($cv$)
for $i$ in $range(1, len(TL)$ do
    if $(TL[i] = y\_test[i])$ then
        return TL[i]
    else
        return y_test[i]
    end if
end for
for each epoch in range (10): do
    Evaluate Loss, Validation Loss
    Evaluate Accuracy and Validation Accuracy
    Evaluate Precision, Recall, F-Score, Roc Curve and Confusion Matrix
end for
return Output
```

B. Ensemble Learning

The idea behind ensemble methods is to combine various classifiers and use the majority voting strategy (e.g., average predicted probability) to estimate the target labels. A voting classifier is one of the ensemble methods. These methods are helpful in balancing the individual weaknesses of well-performing classifiers [27]. To make the intrusion anomaly detection process more efficient, our ensemble method combines the predicted results of the multiple classifiers and outputs the results using a majority voting strategy. Each classifier is tuned to deliver the best results. The votes are collected using (12)

$$\widetilde{y} = argmax(N_c(y_t^1), N_c(y_t^2), \ldots, N_c(y_t^n)) \tag{12}$$

where $N_c(y_t)$ represents the class that receives the highest number of votes. We evaluated several models such as Random Forest (RF), AdaBoost, and Support Vector Machine (SVM).

Random Forest: achieves a reduced variance by aggregating the diverse decision trees, sometimes at the cost of slightly higher bias [28]. In practice, however, the reduction of variance is often significant and results in a better classifier model. The model provides functionality to work with complex data unlike traditional classifiers. The RF generates various decision trees and each decision tree votes for the best target label. The prediction is done by majority voting. The parameters used in our ensemble method are the following: bootstrap = true, criterion = entropy, max-depth = 100, max-leaf-nodes = 2, max-samples = 1000, min-samples-leaf = 1, min-samples-split = 2, n-estimators = 100, n-jobs = 10, and random-state = 2.

AdaBoost: is a classier that combines weak-learning classifiers to enhance the performance [29]. AdaBoost is adaptive in a sense that successive weak learners are pushed in favor of those occurrences that are miss-classified by weak learners. AdaBoost is good for noisy data and outliers. Adaboost parameters used in the ensemble weak learners model are administered by the varying n-estimators. Learning rate decides the participation of the weak-learners, but by default, these weak-learners are decision stumps. A number of weak learners can be specified through a base estimator. Adaboost n-estimators parameter is the number of weak learners that are to be trained iteratively. Adaboost parameters used in our ensemble method are algorithm = SAMME.R, base-estimator = SVC & RF learning-rate = 1.0, n-estimators = 100, random-state = 5)

Support Vector Machine: is a supervised machine learning classification and regression method [30]. It uses kernels to transform data to figure out the optimal boundary between the samples. It maps the data points in a higher dimension space to separate required categories which are divided by a boundary-line or hyper-plane. It works rather well in non-linearly separable problems by the help of kernels. The best parameters used in our ensemble method are: regularization parameter (C) = 1.0, kernel = rbf, polynomial-degree = 3, Kernel-coefficient (gamma) = scale, coeff = 0.0, (shorten training time) shrinking = True, probability-estimates = true, stopping criterion (tol) = 0.001, size of the kernel-cache = 200, verbose = false, decision-function-shape = ovr. When solving hard problems, max-iter = −1 is used to not have the maximum number of iterations by the optimizer.

The test instance is passed through each model which makes prediction. The final decision is based on the number of predictions of a class. If the percentage of a class predictions is above 50%, then it would be the final prediction for that test instance. The ensemble method limitation may occur when none of the prediction votes is above 50%. Then the method is considered not to predict stably. The method can still give prediction value based on the number of votes even though the highest prediction is below 50%.

1) WAVED Algorithm: Let D represent the dataset containing instances $I = \{i_1, i_2, \ldots, i_n\}$. The predicted confidence of each classifier is denoted by PC and TC represent the targeted confidence that is set as a threshold to evaluate the PC of each classifier. Let TL represent the target class labels to be predicted by each classifier and NTL denote the total target classes. The notation IC represents the class of each instance, whereas ICC represents the instance class count that is incremented when a classifier votes for the class label. Each instance in I is given as input to the classification model for prediction as anomaly or normal and appended in IC. The confidence of ICC and TL is then evaluated. Each classifier provides a vote against each observation. The ground truth value of 80% is set to compare the confidences. The data instances need to achieve ground truth score of 80% or more to fall in a particular class. If the requirement is not fulfilled, then another instance is added until the threshold is reached. In the case, more than one class is participating in the classification result, which is the case when 2 or more classes have same number of votes, then any one of those can be randomly selected. If CL is greater than the defined threshold, the target class is then considered as a label of that instance.

