IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 26 April 2024, accepted 11 May 2024, date of publication 14 May 2024, date of current version 28 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3400913

RESEARCH ARTICLE

Securing Electric Vehicle Performance: Machine Learning-Driven Fault Detection and Classification

MAHBUB UL ISLAM KHAN[1], MD. ILIUS HASAN PATHAN[1], MOHAMMAD MOMINUR RAHMAN[2], MD. MAIDUL ISLAM[3], MOHAMMED ARFAT RAIHAN CHOWDHURY[4], MD. SHAMIM ANOWER[5], MD. MASUD RANA[5], MD. SHAFIUL ALAM[6], MAHMUDUL HASAN[7], (Graduate Student Member, IEEE), MD. SHOHANUR ISLAM SOBUJ[1], MD. BABUL ISLAM[8], (Graduate Student Member, IEEE), VEERPRATAP MEENA[9], (Member, IEEE), AND FRANCESCO BENEDETTO[10], (Senior Member, IEEE)

[1]Department of Electrical and Electronic Engineering, Hajee Mohammad Danesh Science and Technology University (HSTU), Dinajpur 5200, Bangladesh
[2]Information and Computing Technology Division, College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar
[3]Department of Electrical and Electronic Engineering, Green University of Bangladesh, Dhaka 1207, Bangladesh
[4]Department of Logistics and Supply Chain Management, Division of Engineering and Decision Science, College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar
[5]Department of Electrical and Electronic Engineering, Rajshahi University of Engineering and Technology (RUET), Rajshahi 6204, Bangladesh
[6]Applied Research Center for Environment and Marine Studies, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[7]Department of Computer Science and Engineering, Hajee Mohammad Danesh Science and Technology University (HSTU), Dinajpur 5200, Bangladesh
[8]Department of DIMES, University of Calabria, 87036 Cosenza, Italy
[9]Department of Electrical and Electronics Engineering, Amrita School of Engineering Bengaluru, Amrita Vishwa Vidyapeetham, Bengaluru 560035, India
[10]SP4TE—Signal Processing for Telecommunications and Economics Laboratory, Economics Department, Roma Tre University, 00145 Rome, Italy

Corresponding authors: Francesco Benedetto (francesco.benedetto@uniroma3.it) and Veerpratap Meena (veerpratapmeena@ieee.org)

ABSTRACT Electric vehicles (EVs) are commonly recognized as environmentally friendly modes of transportation. They function by converting electrical energy into mechanical energy using different types of motors, which aligns with the sustainable principles embraced by smart cities. The motors of EVs store and consume electrical power from renewable energy (RE) sources through interfacing connections using power electronics technology to provide mechanical power through rotation. The reliable operation of an EV mainly relies on the condition of interfacing connections in the EV, particularly the connection between the 3-ϕ inverter output and the brushless DC (BLDC) motor. In this paper, machine learning (ML) tools are deployed for detecting and classifying the faults in the connecting lines from 3-ϕ inverter output to the BLDC motor during operational mode in the EV platform, considering double-line and three-phase faults. Several machine learning-based fault identification and classification tools, namely the Decision Tree, Logistic Regression, Stochastic Gradient Descent, AdaBoost, XGBoost, K-Nearest Neighbour, and Voting Classifier, were tuned for identifying and categorizing faults to ensure robustness and reliability. The ML classifications were developed based on the datasets of healthy and faulty conditions considering the combination of six critical parameters that have significance in reliable EV operation, namely the current supplied to the BLDC motor from the inverter, the modulated DC voltage, output speed, and measured speed, as well as the output of the Hall-effect sensor. In addition, the superiority of the proposed fault detection and classification approaches using ML tools was assessed by comparing the detection and classification efficiency through some statistical performance parameter comparisons among the classifiers.

INDEX TERMS Electric vehicles, renewable energy, brushless DC motor, three-phase inverter, fault detection, simulation, ML classifier.

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

I. INTRODUCTION

One way to address the environmental issues brought on by greenhouse gas emissions from the usage of fossil fuels in daily life is to electrify transportation systems. In this regard, the Electric vehicle (EV) is viewed as a potential technology to change the transportation system into one that is greener and more environmentally friendly shortly by utilizing renewable energy (RE)-based energy sources and reducing the usage of conventional energies [1]. In a report of 2019 and 2021 [2], [3], the researchers mentioned that about 23% of total energy is consumed by transportation sectors. In addition to emerging EVs in the transportation sectors to make a greener environment, the reduction of energy consumption is also focused on by introducing a regenerative braking system, by which a considerable percentage of energy consumption is reducible as per [4]. In addition, the faults generated in the system are also important to be detected in ordeer to obtain the targeted performance. Therefore, a great amount of research has been conducted in recent decades by researchers in perspective to safe, reliable, and efficient motor drive systems with electric vehicles [5], [6], [7], [8], [9]. Recently, artificial intelligence has not only been employed in improving the EV systems but also in enhancing the power systems to supply reliable power to customers [10], [11], [12], [13], [14].

Typically, EVs are equipped with an array of electrical and mechanical components, interconnected through a complex network. The electrical devices include the inverter and converter topologies, electric motors, and control units, whereas the gearbox and wheel are mainly considered mechanical parts. Therefore, faults in any position of the electrical or mechanical devices, along with their connections, will lead to unreliable operations of EVs. Different types of faults, such as bearing faults, rotor winding faults, armature winding faults, inverter or converter faults, faults in connecting lines including single phase, phase to phase, and three phase faults among the devices, and so on, can be occurred in EV as it is referred in [4], [15], and [16]. In case of faults in EV configuration, if proper action is not taken timely, it can result in a continuous spread to other areas or devices of the EV and ultimately lead to the collapse of the entire system [17]. Therefore, it is important to monitor the entire EV platform during driving mode in different operating conditions employing online-based data acquisition, analysis, fault detection, and classification to overcome possible severe threats and enhance the system's safety and reliability. Generally, fault detection, classification, and diagnosis of EVs are investigated following broadly three approaches - physical model-based, mapping-based, and data-driven [18]. However, the mapping-based model, where the data acquisition is managed from graphs, and the model-based model are not suitable for large physical systems, while the data-driven model requires only a large amount of the system's monitored data from different parameters as reported in [19], [20], and [21].

In data-driven fault detection and classification approaches, data is accumulated for the parameters including current, voltage, speed, temperature, pressure and some others to develop machine learning-based tools for monitoring the conditions in different operation moments in real-time EV driving mode [22]. In the case of fault detection in an individual machine, signal processing-based vibration analysis is conducted to detect the bearing fault in [23], whereas higher-order statistical fault analysis for vibration is presented to identify faults in [24]. Fast Fourier transform (FFT) based frequency spectrum analysis is implemented in the stator current of the induction motor (IM) for the machine's real-time health condition monitoring purpose [25], [26], [27], while artificial and clustering technique is hybridized with spectrum analysis for analyzing the current signal to detect faults occurred in rotor bar and bearing of 3-ϕ squirrel cage IM [28]. Bearing fault in stepper machine is identified employing frequency evaluation of current and measuring rotary angle of the corresponding current in [29]. However, these methods are classical approaches in detecting, classifying, and diagnosing faults that occur in electrical machines, and are mostly not suitable for online condition monitoring in case of several operating conditions. Besides the classical models, recently artificial intelligence-based fault detection and classification models are being implemented in different areas to overcome the physical modeling of large complex systems and fault detection time issues.

In references [30], [31], and [32], artificial neural network (ANN) and its improved versions were employed in detecting, classifying, and diagnosing the faults in industrial machines in the quickest possible time to decrease the shutdown time and operational costs along with the possible spreading of faults in others areas of machines. Deep learning-based mechanical faults in rotary machines were identified and diagnosed by the researchers in [33] and [34], while an adaptive neuro-fuzzy inference system (ANFIS) intelligent tool was adopted in [35], [36], and [37]. Moreover, a genetic algorithm (GA) optimized artificial immune system (AIS) was proposed in [38] for bearing fault detection based on vibration data of motor in three steps, where the features were extracted in the first stage using signal processing technique, data was processed then, and finally, the status of the motor is decided using GA-AIS algorithm. Although there were some usages of intelligent techniques in fault detection, classification, and diagnosis in rotary machines in recent studies, these were implemented for the motors individually rather than in a complete EV environment. In reference [39], open and short circuits fault detection and diagnosis procedures for an IM integrated into the EV platform were demonstrated using long short-term memory (LSTM). However, the technique was not suitable for real-time application as it took a long execution time. Furthermore, some research was conducted using data-driven statistical and artificial intelligence-based algorithms in estimating the state of charge (SOC) of EV batteries, fault diagnosis in situations of over-discharging, and so on within the EV environment as referred in [40], [41], [42], and [43].

Addressing these research gaps will contribute to advancing the field of electrical fault detection and classification using ensemble machine learning, leading to more accurate, robust, and practical solutions for ensuring the reliability of electrical systems.

Additionally, the use of nonlinear dynamic indicators is getting popular in fault detection. In a recent research, the authors proposed a multivariate Higuchi fractal dimension (MvHFD) to characterize the complexity of multichannel time series in fault detection. They achieve a recognition rate of 100% for signals in three features, which is at least 17.2% higher than for other metrics [44]. In another study, Li et. al. proposed a variable-step multiscale single threshold SloEn (VSM-StSloEn) [45]. The performance of SO-VSM-StSloEn has been validated through both simulated and real-world experiments. The findings indicate that SO-VSM-StSloEn exhibits insensitivity to signal length and is unaffected by threshold variations. Moreover, it showcases stronger discriminative capabilities compared to similar enhanced algorithms based on SloEn. Additionally, in classifying various categories of real-world signals, SO-VSM-StSloEn outperforms other commonly utilized entropy measures.

In summary, the adoption of more environmentally friendly transportation systems, exemplified as EVs, is a crucial approach to address the ecological consequences of emissions from fossil fuels. The complex interaction between electrical and mechanical elements in EVs requires the implementation of strong fault detection and classification techniques in order to guarantee the safety and dependability of their operations. Current developments in the field of artificial intelligence, namely the utilization of ensemble machine learning models, have demonstrated potential in addressing the challenges associated with real-time fault identification, classification, and diagnosis under various operating situations. Although classical methodologies and specialized techniques have been previously employed, these current breakthroughs offer a promising avenue for overcoming existing limitations. Addressing these deficiencies has the capacity to advance EVs as a fundamental component of environmentally sustainable urban transportation in accordance with the objectives of intelligent and environmentally conscious urban areas. Therefore, in this article, several ensemble ML algorithms are developed, compared, and proposed in identifying and categorizing the faults that occurred in the connections between $3-\phi$ inverter output and BLDC motor during operational mode in EV system. The aim of our approach is to detect and classify the faults during driving mode in the quickest possible time to provide sufficient information to take appropriate action in advance to prevent severe hazards. The main contributions of this research are summarized as follows:

- An efficient ensemble machine learning framework is developed using MATLAB and Python for real-time fault detection and classifications of faults in EVs.
- The data in healthy and faulty conditions during the drive mode in various operating moments are accumulated in an efficient manner to transfer to the Python framework.
- The presented methodology demonstrates a high level of effectiveness in efficiently detecting and classifying faults, hence enabling extensive real-time monitoring of electric EV systems.
- In the real-time drive mode of EV, the tools were implemented in detecting and classifying the faults that were arbitrarily generated in specific positions in different time and operating modes. The developed models exhibited superiority in fault detection and classification compared to previous work, which was guaranteed with several performance indices.

II. METHODOLOGY

The behavior of the motor changes according to the number of variables. We accumulated the vehicle's measured speed, voltage, current readings, output speed, and the motor's Hall-effect sensor output to parameters to prepare the required datasets for modeling the ML classification. Afterward, we developed and implemented several machine learning classifier algorithms to classify and detect vehicle conditions in real-time operation.

A. OVERVIEW OF EV SIMULATION APPROACH

The EV consisted of eight primary sections in the simulation-slider gain, controller, buck converter, three-phase inverter, commutation logic, sensor, BLDC motor, and vehicle body sub-components. The overall connections among these components in the EV model are depicted in Figure 1. Each of the parts of the EV configuration is described in the following sub-sections.

1) SLIDER GAIN

The slider gain block permits a change in scalar gain via the slider during simulation execution. In our work, the slider gain block controlled the vehicle's acceleration and deceleration.

2) CONTROLLER

The PID controller is the most prevalent control method widely used in industrial control applications to regulate temperature, flow, pressure, and other variables. In this work, the speed of the EV was regulated by employing a PID controller. Mainly, the controller's output regulates the motor's speed, which is adjusted based on the difference between the actual and targeted speed. However, the tuning of the PID controller mainly depends on the vital weight parameters, which are linked with proportional, integral, and derivative actions. A simple PID controller with input-output relation is sketched in Figure 2.

3) BUCK CONVERTER

Buck converters are mainly integrated into a system to reduce the output voltage as compared to the corresponding input voltage. Modulated DC voltage was obtained as the output of

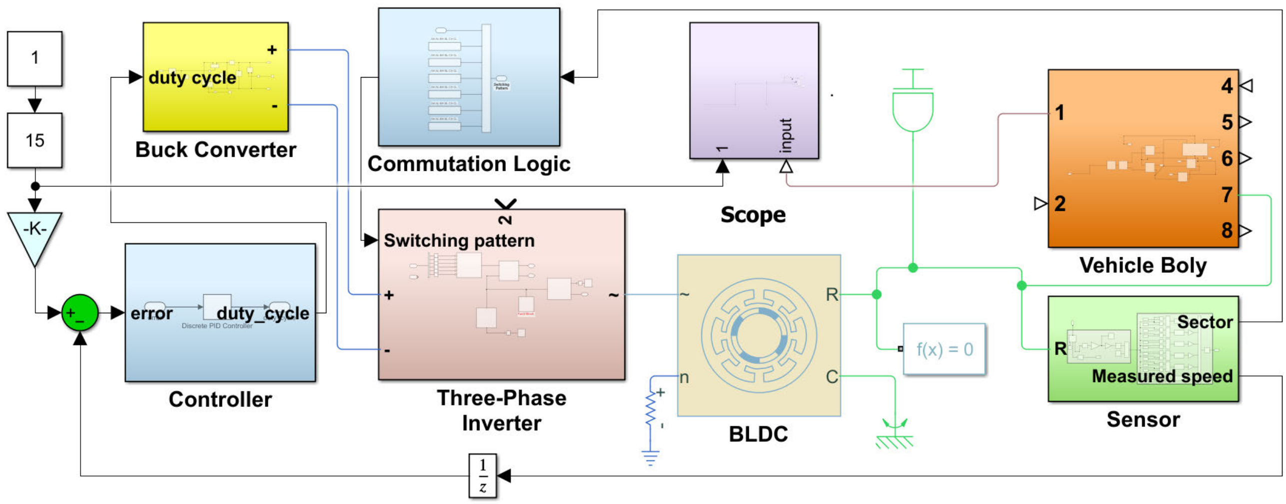


FIGURE 1. Electric vehicle model in MATLAB simulink.

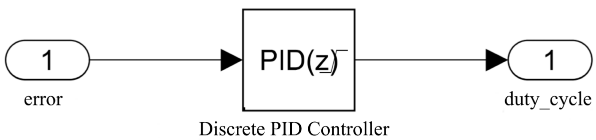


FIGURE 2. Simple PID controller with input-output relation.

the buck converter, where the input voltage was supplied from a DC voltage source. The level of the modulated DC voltage was controlled using the pulse width modulation (PWM) technique based on a pre-determined sampling frequency. The output and input of the buck converter were measured using two voltage sensors. In the simulation procedure of the buck converter, the logic needed to operate the converter was generated using Boolean data-type functions. The complete buck converter operation was designated as just an input-output port by integrating all the components in a block. The operational connections of the converter are shown in Figure 3.

4) HALL EFFECT SENSOR

One of the most efficient strategies for BLDC motor is to identify the rotor position using a Hall sensor. In our investigation, the Hall effect sensor was adopted in measuring the magnetic field induced in each phase to evaluate the vector position of current around 360° of rotation. However, a complete rotation is divided into six segments by equally separating the angles. In this case, the rotor position at the start is assumed to be fixed in the first segment, which is indicated as 'Segment 1', and defined as the location between '0' and '60' degrees. Figure 4 illustrates the change of rotor positions and corresponding angular areas. The locations of the rotor in different particular moments are evaluated based on the following logic as sketched in Figure 5, where the positional angular value of the rotor is reset to zero degrees after completing each rotation.

5) COMMUTATION LOGIC

The commutation logic used in the EV operational procedure is the chronological and sequential order of all potential switching patterns to rotate the rotor depending on segment information. This logic and possible order of switching operations are sketched in Figure 6. As the sensor provides the information on the rotor's segment position, the commutation logic executes the switching accordingly to generate the appropriate inverter output to supply power to the BLDC motor for rotating.

6) BLDC MOTOR AND INVERTER CONNECTION

A BLDC is constructed as a synchronous DC motor, namely a synchronous motor that receives DC electric power to the rotor in producing magnetic fields, which is commutated electronically in adjusting its speed and torque. Its construction and rotation are similar to a permanent magnet synchronous motor (PMSM) or a switched reluctance motor (SRM), except for the electronic speed control, which is controlled electronically. The electronic Controller mainly adjusts the magnitude and phase of DC pulses supplied to the rotor to vary the speed of the rotor smoothly. However, on the other hand, a three-phase power with a particular pulse in shape is supplied to the stator of the BLDC motor from a DC source through a three-phase inverter, as shown in Figure 7.

In this case, the energizing sequence of the stator using three-phase power is maintained according to the commutation logic generated based on the measured segmental positions of the rotor around 360 degrees through sensing by the Hall effect sensor. The switching frequency of a high-frequency MOSFET-based three-phase inverter is controlled following the commutation logic through inciting

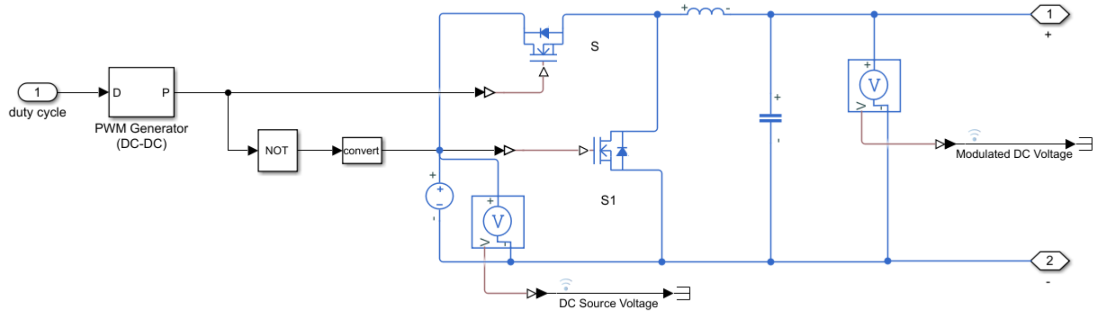


FIGURE 3. Buck converter.

If	Then
$0^0 < \theta \leq 60^0$	Segment = 1
$60^0 < \theta \leq 120^0$	Segment = 2
$120^0 < \theta \leq 180^0$	Segment = 3
$180^0 < \theta \leq 240^0$	Segment = 4
$240^0 < \theta \leq 300^0$	Segment = 5
$300^0 < \theta \leq 360^0$	Segment = 6

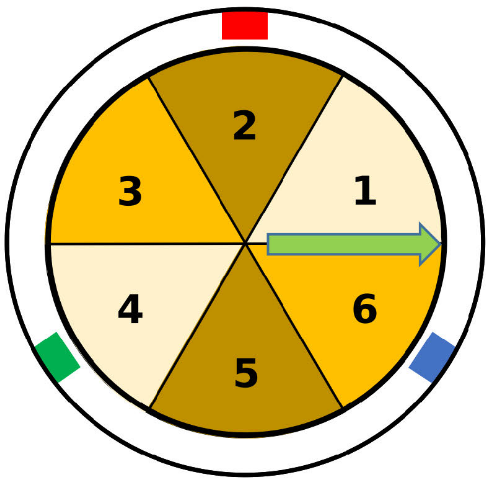


FIGURE 4. Rotor positions change of Segmental with angular area.

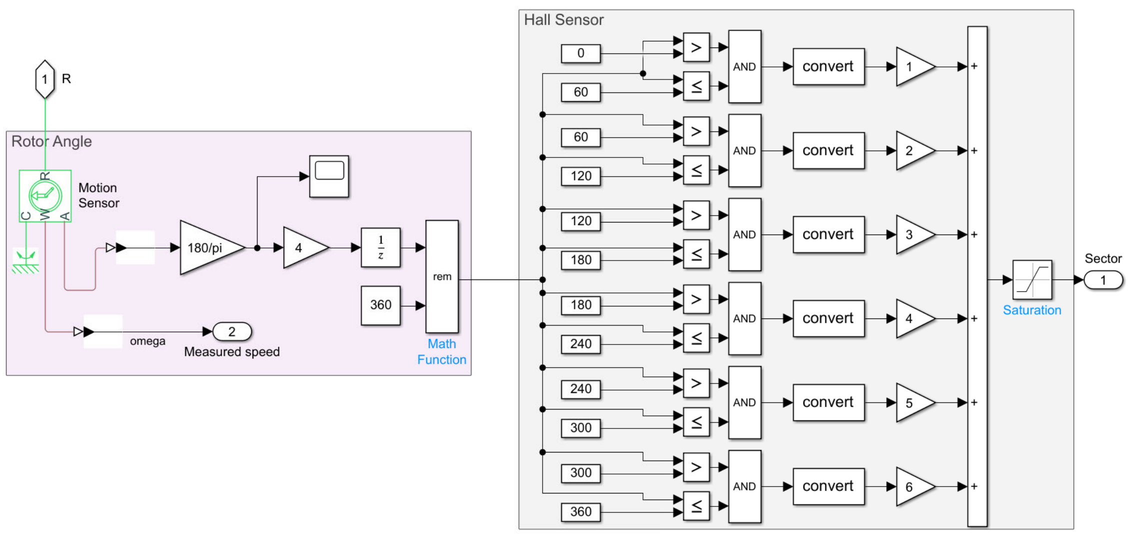


FIGURE 5. Logic diagram of hall effect sensor activation.

the driver circuits of the gate of MOSFETs. The three-phase connections to the stator of the BLDC motor through MOSFET, along with Hall sensor output-based logic controller connections, are sketched in Figure 7.

Unlike the other DC motors, especially the brushed DC motor where the electrical connections with brushes cause sparking, it offers the surpassing sparking issue due to avoiding brushes' usage and making it is suitable for use in

Commutation Logic					
AH	AL	BH	BL	CH	CL
1	0	0	0	0	1
0	0	1	0	0	1
0	1	1	0	0	0
0	1	0	0	1	0
0	0	0	1	1	0
1	0	0	1	0	0

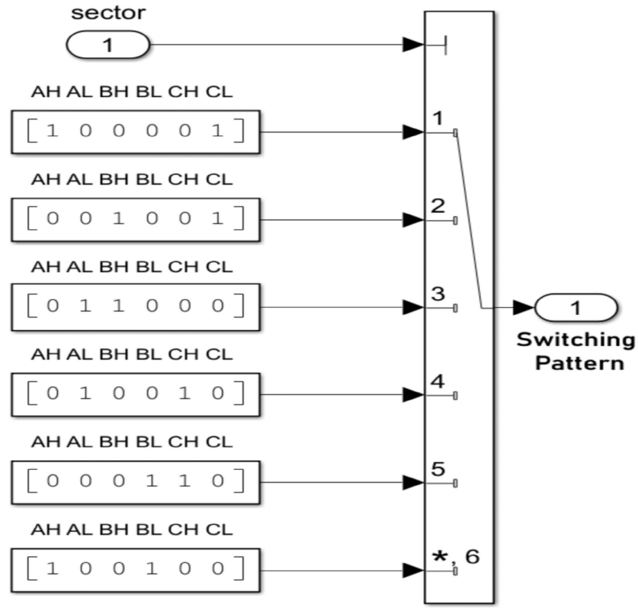


FIGURE 6. Commutation logic and switching patterns.