V. RESULTS

We first evaluate the performance of our designed, i.e., MSALSTM-CNN and WAVED methods on different anomaly types. We then compare the results with a state-of-the-art method [1]. The performance evaluation metrics include (i) accuracy, (ii) sensitivity, (iii) precision, and (iv) F1 score [1]. These standard performance metrics are primarily chosen to testify the capability of the models in accurately distinguishing between normal and anomalous instances.

A. Single Anomaly Types

The detection capability of the MSALSTM-CNN and WAVED methods is examined for the individual types of anomalies, namely, (i) instant, (ii) constant, (iii) gradual drift, and (iv) bias. For this set of experiments, we followed the experiment pattern of Wyk *et al.* [1].

1) Instant: Table II shows the capabilities of the MSALSTM-CNN and WAVED methods in detecting instant anomalies. The performance of the MSALSTM-CNN and WAVED methods consistently increases with the increase in the magnitude of anomalous instances. For lower magnitudes of anomalous sensor values, the methods show moderate

TABLE II

INSTANT ANOMALY DETECTION THROUGH MSALSTM-CNN AND WAVED METHODS

	WAVED(%)				MSALSTM-CNN(%)			
Anomaly Magnitude	Acc	Sens	Prec	F1	Acc	Sens	Prec	F1
base value + 25 x N (0,0.01)	81.01	53.42	96.81	68.84	84.10	54.63	98.12	**70.18**
base value + 100 x N (0,0.01)	93.87	86.54	98.03	91.92	95.8	89.60	98.43	**93.80**
base value + 500 x N (0,0.01)	96.82	99.59	98.18	98.26	96.02	99.79	97.86	**99.02**
base value + 1000 x N (0,0.01)	98.81	96.99	99.87	98.41	98.98	98.16	99.21	**98.68**
base value + 10000 x N (0,0.01)	99.7	99.41	99.61	**99.52**	99.43	98.93	99.75	99.34

TABLE III

CONSTANT ANOMALY DETECTION THROUGH MSALSTM-CNN AND WAVED METHODS

		WAVED(%)				MSALSTM-CNN(%)			
Anomaly Magnitude	Duration, d	Acc	Sens	Prec	F1	Acc	Sens	Prec	F1
base value + U (0,5)	3	94.38	89.56	98.87	93.99	95.05	90.25	99.51	**94.65**
base value + U (0,5)	5	95.07	91.63	99.24	95.28	95.40	92.30	98.97	**95.51**
base value + U (0,5)	10	95.56	94.12	98.93	96.46	96.61	95.61	99.28	**97.41**
base value + U (0,3)	10	95.32	93.78	99.04	96.33	96.44	95.44	98.93	**97.15**
base value + U (0,1)	10	90.84	86.81	98.65	92.35	93.02	90.76	98.69	**94.55**

Algorithm 2 Weight-Adjusted Ensemble of Distinct Classifiers

```
Input: $Reading \leftarrow CAVSensorReadings$
Output: Normal, Anomalous
    Evaluation Measures: Accuracy, F-Score, Recall, Precision
$i \leftarrow [Reading]$                          # Current Instance
$TI \leftarrow []$                                # Total Instances
$PC \leftarrow []$                                # Predicted Confidence
$TC \leftarrow 80$                                # Targeted Confidence
$TL \leftarrow [Normal, Anomalous]$               # target class labels
$CL \leftarrow NULL$                              # Confidence Level
$NTL \leftarrow len(TL)$                          # Total target class labels
$IC \leftarrow NULL$                              # Instance class
$ICC \leftarrow NULL$                             # Instance class Count
for $each\ i\ in\ I$ do
    $TI \leftarrow TI++$
    $IC \leftarrow$ getClassification($i$)
    $ICC[TL] \leftarrow IC++$
    $(CL, NTL) \leftarrow$ getHighestConfidenceLevel($ICC, TL$)
    if $(CL \geq TC)$ then
        $ICC \leftarrow TL$
    end if
end for
return $max(ICC)$
```

performances, as indicated in rows 1 and 2 of Table II. Nevertheless, for smaller values, the variance among the abnormal and normal sensor reading values are usually considerably less to show any considerable risks in the functionality of the vehicle. For those magnitudes that can impact significantly in terms of danger to the functionality of a vehicle, i.e., rows 3 to 5, the methods can identify abnormal responses with high performances. As it is obvious from Table II, the methods show nearly similar performances for instant anomalies with high magnitudes. The maximum F1-Score is observed in row 5, i.e, 99.52% by the WAVED method when the magnitude is maximum while also achieving the maximum accuracy of 99.7%. The minimum F1 score is observed by the WAVED method when anomaly magnitude is the least in row 1. In conclusion, it can be observed that for low anomaly magnitudes, the MSALSTM-CNN method performs better than the WAVED method. This trend continues in the case of high magnitude, except when the anomaly magnitude is as high as base value $+ 10000 \times N(0, 0.01)$, where the WAVED method shows better performance. However, the performance gain in this case by the WAVED method is just 0.18%.

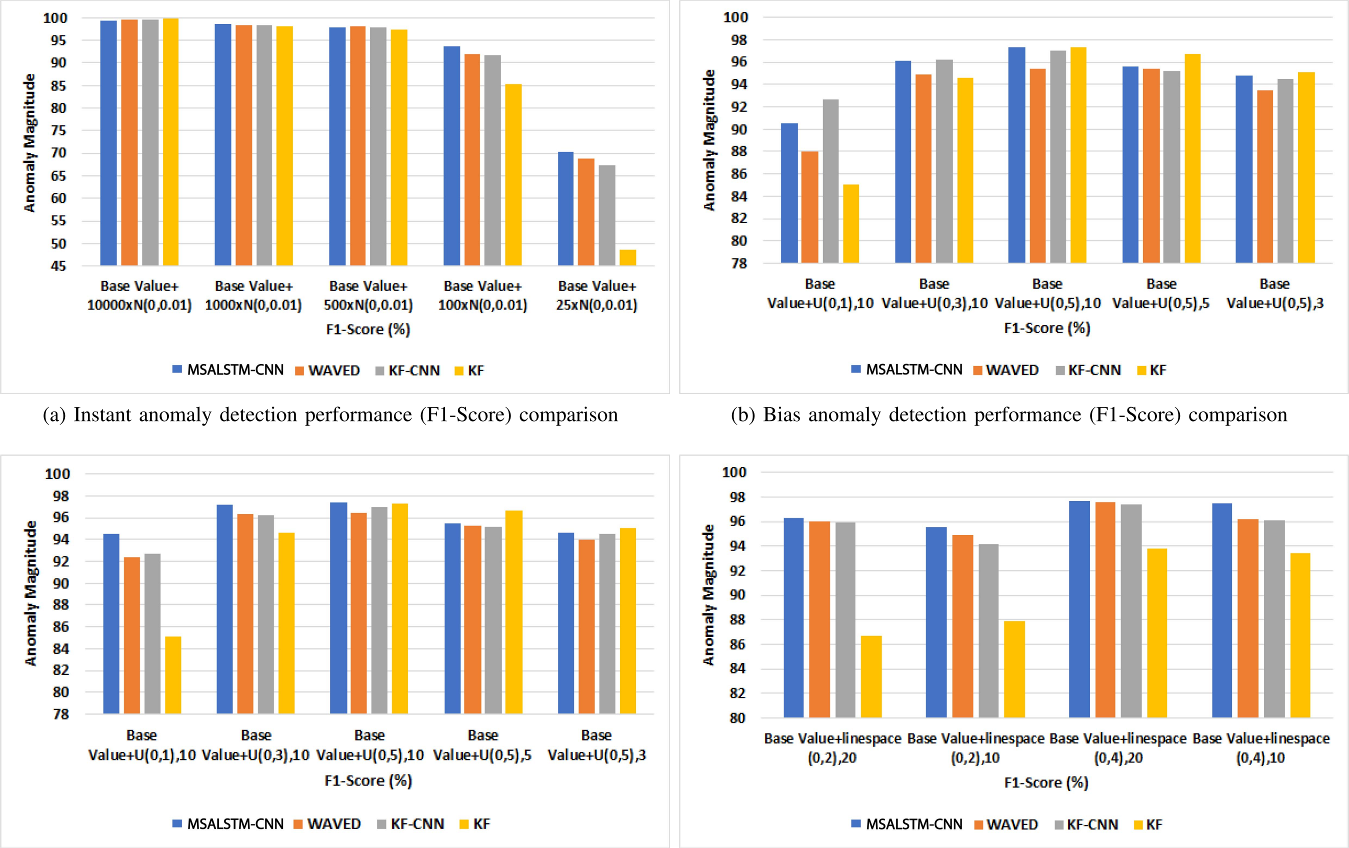
2) Constant: Table III shows the constant anomaly category results. Referring to rows 1 to 3 of Table III, the performance of both MSALSTM-CNN and WAVED methods steadily increases, provided the anomaly magnitudes are projected from the original random variable. In this case, the MSALSTM-CNN method outperforms the WAVED method. Furthermore, as given in rows 3 to 5, having constant anomaly duration, i.e., $d = 10$, the performance of both WAVED and MSALSTM-CNN methods is more proficient in the cases of high magnitude of anomalous instances. The MSALSTM-CNN method, however, has significant gains in F1-score throughout this set of experiments. Furthermore, the MSALSTM-CNN method outperforms the WAVED method in terms of accuracy, sensitivity, and F1-score across all experiments. The WAVED method shows gradual variation in the performance, but the MSALSTM-CNN method shows consistency. However, both methods show descending performance in the last row. The advanced neural network attention method still gets the best feature sequence, which helps to generate more accurate results. The WAVED method tends to perform well as the number of instances increases, but the MSALSTM-CNN method performs well in most of the cases. The maximum F1-Score is observed in row 3, i.e, 97.41% by the MSALSTM-CNN method. The minimum F1-score is observed in row 5 by the WAVED method. This shows the better performance of the MSALSTM-CNN method over the WAVED method is achieved due to increase in duration. To better synthesise the data sequence in the process, the LSTM comes into action to better learn and adapt

TABLE IV
GRADUAL DRIFT ANOMALY DETECTION THROUGH MSALSTM-CNN AND WAVED METHODS

		WAVED(%)				MSALSTM-CNN(%)			
Anomaly Magnitude	Duration, d	Acc	Sens	Prec	F1	Acc	Sens	Prec	F1
base value + $linespace$ (0,4)	10	95.08	93.41	99.16	96.19	96.01	95.85	99.14	**97.46**
base value + $linespace$ (0,4)	20	96.08	95.39	99.31	97.59	96.21	96.03	99.27	**97.62**
base value + $linespace$ (0,2)	10	93.58	91.02	99.51	94.90	94.36	93.09	99.07	**95.58**
base value + $linespace$ (0,2)	20	93.50	93.68	98.48	96.02	94.09	92.78	99.52	**96.03**

TABLE V
BIAS ANOMALY DETECTION THROUGH MSALSTM-CNN AND WAVED METHODS

		WAVED(%)				MSALSTM-CNN(%)			
Anomaly Magnitude	Duration, d	Acc	Sens	Prec	F1	Acc	Sens	Prec	F1
base value + U (0,5)	3	94.12	88.05	99.73	93.52	95.10	90.53	99.47	**94.78**
base value + U (0,5)	5	95.33	92.34	98.87	95.44	95.51	91.99	99.55	**95.62**
base value + U (0,5)	10	94.87	92.34	98.87	95.44	96.56	95.74	99.06	**97.37**