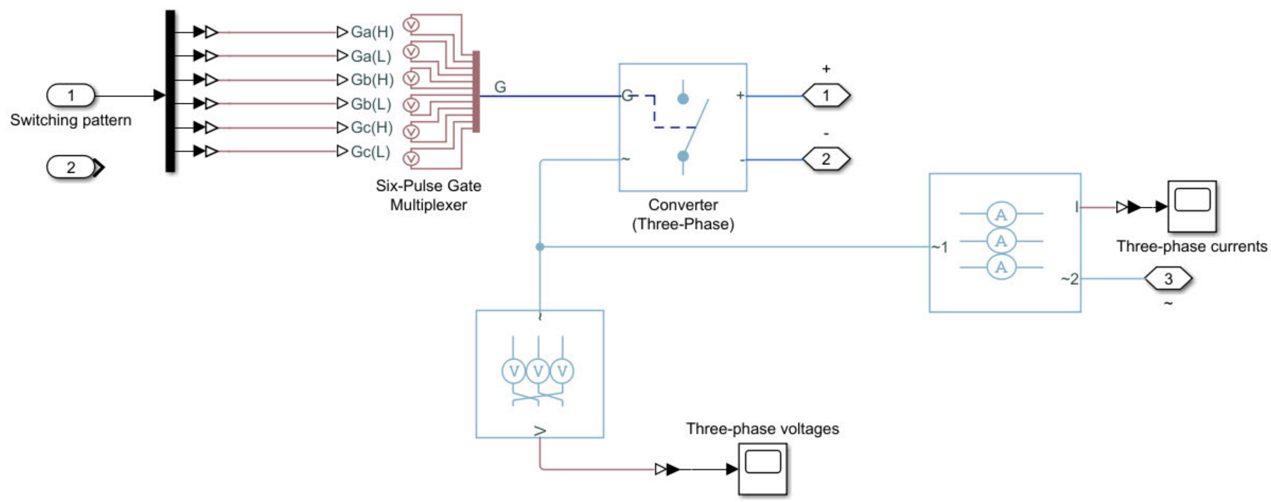


FIGURE 7. Three-phase inverter.

explosive areas. Additionally, the electronic control of motor speed can make the overall speed control smoother, more reliable, easier, and more suitable for complex applications.

7) MECHANICAL CONSTRUCTIONAL OVERVIEW

The overall mechanical body of an EV comprises four sub-divisional components- a basic gear system, a differential unit, tires, and the vehicle body. The EV integrates a fixed-ratio gearbox as the simple gear connected to the differential component. This differential employs a planetary bevel gear train with a transmission bevel gear linked by a pinion gear between the driveshaft and carrier. Although our model, implemented virtually in the MATLAB environment, excludes factors like inertia and driving losses, these can be optimally incorporated to emulate real-world EV movement. Our EV design encompasses tire behavior influenced by longitudinal motion and road interactions, factoring in a formula and accounting for tire inertia, damping, and stiffness. Moreover, our fault detection and classification analysis uses two axles to represent four-wheeled EVs. This approach captures the overall body mass and characteristics introduced by EV acceleration and road interactions

B. MACHINE LEARNING CLASSIFIER

1) DECISION TREE

The decision tree (DT), also known as a category tree, provides a straightforward representation for various processes. Utilizing real-time data from phasor measurement units

(PMUs), a DT algorithm identifies online security conditions in interconnected power networks, enhancing operational stability [22]. DT algorithms, including CART and ID3, are also applied to classify vehicle accident severity [46] and analyze customer data for smart city energy planning [22]. This method involves breaking down samples based on a set of rules, recursively applied using the DT algorithm to partition data into classes or categories. Here, we used '*max_depth=5*' to determine the maximum depth of the decision tree. It limits how many levels or splits the tree can have, and '*min_samplesleaf=5*', defines the minimum amount of samples necessary in a leaf node. It helps control the granularity of splits in the tree.

2) LOGISTIC REGRESSION

Logistic Regression (LR) is a machine learning classification algorithm suitable for binary or multiple class assignments. In our fault detection and classification approach, we employ binary classification iteratively to build the LR model for distinguishing fault classes from healthy conditions. We use a sigmoid function-based LR to assign '0' or '1' values based on a threshold, representing the two classes. Initially, we ascertain the presence of a faulty condition, and if detected, subsequent steps classify fault types. The sigmoid function in (1) governs the LR's input-output relationship.

$$y_i = \frac{1}{1 + e^{-}(B_0 + B_i x_i)} \tag{1}$$

where B_0 is the co-efficient constant and B_i is the weight vector of the corresponding input vector x_i. It is employed in the logistic regression model in estimating the probability membership value (PMV) for each input dataset/operating point. These PMVs were then used in the threshold function for detecting and categorizing the machine's condition. In our work '*penalty='l2'*' as an argument while initializing the logistic regression model to signal that Ridge (L2) regularization applies to the model's training process

3) STOCHASTIC GRADIENT DESCENT

The Stochastic Gradient Descent (SGD) classifier is a versatile learning method for different classification loss functions and penalties. It effectively fits linear classifiers and regressors, including Support Vector Machines and Logistic Regression. SGD focus on large-scale learning is highlighted in recent research [47], with applications in text classification and natural language processing [48]. In our fault analysis, We utilized parameter '*alpha=0.0001*', which implies a relatively tiny learning rate, suggesting that each step in the optimization process is small, which can assist the optimization converges more stably. And '*max _iter=1000*' says that the optimization process will go through a maximum of 1000 iterations.

4) ADABOOST

Adaptive Boosting (AdaBoost) is a classification tool that provides efficient results by combining several weak ones introduced in [49] and then modified in [50]. The algorithm begins with a weak learner that gives equal weight to each sample. Although the AdaBoost algorithm is accomplished by applying different weights, $w_1, w_2, \ldots w_n$, to each of the training samples, that is a process known as boosting the iterations, initially it starts to simulate with weak learner setting an equal weight value, $w = \frac{1}{N}$, to each of the total samples (N). However, the sample weights are changed at each iteration, and this procedure is repeated till to the end of the simulation/process. In this regard, misclassified examples result in an increase in their weights for the following round, while correctly classified examples are caused by decreasing their weights. Finally, the predictions are combined to provide the final prediction. To update the weights, an exponential loss function is devised. Following that, several studies were conducted to improve the performance of the AdaBoost algorithm as referred in [51] and [52]. We used the parameters '*n_estimators=50*', '*learning _rate=1.0*', and '*algorithm='SAMME.R'*' to configure and control the behavior of the AdaBoost algorithm.

5) XGBOOST

The eXtreme Gradient Boosting (XGBoost) is a well-known supervised classification tool that comprises the algorithmic trees in the classification procedure. It functions in predicting a target variable accurately by combining the estimates of a set of simpler and weaker models. Using gradient-boost decision trees, the performance achieved by the XGBoost is superior, along with better speed of execution. It is a way to boost the machines or to apply boosting to the machines, which was first proposed in [31]. Mainly, it is comprised of three major gradient boosting techniques, namely gradient boosting, regularized boosting, and stochastic boosting. Furthermore, it permits the addition and adjustment of regularization parameters, which distinguishes it from other libraries. The method is extraordinarily efficient in reducing processing time and optimizes memory resources to make it better at working as a machine learning model, as reported in [53], [54], and [55]. It also supports parallel structure in building the trees and is the only technique that can adapt and boost the data that has newly been added to the trained model (called "Continued Training") [56]. For this task, we specified the number of boosting rounds or decision trees as '*n_estimators=100*', Using the value of '*learning_rate=0.1*' to regulate the size of each step. To establish the highest depth for every decision tree, simply utilize the parameter '*max_depth=3*'. '*subsample=0.8*' to specify the fraction of samples used for training each individual tree and specify the fraction of features used for training each individual tree, use '*colsample_bytree=0.8*'.

6) K-NEAREST NEIGHBOR

It is one of the most efficient and widely used classification and regression machine learning (ML) tools introduced in [57]. The algorithm has been modified and improved by many researchers to make it more convenient for practical applications, particularly in case of accuracy and execution time as is reported in [58], [59], [60], and [61]. Different types of KNN variants were discussed in [61], while the reference [58] summarized and analyzed the challenges of the KNN algorithm including the different types of 'k' selection approaches, nearest neighbors searching cases, and classification approaches. Accuracy was investigated using the KNN algorithm with an appropriate selection of 'k' value in [60]. Additionally, two optimal 'k' value selection procedures were discussed in [59], where the approaches were constructed based on a decision tree. However, in our research, we have chosen variable 'k' and selected the most optimal one based on the training dataset and, used it during the testing period. In addition to calculating the nearest neighbors, we have followed the Euclidian distance measurement approach. This procedure can be explained as follows: If the training dataset 'S' contains multiple samples 'x_i' along with the class label 'y_i', then the set can be expressed as:

$$S = (x_i, y_i); i = 1, 2, 3, \cdots N \tag{2}$$

$$y_i \epsilon m_j; j = 1, 2, 3, \cdots M \tag{3}$$

where 'N' is the number of training samples and 'M' is the number of classes for which the dataset is to be classified using KNN. In case of the dataset consists of 'p' number of features, then the training and testing data can be represented as follows:

$$x_i = x_i^1, x_i^2, x_i^3, \cdots x_i^p \tag{4}$$

$$x_t = x_t^1, x_t^2, x_t^3, \cdots x_t^p \tag{5}$$

Therefore, the Euclidian distance can be calculated using the following formula:

$$d_{t,i} = \sqrt{\sum_{q=1}^{p} (x_t^q - x_i^q)^2} \tag{6}$$

where, 'd' is the Euclidian distance for a particular test sample towards 'N' number of training samples, from which distance set, 'k' number of nearest neighbors were considered to classify the test data point. In our investigation, we used '*leaf_size=30*', which refers to the size of the leaf nodes in the KD-Tree, and the'*metric='minkowski*'', a parameter that determines the distance metric used for computing distances between data points in KNN.

7) VOTING CLASSIFIER

An effective machine learning model can be a soft voting classifier with decision tree, logistic regression, stochastic gradient descent, AdaBoost, XGBoost, and K-nearest neighbors algorithms. A well-known approach for binary classification, logistic regression, decision trees, and stochastic gradient descent can handle complicated feature interactions. Multiple weak models can be combined in AdaBoost to improve the performance of weak learners, and k-nearest neighbors can find patterns in the data and forecast the future based on close occurrences. A more reliable and accurate model can be produced by combining these algorithms in a soft voting ensemble since they can each be used to their best advantage. The outcome can be determined based on the likelihood of each base model using the soft voting method. With the soft voting method, the classifier's overall accuracy and robustness can be increased because the final prediction can be based on the probabilities of each base model. A soft vote classifier can be a strong tool for addressing various classification issues when the correct models are combined.