base value + U (0,3)	10	93.57	90.76	99.42	94.89	94.99	93.82	98.50	**96.10**
base value + U (0,1)	10	86.20	79.51	98.62	88.03	88.55	85.81	95.89	**90.57**



(a) Instant anomaly detection performance (F1-Score) comparison

(b) Bias anomaly detection performance (F1-Score) comparison

(c) Constant anomaly detection performance (F1-Score) comparison

(d) Gradual drift anomaly detection performance (F1-Score) comparison

Fig. 2. The performance comparison with the baseline method [1].

the pattern to make the MSALSTM-CNN method better in identifying constant anomalies more effectively.

3) Gradual Drift: Table IV highlights the results of the gradual drift anomaly detection by the MSALSTM-CNN and WAVED models. The MSALSTM-CNN and WAVED methods perform reasonably well. The maximum F1-Score is observed in the row 2, i.e, 97.62% by the MSALSTM-CNN method when the duration is 20 epochs and the anomaly magnitude configuration is $(basevalue + linespace(0, 4))$. The minimum F1-score is observed when the anomaly magnitude is least in row 3, i.e, 94.90% by the WAVED method. Moreover, the MSALSTM-CNN method maintained consistency in the performance, with minor variation, throughout the experiments, but the WAVED method shown more variations. The superiority of the MSALSTM-CNN over WAVED method is mainly because the gradual drift is challenging data to work with, but by employing the advance attention method with

CNN to figure out the best feature enables LSTM to handle complex patterns.

4) Bias: Table V presents the results of the bias anomaly detection by the MSALSTM-CNN and WAVED methods. The results in rows 1 to 3 show the ascending performance of the MSALSTM-CNN and WAVED methods to detect anomalies. Similarly, in rows 3 to 5, the performance of both the MSALSTM-CNN and WAVED methods descend when the magnitude of the abnormal values randomly decreases. The maximum F1-Score is observed in row 3, i.e, 97.37% by the MSALSTM-CNN method when the magnitude is maximum, while also having the maximum accuracy of 96.56%. The minimum F1-score is observed when the anomaly magnitude is least in row 1 by the WAVED method. However, a side by side comparison of the MSALSTM-CNN and WAVED methods shows that the MSALSTM-CNN method performs better than the WAVED method, due to its capabilities such as advance neural network performs better with large data, long duration, and with complex sequences of data. Our designed method is generic in nature such that with the assistance of attention mechanism, it handles the short duration complex data and figures out the significant patterns to identify the anomalies.

B. Mixed Anomaly Types

In this section, we examine the performance of the MSALSTM-CNN method in the case of mixed anomaly types. As shown in Table VI, the MSALSTM-CNN method performs well in detecting mixed anomaly types. We test our method with instant, constant, gradual drift, and bias anomalies with different parametric configurations and sensors. Each category has 3 sensors whose anomaly values are tested. In the case of the instant anomaly type, the model shows the highest F1-score of 78.11% for sensor 1. Similarly, in the case of the constant anomalies, MSALSTM-CNN shows the highest F1-score of 90.43% for sensor 1. In the case of the gradual drift anomalies, MSALSTM-CNN shows the highest F1-score of 84.30%. In the case of the bias anomalies, MSALSTM-CNN shows the highest F1-score of 90.45% for sensor 1. The F1-score shows varying behavior in different anomaly types and sensors. The highest observed F1-score is 90.45% for detecting the bias anomalies. The overall variance in values is mainly because of different configurations, but the values illustrate the promising performance.

TABLE VI

PERFORMANCE OF THE MSALSTM-CNN METHOD IN THE CASE OF MIXED ANOMALY TYPES

Anomaly Type	Sensor	Acc	Sens	Prec	F1
Instant, 1000 x N(0,0.01)	1	91.34	64.51	98.91	78.11
	2	88.8	60.17	94.88	73.38
	3	89.92	48.03	97.89	64.44
Constant, U(0,5), d = 20	1	95.64	88.13	92.87	90.43
	2	91.35	71.52	93.66	81.10
	3	91.30	67.35	92.08	77.79
GD $linespace$(0,4),d=20	1	93.61	77.38	92.59	84.30
	2	91.89	72.16	94.56	81.35
	3	89.94	68.04	86.93	76.33
Bias, U(0, 5), d = 10	1	96.02	88.93	92.04	90.45
	2	93.10	72.74	97.83	81.43
	3	90.40	63.09	96.11	76.17

TABLE VII

PERFORMANCE COMPARISON OF THE TRAINING METHODS WITH THE BASELINE APPROACH [1] FOR MIXED ANOMALY TYPES

		MSALSTM-CNN		Baseline [1]	
Anomaly Type	Sensor	Acc	F1	Acc	F1
Instant, 1000 x N(0,0.01)	1	91.34	**78.11**	91.1	77.9
	2	88.8	**73.38**	88.1	72.8
	3	89.92	**64.44**	85.4	61.2
Constant, U(0,5), d = 10	1	95.64	**90.43**	95.3	90.1
	2	91.35	**81.10**	90.9	80.7
	3	91.30	**77.79**	89.2	76.0
GD $linespace$(0,4),d=20	1	93.61	**84.30**	92.5	83.3
	2	91.89	**81.35**	90.6	80.2
	3	89.94	**76.33**	88.2	74.7
Bias, U(0, 5), d = 10	1	96.02	**90.45**	94.8	89.3
	2	93.10	81.43	91.7	**82.1**
	3	90.40	**76.17**	89.3	75.1

C. Comparative Overview

As shown in Table VII, our method shows promising results for mixed anomaly types. In all cases, i.e., instant, constant, gradual drift, and bias, our method outperforms the baseline approach, except for sensor 2 in the case of Bias anomaly type. Table VIII shows a detailed comparison of the MSALSTM-CNN and WAVED methods with the baseline approaches, i.e, KF and CNN-KF. It can be observed from Table VIII, that the overall performance boost is achieved in terms of accuracy and F1-score by the MSALSTM-CNN method with a maximum F1-score of 98.68% in detecting instant anomalies, 97.41% in detecting constant anomalies, 97.62% in gradual drift anomalies, and 97.37% in detecting bias anomalies. Fig. 3 shows the graphical depiction of the detailed comparison between the MSALSTM-CNN and the baseline competitor: KF-CNN. The comparison is based on F1 - Score. It can be seen from the graph the MSALSTM-CNN method shows a step ahead performance than the baseline approach in the case of mixed anomalies. Senor 1, Sensor 2, and Sensor 3 represent the In-Vehicle Longitudinal speed sensor, in-Vehicle Longitudinal acceleration sensor, and GPS speed sensor.

Fig. 3 shows the graphical depiction of the detailed comparison between the MSALSTM-CNN and the baseline competitor: KF-CNN [1]. This comparison is based on F1 - Score. It can be seen from the graph the MSALSTM-CNN method shows a step ahead of performance than the baseline approach in the case of mixed anomalies.