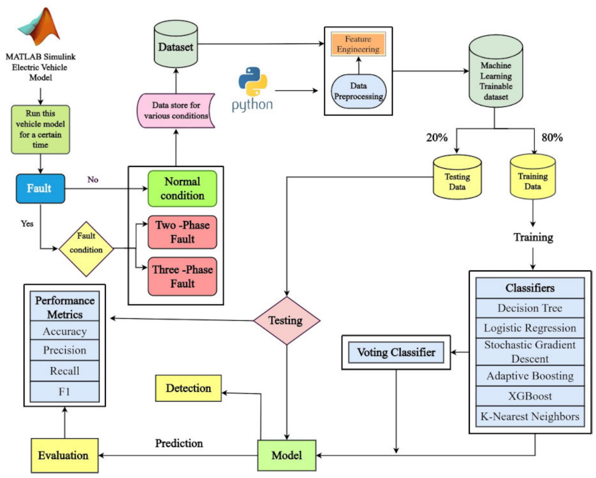


FIGURE 8. Workflow of our study.

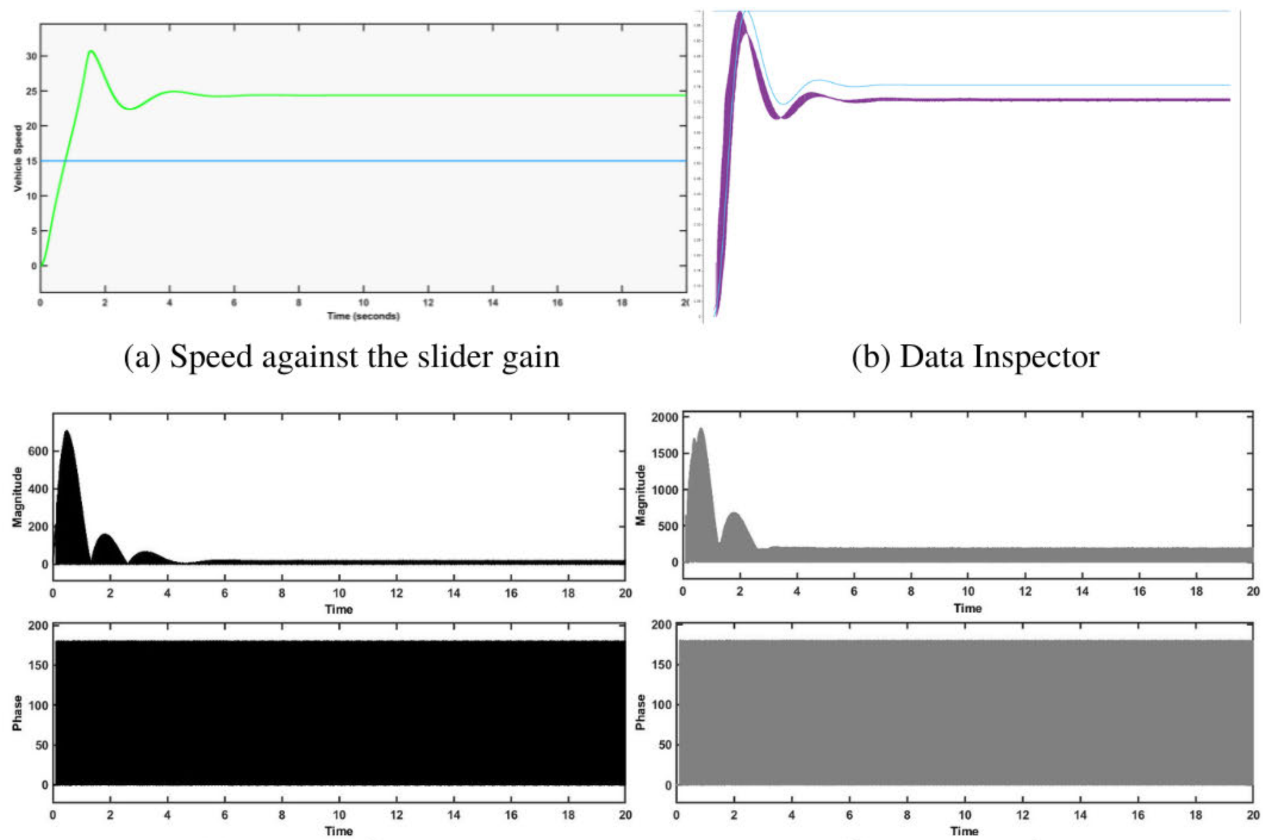
In this study, we have utilized MATLAB to simulate an electric vehicle model. The established Simulink model was intended to operate in both states - normal and faulty. The overall workflow of the proposed machine learning models is sketched in Figure 8. The framework involves constructing EV models and conducting simulations encompassing both normal and different fault scenarios. The resultant datasets are meticulously preprocessed and subsequently partitioned into distinct training and testing sets. Through this approach, an array of ensemble classifier algorithms are adeptly trained using the training data, with the resultant models being diligently preserved for subsequent validation against distinct testing datasets.

III. DATA GENERATION AND MODEL DEVELOPMENT

A. NORMAL CONDITION DATA

In the Normal condition of our EV, we run the simulation model for 20 seconds to generate a variety of data to store the required parameter values of the EV to develop the dataset for employing the ML models.

During no-fault condition, all components functioned seamlessly. Figure 9 (a) displays the constant speed at which the slider gain operates, while Figure 9 (b) shows the output DC modulated voltage and measured speed of the buck converter input for the DC source voltages. Our simulation model employs a 201.6-volt DC voltage source obtained from



(a) Speed against the slider gain
(b) Data Inspector
(c) Inverter Currents
(d) Inverter voltages

FIGURE 9. MATLAB model output for normal condition.

the output of the buck converter. Figure 9(c) and (d) illustrate the variable currents and voltages of the inverter, respectively.

B. FAULT GENERATION

The fault mainly arises along the connection path between the three-phase inverter and the BLDC motor. In Figure 10, we included a block from the Simulink library that can initiate faults for generating faulty data. We connected this block to the terminal of the three-phase inverter, which delivers power to the Brushless DC motor. This block can represent any combination of single, double, or three-phase grounded or ungrounded faults. Faults can be initiated at preset times or via the external fault input. The fault block parameter is modifiable using block properties. In case of a fault occurring, the properties of the block modify fault types and enable the temporal trigger to reflect the fault's anticipated duration on the three-phase inverter to the BLDC motor.

Table 1 summarizes the EV running duration for each case, fault initiation moment, and fault duration maintained in seconds for different situations. EV run time displays the entire simulation runtime for each condition, which is 20 seconds for each of all three cases.

During the EV's normal operating condition, when no issues are considered, the EV moves in normal mode. In the case of two-phase and three-phase fault conditions, the fault initiates in the EV at the 5th second during the simulation and lasts for 10 seconds. Then, it operates normally for both faulty situations until the end of 20 seconds. We assessed the simulation runtime under normal and fault situations, noticed the commencement and duration of faults in the EV system, and recorded the data for the ML classification.

TABLE 1. MATLAB model run-time and fault duration.

Name	Normal Condition	Two-Phase Fault Condition	Three-Phase Fault Condition
EV run duration (Sec)	20	20	20
EV fault start (Sec)	N/A	5	5
EV fault duration (Sec)	N/A	10	10

1) TWO-PHASE FAULT

The two-phase fault condition moments are shown in Figure 11. In this case, the EV starts with a normal running mode and then a two-phase fault is initiated at the 5th second for a duration of 10 seconds according to the pre-determined

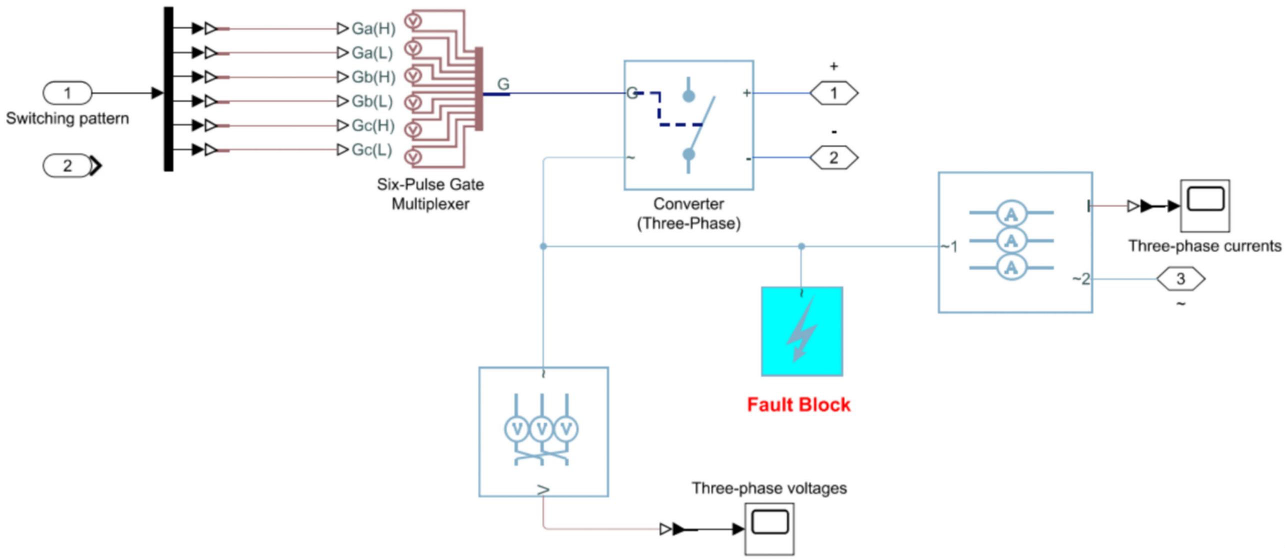
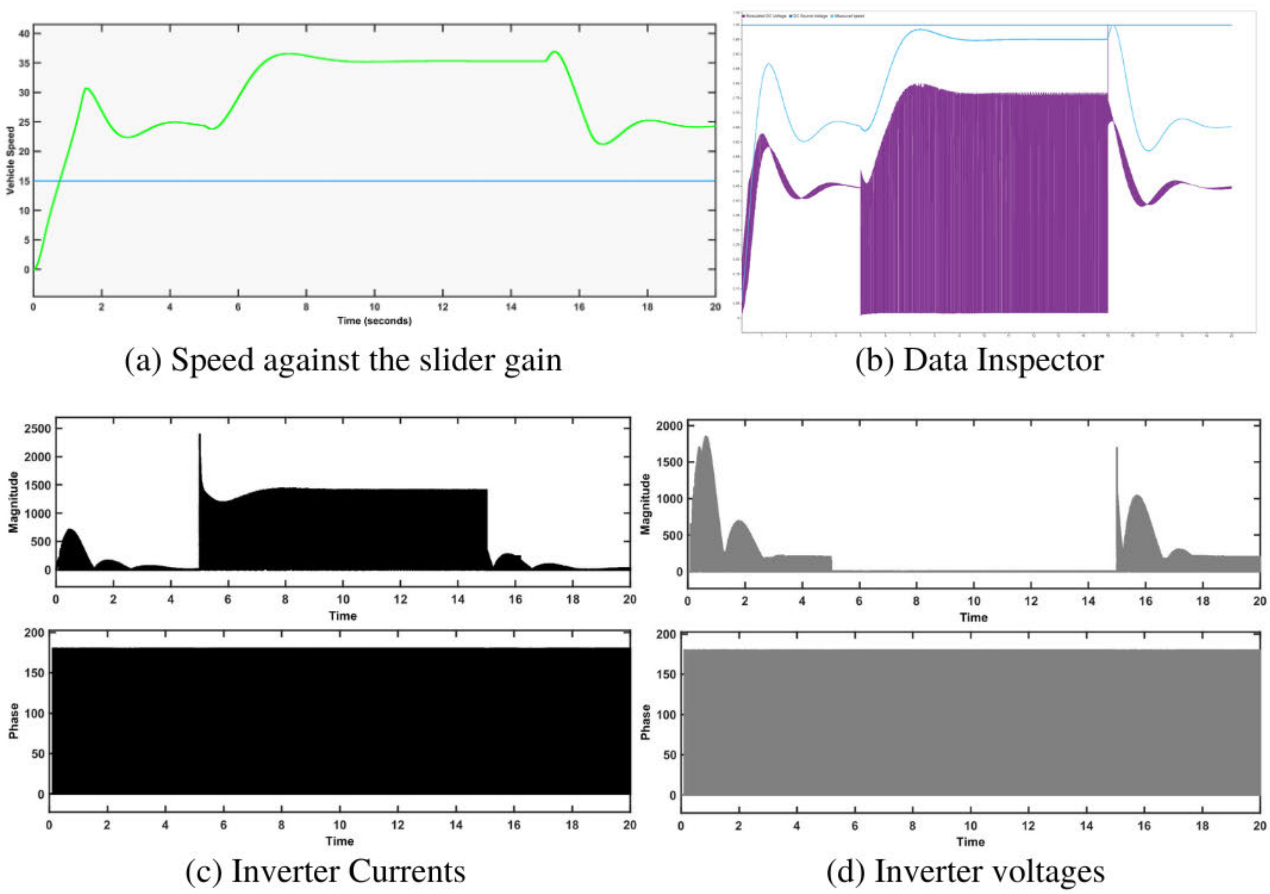


FIGURE 10. Fault block between connection $3-\phi$ inverter and the BLDC motor.