Table VIII shows a detailed comparison of the MSALSTM-CNN and WAVED methods for single anomaly types with the baseline approaches, i.e, kalman filter (KF) and convolutional neural network-based Kalman filter (CNN-KF) [1]. It can be observed from Table VIII, that the overall performance boost is achieved in terms of accuracy and F1-score by the MSALSTM-CNN method with the highest F1-score of 99.02% in detecting instant anomalies, 97.41% in detecting constant anomalies, 97.62% in gradual drift anomalies, and 97.37% in detecting bias anomalies. Fig. 2a, 2b, 2c, and 2c show comparison of the MSALSTM-CNN and WAVED methods with the

TABLE VIII
COMPARISON OF MSALSTM-CNN AND WAVED WITH BASELINE APPROACH [1] FOR SINGLE ANOMALY TYPE

			KF [1]		CNN-KF [1]		WAVED		MSALSTM-CNN	
Anomaly Type	Anomaly Magnitude	Duration, d	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Instant Anomaly	base value + 25 x N (0,0.01)	-	95.7	48.5	80.0	67.4	81.01	68.84	84.10	**70.18**
	base value + 100 x N (0,0.01)	-	98.6	85.3	93.6	91.7	93.87	91.92	95.8	**93.80**
	base value + 500 x N (0,0.01)	-	99.7	97.3	98.3	96.82	98.26	98.18	96.02	**99.02**
	base value + 1000x N (0,0.01)	-	99.8	98.1	98.8	98.4	98.81	98.41	98.98	**98.68**
	base value + 10000 x N (0,0.01)	-	99.9	**99.8**	99.7	99.5	99.7	99.52	99.43	99.34
Constant Anomaly	base value + U (0,5)	3	98.5	95.0	94.9	94.5	94.38	93.99	95.05	**94.65**
	base value + U (0,5)	5	**98.5**	96.7	95.1	95.2	95.07	95.28	95.40	95.51
	base value + U (0,5)	10	97.8	97.3	96.2	97.0	95.56	96.46	96.61	**97.41**
	base value + U (0,3)	10	95.7	94.6	95.3	96.2	95.32	96.33	96.44	**97.15**
	base value + U (0,1)	10	88.8	85.1	91.2	92.7	90.84	92.35	93.02	**94.55**
Gradual Drift Anomaly	base value + $linespace(0, 4)$	10	94.7	93.4	94.7	96.1	95.08	96.19	96.01	**97.46**
	base value + $linespace(0, 4)$	20	92.2	93.8	96.0	97.4	96.08	97.59	96.21	**97.62**
	base value + $linespace(0, 2)$	10	90.3	87.9	93.0	94.2	93.58	94.90	94.36	**95.58**
	base value + $linespace(0, 2)$	20	83.1	86.7	93.7	95.9	93.50	96.02	94.09	**96.03**
Bias Anomaly	base value + U (0,5)	3	98.5	**95.7**	94.6	94.2	94.12	93.52	95.10	94.78
	base value + U (0,5)	5	98.5	**94.8**	94.9	95.2	95.33	95.44	95.51	95.62
	base value + U (0,5)	10	97.3	96.6	95.9	96.7	94.87	95.44	96.56	**97.37**
	base value + U (0,3)	10	95.9	94.8	94.4	95.5	93.57	94.89	94.99	**96.10**
	base value + U (0,1)	10	90.1	86.9	88.0	90.0	86.20	88.03	88.55	**90.57**

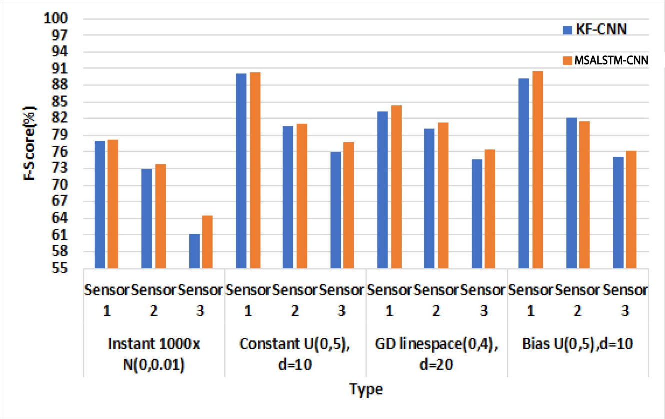


Fig. 3. The performance comparison of the training methods with the baseline approach [1] for mixed anomaly types.

baseline approaches, namely, KF-CNN and KF. In the case of instant anomalies, Fig. 2a shows the highest F1-score of 99.02%, dominating all approaches except when the anomaly magnitude is "base value + 10000 x N (0,0.01)" by MSALSTM-CNN. Similarly, in the case of constant anomalies, Fig. 2c shows the highest F1-score of 97.41% when the anomaly magnitude is "base value + U (0,5) and duration 3" by MSALSTM-CNN. In the case of gradual drift anomalies, Fig. 2d shows the highest F1-score of 97.62% by MSALSTM-CNN. In the case of bias anomalies, Fig. 2c shows the highest F1-score of 97.37% by MSALSTM-CNN. It is obvious from the graphs that the MSALSTM-CNN method performs better in detecting anomalies of lower magnitude and outperforms other approaches with a considerable gain in performance.