(a) Speed against the slider gain
(b) Data Inspector
(c) Inverter Currents
(d) Inverter voltages

FIGURE 11. MATLAB model output for two-phase fault condition.

condition given to the fault initiation simulink block while it goes at high speed as is shown in Figure 11 (a) after completing ten seconds of fault period at the fifteen seconds, it becomes stable as normal condition. Moreover, Figure 11 (b) displays the variation of output DC modulated voltage and measured buck converter input speed for the DC source

TABLE 2. Data Collection Details.

Sample Type	Number of Instances	% of Total Data	Flag Indication
Normal Condition	400000	40%	'0'
Two-Phase Fault Condition	300000	30%	'1'
Three-Phase Fault Condition	300000	30%	'2'

voltages in the data inspector. Figure 11 (c) and (d) represent the graphs of fluctuating values of the inverter current and voltage during the faulty operation.

2) THREE-PHASE FAULT

During the Three-Phase Fault state, the Electric Vehicle operates differently than during the Normal and Two-Phase Fault circumstances seen in Figure 12. In Figure 12 (a), When the preset problem occurs in the fifth second, the car rapidly decelerates and comes close to stopping. After ten seconds of fault operations, the vehicle regains speed and usually behaves.

In Figure 12 (b), it is displayed the output DC modulated voltage fluctuation and measured buck converter input speed for DC source voltage in the data inspector for the three-phase fault case. The fault operation graphs for the inverter current and voltage are sketched in Figure 12 (c) and (d).

C. DATA COLLECTION

In developing the ML classification model for identifying two-phase and three-phase faults that occur in BLDC motors integrated into EVs for normal operating moments of that corresponding EV, we prepared a set of data with millions of instances concerning time considering three situations - normal condition and two-phase and three-phase faults conditions.

In this regard, the EV was maintained in running mode in the MATLAB Simulink platform for 20 seconds in each condition. We generated 40% of the total data in normal conditions, and 30% of the total data was generated for each fault condition. Unique flag indication was mentioned for each of the three categories of datasets as '0' for normal data, '1' for a two-phase fault (line-to-line fault) data, and '2' for the three-phase fault data, and was combined to make a single dataset. Details of the data generation and collection are summarized in Table 2. We considered six features, including voltages and currents of each of the three phases, Hall-effect sensor output, modulated DC voltage, slider gain output, and motor speed, in the case of all three modes of operations. Three-phase voltages and currents were measured from the three-phase inverter output terminal. In contrast, the modulated DC voltage was the output of the buck converter for the input DC voltage supplied to that corresponding converter.

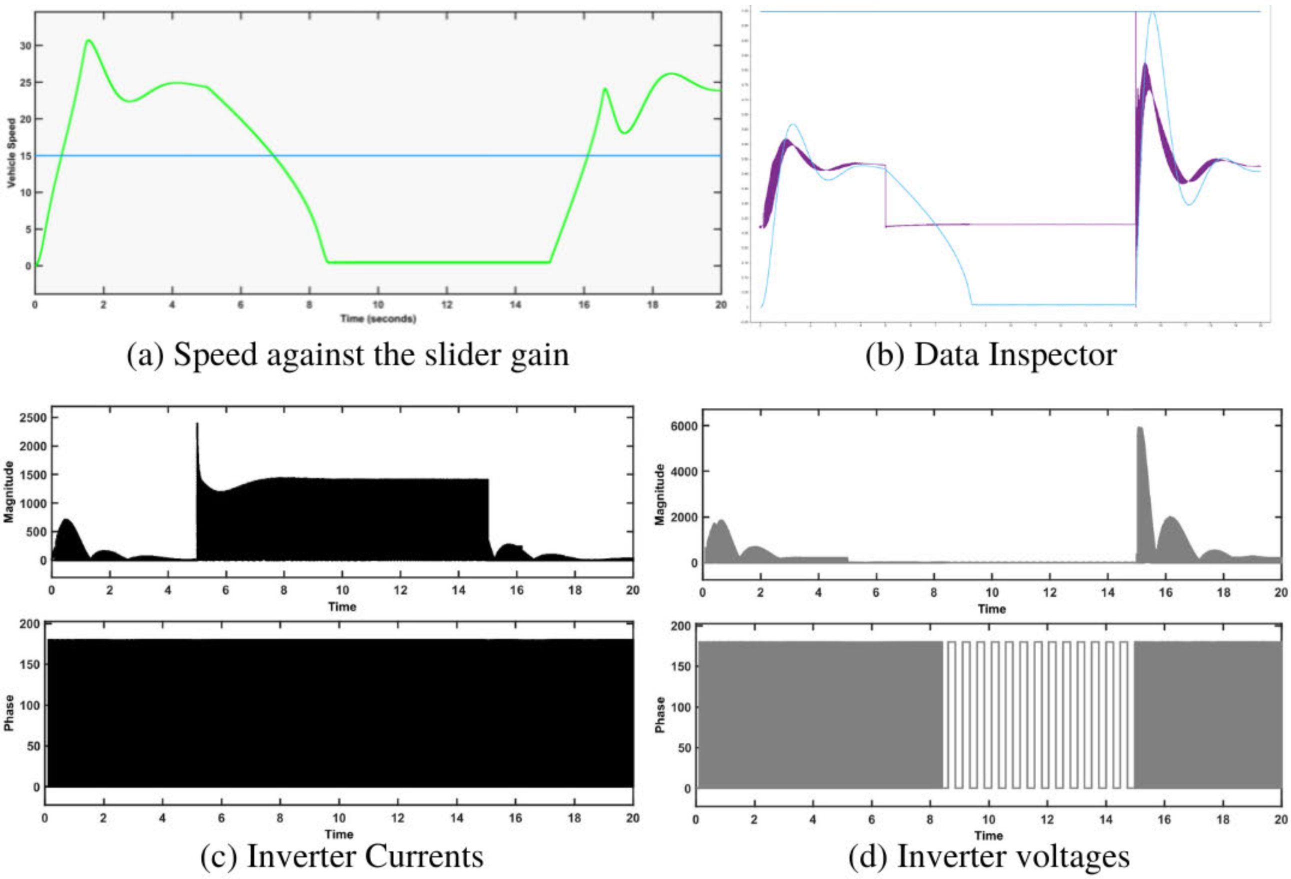
Our dataset uses a feature called "Target" to classify the various vehicle statuses. In this section, we categorize our generated data based on several conditions. We classify it as 0 for the normal condition, 1 for the two-phase condition, and 2 for the three-phase situation. The time the data was measured for each sample, three-phase current, voltage, Hall effect sensor value, output speed, modulated DC voltage, measured speed, and the "Target" case corresponding to EV operating condition were considered for preparing the dataset. The sensor provides the Hall sensor values necessary to determine the rotor's location, while the output speed is the EV speed designed in the Simulink platform. On the other hand, the measured DC voltage compares the output voltage of the Buck Converter to the DC source voltage. However, the estimated speed corresponds to the speed values determined by the input slider gain.

D. CLASSIFICATION MODEL DEVELOPMENT AND EVALUATION METRICS

In this research, six machine learning classification models, namely the Decision Tree (DT), Logistic Regression (LR), Stochastic Gradient Descent (SGD), AdaBoost, XGBoost, and K-Nearest Neighbor (KNN), were taken into consideration for developing the proposed ML classification model employing the dataset generated in MATLAB environment to detect and classify the faulty conditions happened in EV configuration. The proposed voting classifier (VT) was also developed as an ensemble ML classifier combining all the individual classifiers modeled in our research. Although the data was generated in the MATLAB platform, the models were developed using Python with a sci-kit-learn library, where 80% of the data was used for training, and the rest of the data was employed for testing purposes of the models. A systematic and gradual trial and error approach was adopted for tuning the parameters of the models of ML classification tools. Furthermore, 10-fold cross-validation was performed in the case of developing the tools to ensure the model's performance. As an indication of fine-tuning of the developed models, the confusion matrices of the proposed ML classification models were assessed at the end of the training and testing procedures. In our modeling, the classification tool, the simple average, was considered regarding obtaining the final model from the individually measured eight models in the case of 10-folded datasets. We also measured the accuracy, precision, recall, and F1 score for our model to ensure the performance of the developed model. The performance indices are explained in the following subsections.

1) CONFUSION MATRIX

The evaluation of classification model prediction performance for specific test data involves using a confusion matrix, which relies on four parameters derived from classification outputs. These parameters are "true positive (TP)," "true negative (TN)," "false positive (FP)," and "false negative (FN)." When forming a confusion matrix for data classified into two classes, typically labeled as "positive" and "negative," outcomes where a model predicts



(a) Speed against the slider gain (b) Data Inspector

(c) Inverter Currents (d) Inverter voltages

FIGURE 12. MATLAB model output for two-phase fault condition.

a data point as "positive" when it truly belongs to the "positive" class are considered as "TP". Conversely, if a model predicts a data point as "positive" when it's not in the "positive" class, it is deemed as "FP". Similar relationships hold for "negative" class determination with "TN" and "FN". In the context of detecting and categorizing faults in the EV platform, the flag "0" signifies normal EV operation, "1" represents a two-phase fault, and "2" indicates a three-phase fault. Predictions aligning with the actual normal state are counted as "true" cases, denoted as either "TP" or "TN," while any other predictions, such as "1" or "2," are considered as "false" cases, i.e., "FP" or "FN." This evaluation extends to the other cases in our fault analysis.

Considering three classes, we assess the performance of machine learning (ML) classification models for fault detection and classification in EV connections by constructing confusion matrices. Predictions of "0" flag during normal EV operation are labeled "true case 1," while predictions of "0" for fault cases are "false case 1." Similarly, predictions of "1" for two-phase fault are selected as "true case 2", and "false case 2" when the actual data belongs to two-phase fault condition. This approach extends in predicting for "true case 3", and "false case 3" for the three-phase fault condition of EV.

TABLE 3. Confusion matrix formation procedure.

Observations	Actual Values			
Predicted Values	Classes	Flag '0'	Flag '1'	Flag '2'
	Flag '0'	True Case 1	False Case 1	False Case 1
	Flag '1'	False Case 2	True Case 2	False Case 2
	Flag '2'	False Case 3	False Case 3	True Case 3

To comprehensively analyze the performance of each ML tool's development, we constructed complete confusion matrices, summarizing their outcomes according to Table 3. These matrices comprehensively assess how well the developed classification ML tools detect and classify faults.

Based on the implemented approach to construct the confusion matrix as it is described and tabulated for our research, the following Figure 13 shows the confusion matrices for the developed ML classification tools applied in the test dataset.

2) CONFUSION MATRIX BASED PERFORMANCE PARAMETERS ANALYSIS

$$Accuracy = \frac{(TrueCase\ 1 + TrueCase\ 2 + TrueCase\ 3)}{(TotalObservations)} \tag{7}$$

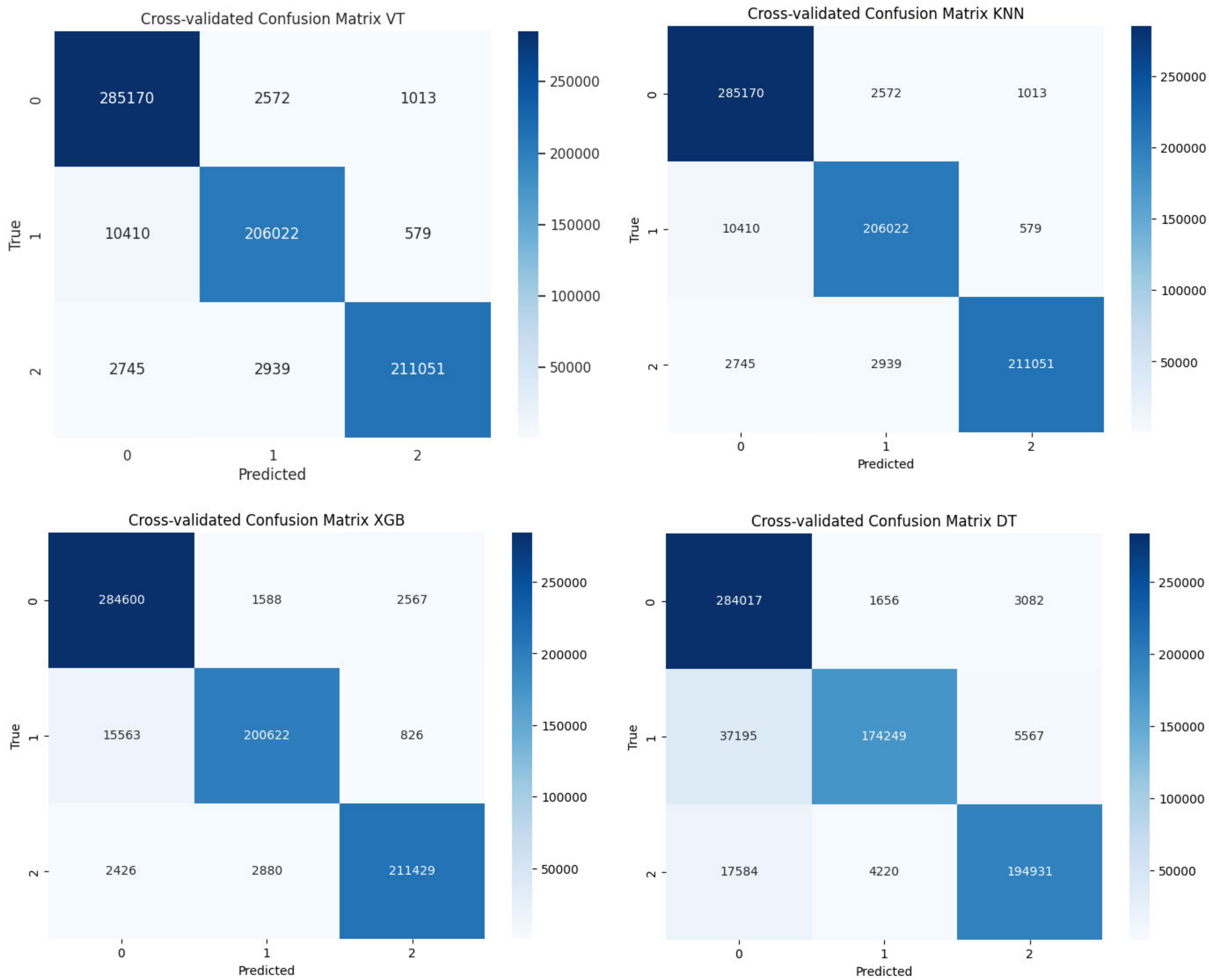


FIGURE 13. **Confusion matrices for ML classification based on accuracy.**

In the case of precision, it is a representation of total true predictions for a class out of the total number of times it is predicted in favour of that corresponding class. Although the unity value of precision measures the better performance for an ML classification model through classifying a particular category, it cannot alone confirm the perfectness in classifying data for that corresponding model because of overlooking the incorrect prediction of other categories. Therefore, the 'recall' is a performance parameter for classification models that needs to be introduced along with the precision in case to ensure the overall superiority in classifying an ML model.

$$Precision = \frac{(TrueCase\ 1)}{(TrueCase\ 1 + FalseCase\ 1)} \tag{8}$$

$$Recall = \frac{(TrueCase\ 1)}{(TotalFlag'0'Observations)} \tag{9}$$

However, similar to the precision, the recall also can't ensure the overall classification performance alone, although the unity value represents the better performance for a model. In this regard, it can be stated that a model having high precision and low recall can never be comparable in the case of superiority over other models having low precision and high recall. Therefore, another performance parameter, the F-1 score, which combines the impact of precision and recall at the same time, was also considered in this research to evaluate the true performance of the developed fault detection and classification ML models. The value of the F1 score becomes maximum for a particular ML classification tool when the model ensures the same value of precision and recall, and the best value of the F1 score is unity, while zero represents the worst case.

$$F1Score = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \tag{10}$$

The associated (7)-(10) were used in our research to measure the values of the described performance indices.

E. EXPERIMENTAL SETUP (MATLAB AND PYTHON)

Our modeling and simulation work was done on a Windows 10 Business 64-bit desktop with an Intel Core i7 processor, 16 GB of DDR4 Memory, and an NVIDIA GeForce GTX 1070 8GB DDR5 GPU. These data sets

TABLE 4. Performance measurement of different types of machine learning classifiers.

Classifier	Precision	Recall	F1-Score	Accuracy
DT	0.981	0.903	0.913	0.911
LR	0.856	0.802	0.814	0.824
SGD	0.786	0.809	0.813	0.803
AdB	0.889	0.886	0.892	0.889
XGB	0.952	0.949	0.954	0.952
KNN	0.974	0.971	0.967	0.971
VT	0.983	0.981	0.985	0.983

were generated using MATLAB R2021a version through the simulations of a BLDC motor powering a model of an electric vehicle and its driving control system. The scikit-learn library, written in Python 3.9.13, was used to model the machine learning classifiers and evaluate performance. Using MATLAB and Python made simulation and modeling much more manageable, while scikit-learn supplied a popular, well-documented package for machine learning and performance evaluation.

IV. RESULT AND DISCUSSION

A. MACHINE LEARNING BASED CLASSIFICATION

After performing the simulation model, we collect data from the MATLAB workspace and turn the raw data into a machine-trainable dataset. We grouped the data into normal conditions, two-phase fault conditions, and three-phase conditions. We preprocessed the dataset by applying a feature engineering technique that turns existing datasets into sets of figures needed for our job. We deployed six machine learning (ML) classifiers and one voting classifier for training and evaluating the performance of the algorithms. We employed a categorization report for measuring performance that includes - precision, recall, F1 score, and accuracy. Table 4 provides a clear illustration of the effectiveness of our developed ML classifiers.

The performance of each classifier was tested using precision, recall, F1 score, and accuracy measures. From Figure 14, we obtained that the voting classifier, which combines the other developed classifiers, yielded the highest precision 0.983, recall 0.981, F1 score 0.985, and accuracy 0.983. The ensembling ability to harness the capabilities of several classifiers resulted in accurate predictions over a wide variety of fault conditions.

K-Nearest Neighbors (KNN) displayed the second high precision of 0.974, recall of 0.971, and F1 score of 0.967, demonstrating its applicability for scenarios when class borders are well-separated in the feature space. XGBoost is a powerful method that often works effectively. Its intense precision of 0.952, recall of 0.949, F1 score of 0.954, and accuracy of 0.952 suggest that it can successfully handle the complexity. The Decision Tree classifier also displayed balanced performance with a good precision value of 0.918 and recall of 0.903, resulting an F1 score of 0.911 and an accuracy of 0.911. Decision Tree effectively caught the underlying patterns in the data and delivered accurate forecasts across a variety of fault circumstances.

TABLE 5. Comparison of fault diagnosis techniques.

Ref.	Precision	Recall	F1-Score	Accuracy
[62]	-	-	-	0.983
[63]	-	-	-	0.991 (AUC)
[64]	0.97	0.97	0.989	0.97
Proposed	0.983	0.981	0.985	0.983

Adaptive Boosting often performs well since it combines multiple bad learners into a strong one. The considerable precision value is 0.889, recall is 0.886, F1 score is 0.892, and accuracy is 0.889, which suggest that it can effectively capture the complicated relationships in the data.

In comparison, Logistic Regression displayed a moderate precision value of 0.856 and a recall of 0.802, resulting in an F1 score of 0.814. The linear character of Logistic Regression might have hampered its ability to capture complicated associations contained in the dataset, resulting the significantly lower performance. Moreover, the Stochastic Gradient Descent (SGD) also achieved a poorer precision of 0.786 and an F1 score of 0.813 along with a recall of 0.809. Overall, the proposed model (Voting Classifier) displayed higher performance, showing their usefulness in handling the complexity of the dataset.

In our study, the Voting Classifier emerged as the most effective model for fault detection in electric vehicles, integrating diverse algorithms such as KNN, DT, AdB, XGB, LR, and SGD. This ensemble approach capitalized on the unique strengths of each algorithm, achieving outstanding performance metrics with precision, recall, F1 score, and accuracy all exceeding 0.98. The robustness and high accuracy of the Voting Classifier across various fault conditions underscore its suitability for enhancing the reliability and safety of EV systems.

B. COMPARISON WITH OTHER TECHNIQUES

We compare our machine learning-based fault detection approach with existing methods in the literature. Table 5 summarizes the performance comparison.

Our approach, based on machine learning classifiers and a voting ensemble, demonstrates high precision, recall, F1-score, and accuracy in fault detection for electric vehicles. Compared to related studies, such as the work by Ali et al. on single- and multi-fault diagnosis using machine learning for variable frequency drive-fed induction motors [62], our method achieves comparable or higher performance across key metrics.

Additionally, the study by Principi et al. on unsupervised electric motor fault detection using deep autoencoders showcases alternative techniques with high accuracy (99.11% AUC) [63]. Meanwhile, the hybrid approach proposed by Toma et al. combining genetic algorithms and machine learning achieves robust results in bearing fault diagnosis [64].

Our method, leveraging diverse machine learning classifiers within a voting framework, offers an effective and versatile solution for fault detection in electric vehicles,

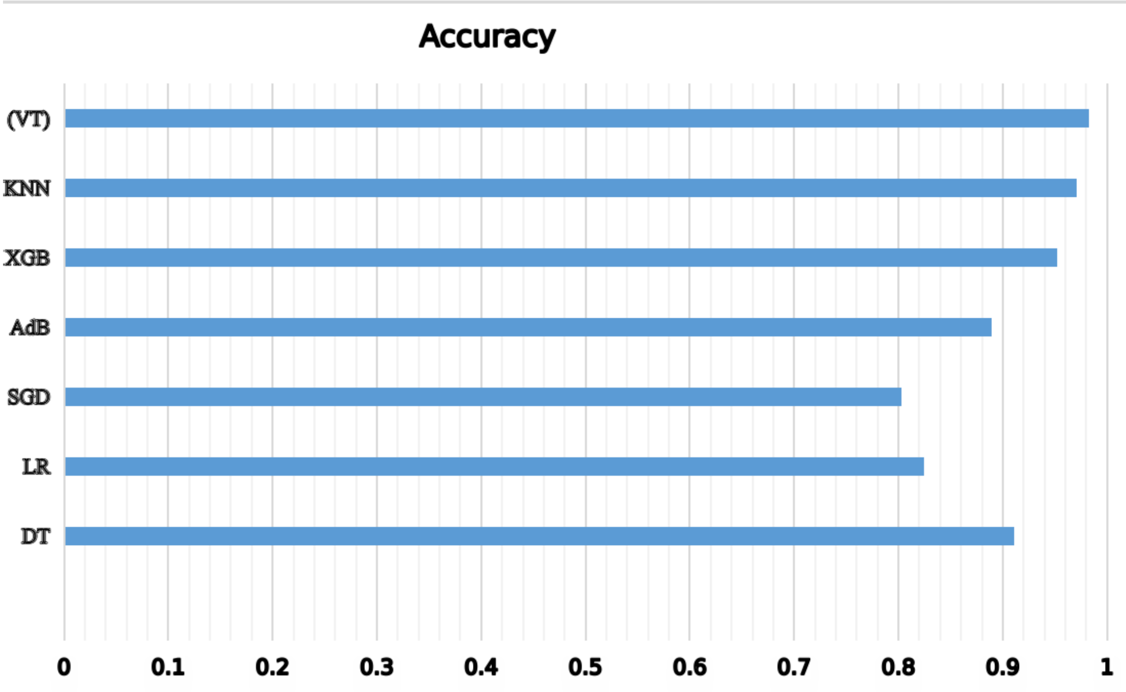


FIGURE 14. Performances graph of different types of classifiers.