VI. CONCLUSION AND FUTURE WORK

Connected and Automated Vehicles are pivotal elements of modern smart cities. The sensors, installed within the automated vehicles, in this paradigm generate big amount of data that are used for offering different related services. The sensor-generated data may possess anomalies due to faults, errors, or cyberattacks, which need to be accurately detected. In this article, we designed and evaluated a framework to detect anomalous sensor readings in automated vehicles based on the Multi-Stage Attention mechanism with a Long Short-Term Memory (LSTM)-based CNN. A fine-tuned ensemble method, namely, WAVED, with optimum parameters, is also designed and analyzed to show the effectiveness of the MSALSTM-CNN method. Results show that the proposed method effectively enhances the anomaly detection rate with low magnitude of anomalous instances in the dataset. In the future work, we intend to design and analyze probabilistic confidence methods to substantiate the outcome of our proposed framework.

REFERENCES

[1] 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.

[2] Y. Wang, N. Masoud, and A. Khojandi, "Anomaly detection in connected and automated vehicles using an augmented state formulation," 2020, *arXiv:2004.09496*. [Online]. Available: http://arxiv.org/abs/2004.09496

[3] United States Department of Transportation. *Connected Vehicles and Cyber Security*. Accessed: May 12, 2020. [Online]. Available: https://www.its.dot.gov/factsheets/

[4] Y. Wang, N. Masoud, and A. Khojandi, "Real-time sensor anomaly detection and recovery in connected automated vehicle sensors," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 5, 2020, doi: 10.1109/TITS.2020.2970295.

[5] T. Zhang, Y. Zou, X. Zhang, N. Guo, and W. Wang, "Data-driven based cruise control of connected and automated vehicles under cyber-physical system framework," *IEEE Transactions on Intelligent Transportation Systems*, early access, May 12, 2020, doi: 10.1109/TITS.2020.2991223.

[6] A. Irshad, M. Usman, S. A. Chaudhry, H. Naqvi, and M. Shafiq, "A provably secure and efficient authenticated key agreement scheme for energy Internet based vehicle-to-grid technology framework," *IEEE Trans. Ind. Appl.*, vol. 56, no. 4, pp. 4425–4435, Jul./Aug. 2020.

[7] P. M. Laso, D. Brosset, and J. Puentes, "Analysis of quality measurements to categorize anomalies in sensor systems," in *Proc. Comput. Conf.*, Jul. 2017, pp. 1330–1338.

[8] M. Usman, V. Muthukkumarasamy, and X.-W. Wu, "Mobile agent-based cross-layer anomaly detection in smart home sensor networks using fuzzy logic," *IEEE Trans. Consum. Electron.*, vol. 61, no. 2, pp. 197–205, May 2015.

[9] B. Du *et al.*, "Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 972–985, Mar. 2020.

[10] S. Lee, Y. Cho, and B.-C. Min, "Attack-aware multi-sensor integration algorithm for autonomous vehicle navigation systems," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 3739–3744.

[11] F. van Wyk, A. Khojandi, and N. Masoud, "A path towards understanding factors affecting crash severity in autonomous vehicles using current naturalistic driving data," in *Proc. SAI Intell. Syst. Conf.*, 2019, pp. 106–120.

[12] F. Farivar, M. S. Haghighi, A. Jolfaei, and M. Alazab, "Artificial intelligence for detection, estimation, and compensation of malicious attacks in nonlinear cyber-physical systems and industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2716–2725, Apr. 2020.

[13] M. H. Basiri, J. G. Thistle, J. W. Simpson-Porco, and S. Fischmeister, "Kalman filter based secure state estimation and individual attacked sensor detection in cyber-physical systems," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 3841–3848.

[14] R. Wang, K. Nie, T. Wang, Y. Yang, and B. Long, "Deep learning for anomaly detection," in *Proc. 13th Int. Conf. Web Search Data Mining*, Jan. 2020, p. 894, doi: 10.1145/3336191.3371876.

[15] M. Kordestani, A. Chaibakhsh, and M. Saif, "Sms—A security management system for steam turbines using a multisensor array," *IEEE Syst. J.*, vol. 14, no. 3, pp. 3813–3824, Sep. 2020.

[16] F. S. Mozaffari, H. Karimipour, and R. M. Parizi, "Learning based anomaly detection in critical cyber-physical systems," in *Security Cyber-Physical Systerm*. Cham, Switzerland: Springer, 2020, pp. 107–130.

[17] D. Bezzina and J. Sayer, *Safety Pilot Model Deployment: Test Conductor Team Report*. Washington, DC, USA: U.S. Dept. Transportation, 2014.