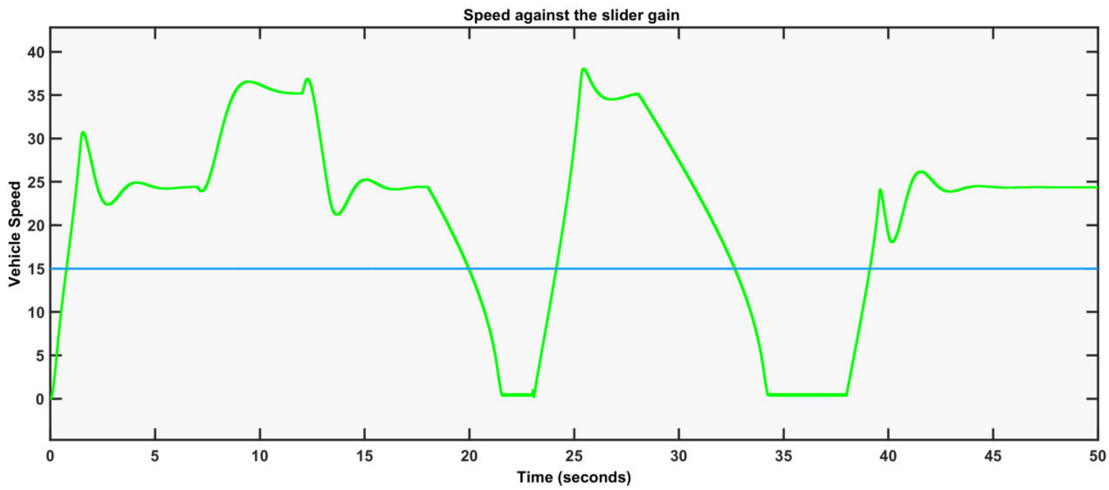


FIGURE 15. Simulation of multi-type conditions in Simulink model running mode of EV.

showing competitive performance against other state-of-the-art techniques in the field.

C. DETECTION

We ran our model with a fixed time set to test the efficiency of our proposed voting classifier (VT) model. We applied multiple vehicle conditions (normal, two-phase fault, and three-phase fault), and our proposed model successfully detected the vehicle's condition types as sketched in Figure 15.

V. CONCLUSION

This article has developed several machine learning (ML) classifiers to detect and classify faults in electric vehicle (EV) configurations immediately and accurately. In our analysis, we designed a prototype EV in MATLAB Simulink. We initiated necessary faults (two-phase and three-phase) in the configuration of the EV during its running mode to generate the required dataset for modeling the ML classifier tools. We mainly generated the faults between the connection of the inverter output to the motor of the EV during running mode and stored data for preparing the dataset, which considered six parameters for EV fault detection and classification investigation the ML algorithms, including decision tree (DT), logistic regression (LR), stochastic gradient descent (SGD), AdaBoost, XGBoost, K-nearest neighbor (KNN), and voting classifier (VT). Four statistical parameters, namely the accuracy, precision, recall, and F1-score, were calculated from the generated confusion matrices for evaluating the performance of the developed machine learning models in detecting and classifying the faulty condition of electric vehicles. In the case of individual ML models, the KNN outperformed other models as it maintained the highest accuracy, recall, and F1-score alongside slightly less precision value than the DT, where it was the highest one. On the contrary, the stochastic gradient descent (SGD) demonstrated comparatively lower performance than the others. Moreover, in the case of VT, the best performance was exhibited in all aspects of performance parameters (i.e. accuracy is 0.983, recall is 0.981, F1-score is 0.985, and precision is 0.983) in detecting and classifying the faults initiated in EV configurations compared to individually applied ML classifiers. However, as an extension of future work, some other faults, including malfunctioning the inverter itself, can be considered as they are crucial in powering the EV motor to be driven reliably; gearbox scratches or holes can also be regarded as mechanical faults. Extensive analysis, including additional features and performance parameters, can also provide more robustness of the fault detection and classification of electric vehicle operation in boosting more reliable transportation systems.

ABBREVIATIONS

The following abbreviations are used in this manuscript:

EV	Electric Vehicle.
ANN	Artificial Neural Network.
PWM	Pulse Width Modulation.
ML	Machine Learning.
AI	Artificial Intelligence.
BLDC	Brushless DC motor.
PID	Proportional Integral Derivative.
DT	Decision Tree.
PMU	Phasor Measurement Units.
LR	Logistic Regression.
PMV	Probability Membership Value.
SGD	Stochastic Gradient Descent.
KNN	K-Nearest Neighbor.
RF	Random Forest.
AdaBoost/AdB	Adaptive Boosting.
XGBoost/XGB	eXtreme Gradient Boosting.
VT	Voting Classifier.

REFERENCES

[1] H. S. Das, M. M. Rahman, S. Li, and C. W. Tan, "Electric vehicles standards, charging infrastructure, and impact on grid integration: A technological review," *Renew. Sustain. Energy Rev.*, vol. 120, Mar. 2020, Art. no. 109618.

[2] S. S. Mohammed, T. P. I. Ahamed, S. H. E. A. Aleem, and A. I. Omar, "Interruptible charge scheduling of plug-in electric vehicle to minimize charging cost using heuristic algorithm," *Electr. Eng.*, vol. 104, no. 3, pp. 1425–1440, Jun. 2022.

[3] M. M. Rahman, E. A. Al-Ammar, H. S. Das, and W. Ko, "Technical assessment of plug-in hybrid electric vehicle charging scheduling for peak reduction," in *Proc. 10th Int. Renew. Energy Congr. (IREC)*, Mar. 2019, pp. 1–5.

[4] C. Wu, R. Sehab, A. Akrad, and C. Morel, "Fault diagnosis methods and fault tolerant control strategies for the electric vehicle powertrains," *Energies*, vol. 15, no. 13, p. 4840, Jul. 2022.

[5] C. Sankavaram, B. Pattipati, K. R. Pattipati, Y. Zhang, and M. Howell, "Fault diagnosis in hybrid electric vehicle regenerative braking system," *IEEE Access*, vol. 2, pp. 1225–1239, 2014.

[6] H. Vidhya and S. Allirani, "A literature review on electric vehicles: Architecture, electrical machines for power train, converter topologies and control techniques," in *Proc. Int. Conf. Comput. Perform. Eval. (ComPE)*, Dec. 2021, pp. 565–575.

[7] A. Ahmadian, B. Mohammadi-Ivatloo, and A. Elkamel, "A review on plug-in electric vehicles: Introduction, current status, and load modeling techniques," *J. Modern Power Syst. Clean Energy*, vol. 8, no. 3, pp. 412–425, May 2020.

[8] S. G. Selvakumar, "Electric and hybrid vehicles—A comprehensive overview," in *Proc. IEEE 2nd Int. Conf. Electr. Power Energy Syst. (ICEPES)*, Dec. 2021, pp. 1–6.

[9] V. V. Jadhav, R. S. Shendge, P. B. Warule, and K. S. Wani, "E-drive system selection criterion for EV commercial and passenger vehicles segments," SAE Tech. Paper 2024-26-0253, 2024.

[10] M. I. Pathan, M. S. Shahriar, M. M. Rahman, M. S. Hossain, N. Awatif, and M. Shafiullah, "Comparative analysis of machine learning approaches in enhancing power system stability," in *Artificial Intelligence-based Smart Power Systems*. Hoboken, NJ, USA: Wiley, 2023, pp. 157–177.

[11] M. I. H. Pathan, M. J. Rana, M. S. Shahriar, M. Shafiullah, M. H. Zahir, and A. Ali, "Real-time LFO damping enhancement in electric networks employing PSO optimized ANFIS," *Inventions*, vol. 5, no. 4, p. 61, Dec. 2020.

[12] M. S. Shahriar, M. Shafiullah, M. I. H. Pathan, Y. A. Sha'aban, H. R. E. H. Bouchekara, M. A. M. Ramli, and M. M. Rahman, "Stability improvement of the PSS-connected power system network with ensemble machine learning tool," *Energy Rep.*, vol. 8, pp. 11122–11138, Nov. 2022.

[13] P. Ilius, M. Almuhaini, M. Javaid, and M. Abido, "A machine learning-based approach for fault detection in power systems," *Eng. Technol. Appl. Sci. Res.*, vol. 13, no. 4, pp. 11216–11221, 2023.

[14] M. Shafiullah, M. I. H. Pathan, M. S. Shahriar, A. Ali, M. I. Hossain, and M. S. Alam, "Real-time solution of PSS parameter tuning by GA-ANFIS in stabilizing the electrical power system," *Arabian J. Sci. Eng.*, vol. 48, no. 5, pp. 6925–6938, May 2023.

[15] H. Henao, G.-A. Capolino, M. Fernandez-Cabanas, F. Filippetti, C. Bruzzese, E. Strangas, R. Pusca, J. Estima, M. Riera-Guasp, and S. Hedayati-Kia, "Trends in fault diagnosis for electrical machines: A review of diagnostic techniques," *IEEE Ind. Electron. Mag.*, vol. 8, no. 2, pp. 31–42, Jun. 2014.

[16] W. Lang, Y. Hu, C. Gong, X. Zhang, H. Xu, and J. Deng, "Artificial intelligence-based technique for fault detection and diagnosis of EV motors: A review," *IEEE Trans. Transport. Electrific.*, vol. 8, no. 1, pp. 384–406, Mar. 2022.

[17] S. S. S. R. S. Duvvuri, "Rotor inter-turn short-circuit fault diagnostics (don't short) relevant modeling of slip-ring induction motor," in *Proc. IEEE 8th Power India Int. Conf. (PIICON)*, Dec. 2018, pp. 1–6.

[18] K. Pattipati, A. Kodali, J. Luo, K. Choi, S. Singh, C. Sankavaram, S. Mandal, W. Donat, S. M. Namburu, S. Chigusa, and L. Qiao, "An integrated diagnostic process for automotive systems," in *Computational Intelligence in Automotive Applications*, vol. 132. Berlin, Germany: Springer, 2008, pp. 191–218, doi: 10.1007/978-3-540-79257-4_11.

[19] C. Sankavaram, B. Pattipati, A. Kodali, K. Pattipati, M. Azam, S. Kumar, and M. Pecht, "Model-based and data-driven prognosis of automotive and electronic systems," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, Aug. 2009, pp. 96–101.

[20] J. Luo, H. Tu, K. Pattipati, L. Qiao, and S. Chigusa, "Diagnosis knowledge representation and inference," *IEEE Instrum. Meas. Mag.*, vol. 9, no. 4, pp. 45–52, Jul. 2006.