[18] R. Currie, *Developments in Car Hacking*. Bethesda, MD, USA: SANS Institute, 2015.

[19] M. Waniek, G. Raman, B. AlShebli, J. Chih-Hsien Peng, and T. Rahwan, "Traffic networks are vulnerable to disinformation attacks," 2020, *arXiv:2003.03723*. [Online]. Available: http://arxiv.org/abs/2003.03723

[20] A. Gaddam, T. Wilkin, M. Angelova, and J. Gaddam, "Detecting sensor faults, anomalies and outliers in the Internet of Things: A survey on the challenges and solutions," *Electronics*, vol. 9, no. 3, p. 511, Mar. 2020.

[21] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Veh. Commun.*, vol. 21, Jan. 2020, Art. no. 100198.

[22] S. Xiao, X. Ge, Q.-L. Han, and Y. Zhang, "Distributed resilient estimator design for positive systems under topological attacks," *IEEE Trans. Cybern.*, early access, Apr. 17, 2020, doi: 10.1109/TCYB.2020.2981646.

[23] G. De La Torre, P. Rad, and K.-K.-R. Choo, "Driverless vehicle security: Challenges and future research opportunities," *Future Gener. Comput. Syst.*, vol. 108, pp. 1092–1111, Jul. 2020.

[24] R. Li, Z. Pan, Y. Wang, and P. Wang, "A convolutional neural network with mapping layers for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3136–3147, May 2020.

[25] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 577–585.

[26] M. Li, G. Clinton, Y. Miao, and F. Gao, "Short text classification via knowledge powered attention with similarity matrix based CNN," 2020, *arXiv:2002.03350*. [Online]. Available: http://arxiv.org/abs/2002.03350

[27] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Frontiers Comput. Sci.*, vol. 5, pp. 1–18, May 2020.

[28] A. Paul, D. Prasad Mukherjee, P. Das, A. Gangopadhyay, A. Rao Chintha, and S. Kundu, "Improved random forest for classification," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 4012–4024, Aug. 2018.

[29] Z. Qi, F. Meng, Y. Tian, L. Niu, Y. Shi, and P. Zhang, "AdaBoost-LLP: A boosting method for learning with label proportions," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3548–3559, Aug. 2018.

[30] K. Huang, H. Jiang, and X. Zhang, "Field support vector machines," *IEEE Trans. Emerging Topics Comput. Intell.*, vol. 1, no. 6, pp. 454–463, Dec. 2017.

Abdul Rehman Javed received the master's degree in computer science from the National University of Computer and Emerging Sciences, Pakistan. He is currently a Lecturer with the Department of Cyber Security, Air University, Pakistan. His current research interests include, but are not limited to, mobile and ubiquitous computing, data analysis, knowledge discovery, data mining, natural language processing, and smart homes, and their applications in human activity analysis, human motion analysis, and e-health.

Muhammad Usman received the M.S. degree in computer science from PMAS Arid Agriculture University, Pakistan, and the Ph.D. degree from the School of ICT, Griffith University. He was a Post-Doctoral Research Fellow in cyber security and machine learning with the University of Surrey, U.K. He is currently working as a Senior Lecturer of Cyber Security with the University of South Wales, U.K. He has authored over 45 research articles. His current research interests include design and analysis of security, privacy, and trust techniques for complex cyber-physical domains, formal and statistical modeling, applied machine learning, and data analytics.

Saif Ur Rehman is currently pursuing the B.Sc. degree in computer science with the Faculty of Computing and AI, Air University, Islamabad, Pakistan. His current research interests include, but are not limited to, cybersecurity, artificial intelligence, computer vision, network security, the Internet of Things (IoT), smart city, and application development for smart living. He aims to contribute to interdisciplinary research of computer science and human-related disciplines.

Mohib Ullah Khan received the master's degree in computer science from the National University of Computer and Emerging Sciences, Pakistan. He is currently a Lecturer with the University of Wah, Pakistan. His current research interests include, but are not limited to, data analytics, computational intelligence, data mining, data science, natural language processing, human motion analysis, e-health, employing computer vision in emotion recognition applications, and to predict COVID-19 case occurrence in humans.

Mohammad Sayad Haghighi (Senior Member, IEEE) is currently the Head of the IT Department, School of Electrical and Computer Engineering, University of Tehran, Iran. Prior to joining the University of Tehran, he was an Assistant Professor with the Iran Telecom Research Center. Since 2009, he has been holding research positions at Australian universities. His research interests include wireless ad-hoc networks and cybersecurity. He has served as a PC Member of many conferences, such as IEEE WNS, IEEE SICK, IEEE HPCC, IEEE DASC, and IEEE LCN. He has won several national grants, including one from the Iran National Science Foundation.