[21] C. Sankavaram, A. Kodali, D. F. M. Ayala, K. Pattipati, S. Singh, and P. Bandyopadhyay, "Event-driven data mining techniques for automotive fault diagnosis," in *Proc. Annu. Conf. PHM Soc.*, vol. 2, 2010, pp. 1–8.

[22] K. Sun, S. Likhate, V. Vittal, V. S. Kolluri, and S. Mandal, "An online dynamic security assessment scheme using phasor measurements and decision trees," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1935–1943, Nov. 2007.

[23] F. Immovilli, C. Bianchini, M. Cocconcelli, A. Bellini, and R. Rubini, "Bearing fault model for induction motor with externally induced vibration," *IEEE Trans. Ind. Electron.*, vol. 60, no. 8, pp. 3408–3418, Aug. 2013.

[24] T. W. S. Chow and H.-Z. Tan, "HOS-based nonparametric and parametric methodologies for machine fault detection," *IEEE Trans. Ind. Electron.*, vol. 47, no. 5, pp. 1051–1059, 2000.

[25] Z. Wang and C. S. Chang, "Online fault detection of induction motors using frequency domain independent components analysis," in *Proc. IEEE Int. Symp. Ind. Electron.*, Jun. 2011, pp. 2132–2137.

[26] Z. Wang, C. S. Chang, and Y. Zhang, "A feature based frequency domain analysis algorithm for fault detection of induction motors," in *Proc. 6th IEEE Conf. Ind. Electron. Appl.*, Jun. 2011, pp. 27–32.

[27] E. H. El Bouchikhi, V. Choqueuse, and M. Benbouzid, "Induction machine faults detection using stator current parametric spectral estimation," *Mech. Syst. Signal Process.*, vols. 52–53, pp. 447–464, Feb. 2015.

[28] A. Soualhi, G. Clerc, and H. Razik, "Detection and diagnosis of faults in induction motor using an improved artificial ant clustering technique," *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 4053–4062, Sep. 2013.

[29] X. Wang, S. Lu, and S. Zhang, "Rotating angle estimation for hybrid stepper motors with application to bearing fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 8, pp. 5556–5568, Aug. 2020.

[30] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors*, vol. 17, no. 2, p. 425, Feb. 2017.

[31] A. Shenfield and M. Howarth, "A novel deep learning model for the detection and identification of rolling element-bearing faults," *Sensors*, vol. 20, no. 18, p. 5112, 2020.

[32] W. Zhang, C. Li, G. Peng, Y. Chen, and Z. Zhang, "A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load," *Mech. Syst. signal Process.*, vol. 100, pp. 439–453, Feb. 2018.

[33] W. Mao, W. Feng, Y. Liu, D. Zhang, and X. Liang, "A new deep auto-encoder method with fusing discriminant information for bearing fault diagnosis," *Mech. Syst. Signal Process.*, vol. 150, Mar. 2021, Art. no. 107233.

[34] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," *Mech. Syst. Signal Process.*, vols. 72–73, pp. 303–315, May 2016.

[35] B.-S. Yang, M.-S. Oh, and A. C. C. Tan, "Fault diagnosis of induction motor based on decision trees and adaptive neuro-fuzzy inference," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 1840–1849, 2009.

[36] J. Zarei, H. Hassani, Z. Wei, and H. R. Karimi, "Broken rotor bars detection via Park's vector approach based on ANFIS," in *Proc. IEEE 23rd Int. Symp. Ind. Electron. (ISIE)*, Jun. 2014, pp. 2422–2426.

[37] H. Abu-Rub, S. M. Ahmed, A. Iqbal, M. Rahimian, and H. A. Toliyat, "Incipient bearing fault diagnostics for inverter fed induction motor drive using ANFIS," in *Proc. XIX Int. Conf. Electr. Mach. - ICEM*, Sep. 2010, pp. 1–5.

[38] A. Abid, M. T. Khan, and M. S. Khan, "Multidomain features-based GA optimized artificial immune system for bearing fault detection," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 348–359, Jan. 2020.

[39] H. Kaplan, K. Tehrani, and M. Jamshidi, "A fault diagnosis design based on deep learning approach for electric vehicle applications," *Energies*, vol. 14, no. 20, p. 6599, Oct. 2021.

[40] K. Vinothini, K. S. Harshavardhan, J. Amerthan, and M. Harish, "Fault detection of electric vehicle using machine learning algorithm," in *Proc. 3rd Int. Conf. Electron. Sustain. Commun. Syst. (ICESC)*, Aug. 2022, pp. 878–881.

[41] S. Nangrani and I. S. Nangrani, "Artificial intelligence based state of charge estimation of electric vehicle battery," in *Proc. Int. Conf. Smart Technol. Energy, Environ., Sustain. Develop.* Singapore: Springer, 2020, pp. 679–686, doi: 10.1007/978-981-16-6879-1_65.

[42] N. Gan, Z. Sun, Z. Zhang, S. Xu, P. Liu, and Z. Qin, "Data-driven fault diagnosis of lithium-ion battery overdischarge in electric vehicles," *IEEE Trans. Power Electron.*, vol. 37, no. 4, pp. 4575–4588, Apr. 2022.

[43] Y. Zhao, P. Liu, Z. Wang, L. Zhang, and J. Hong, "Fault and defect diagnosis of battery for electric vehicles based on big data analysis methods," *Appl. Energy*, vol. 207, pp. 354–362, Dec. 2017.

[44] Y. Li, S. Zhang, L. Liang, and Q. Ding, "Multivariate multiscale Higuchi fractal dimension and its application to mechanical signals," *Fractal Fractional*, vol. 8, no. 1, p. 56, Jan. 2024.

[45] Y. Li, B. Tang, S. Jiao, and Q. Su, "Snake optimization-based variable-step multiscale single threshold slope entropy for complexity analysis of signals," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.

[46] A. Azhar, N. M. Ariff, M. A. A. Bakar, and A. Roslan, "Classification of driver injury severity for accidents involving heavy vehicles with decision tree and random forest," *Sustainability*, vol. 14, no. 7, p. 4101, Mar. 2022.

[47] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*, Y. Lechevallier and G. Saporta, Eds. Berlin, Germany: Springer, 2010, doi: 10.1007/978-3-7908-2604-3_16.

[48] Y. Wang, W. Deng, and G. Lin, "Bayesian sparse learning with preconditioned stochastic gradient MCMC and its applications," *J. Comput. Phys.*, vol. 432, May 2021, Art. no. 110134.

[49] Y. Freund, "Boosting a weak learning algorithm by majority," *Inf. Comput.*, vol. 121, no. 2, pp. 256–285, Sep. 1995.

[50] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[51] R. E. Schapire, "The boosting approach to machine learning: An overview," in *Nonlinear Estimation and Classification* (Lecture Notes in Statistics), vol. 171. New York, NY, USA: Springer, 2003, pp. 149–171, doi: 10.1007/978-0-387-21579-2_9.

[52] E. N. de Souza and S. Matwin, "Improvements to Adaboost dynamic," in *Proc. 25th Can. Conf. Artif. Intell.*, Toronto, ON, Canada. Berlin, Germany: Springer, 2012, pp. 293–298, doi: 10.1007/978-3-642-30353-1_26.

[53] M. Nalluri, M. Pentela, and N. R. Eluri, "A scalable tree boosting system: XG boost," *Int. J. Res. Stud. Sci. Eng. Technol.*, vol. 7, no. 12, pp. 36–51, 2020.

[54] X. Li, C. Liu, Z. Mao, M. Xiao, L. Wang, S. Qi, and F. Zhou, "Predictive values of neutrophil-to-lymphocyte ratio on disease severity and mortality in COVID-19 patients: A systematic review and meta-analysis," *Crit. Care*, vol. 24, no. 1, pp. 1–10, Dec. 2020.

[55] M. S. Alam, F. S. Al-Ismail, M. S. Hossain, and S. M. Rahman, "Ensemble machine-learning models for accurate prediction of solar irradiation in Bangladesh," *Processes*, vol. 11, no. 3, p. 908, Mar. 2023.

[56] S. Dhaliwal, A.-A. Nahid, and R. Abbas, "Effective intrusion detection system using XGBoost," *Information*, vol. 9, no. 7, p. 149, Jun. 2018.

[57] L. Wehenkel, M. Pavella, E. Euxibie, and B. Heilbronn, "Decision tree based transient stability method a case study," *IEEE Trans. Power Syst.*, vol. 9, no. 1, pp. 459–469, 1994.

[58] S. Zhang, "Cost-sensitive KNN classification," *Neurocomputing*, vol. 391, pp. 234–242, May 2020.

[59] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1774–1785, May 2018.

[60] P. Wang, Y. Zhang, and W. Jiang, "Application of K-nearest neighbor (KNN) algorithm for human action recognition," in *Proc. IEEE 4th Adv. Inf. Manage., Communicates, Electron. Autom. Control Conf. (IMCEC)*, vol. 4, Jun. 2021, pp. 492–496.

[61] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," in *Proc. Int. Conf. Intell. Comput. Control Syst. (ICCS)*, May 2019, pp. 1255–1260.

[62] M. Z. Ali, M. N. S. K. Shabbir, S. M. K. Zaman, and X. Liang, "Single- and multi-fault diagnosis using machine learning for variable frequency drive-fed induction motors," *IEEE Trans. Ind. Appl.*, vol. 56, no. 3, pp. 2324–2337, May 2020.

[63] E. Principi, D. Rossetti, S. Squartini, and F. Piazza, "Unsupervised electric motor fault detection by using deep autoencoders," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 2, pp. 441–451, Mar. 2019.

[64] R. N. Toma, A. E. Prosvirin, and J.-M. Kim, "Bearing fault diagnosis of induction motors using a genetic algorithm and machine learning classifiers," *Sensors*, vol. 20, no. 7, p. 1884, Mar. 2020.

MAHBUB UL ISLAM KHAN received the B.Sc. (Eng.) degree in electrical and electronic engineering (EEE) from Hajee Mohammad Danesh Science and Technology University, Dinajpur, Bangladesh, in 2022, where he is currently pursuing the M.Sc. degree in master mathematics for data and resource sciences (mathematical data science). He is a former Project Officer with Bangladesh Open Source Network, Dhaka, Bangladesh. His research interests include power and energy systems, electronics, microsystems, automation, artificial engineering, and machine learning.

MD. ILIUS HASAN PATHAN received the B.Sc. degree in electrical and electronic engineering from Rajshahi University of Engineering and Technology (RUET), Rajshahi, Bangladesh, in 2011, and the M.Sc. degree in electrical engineering from the King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 2018. He is currently an Assistant Professor with the Department of Electrical and Electronic Engineering, Hajee Mohammad Danesh Science and Technology University, Dinajpur, Bangladesh. His current research interests include but are not limited to the integration of renewable energy and EVs in smart grids and their wide distance monitoring, controlling, solving protection and security issues, implementation of artificial intelligence, machine learning, and optimization techniques in smart grids, power industry and machine drives, and power quality.

MOHAMMAD MOMINUR RAHMAN received the B.Sc. degree in electrical and electronic engineering from the Ahsanullah University of Science and Technology, Dhaka, Bangladesh, in 2010, and the master's degree from the Electrical Engineering Department, King Saud University, Saudi Arabia, in 2020. He is currently pursuing the Ph.D. degree in computer science and engineering with Hamad Bin Khalifa University, Doha, Qatar. His research interest includes electric vehicle grid integration with renewable energy. Besides that, he has also worked on nanomaterials like phase change material and cadmium selenium quantum dots solar cells. His research will mainly cover artificial intelligence and machine learning techniques in the big data platform.

MD. MAIDUL ISLAM received the B.Sc. degree in electrical and electronic engineering from the Green University of Bangladesh, Dhaka, Bangladesh, in 2019. His research interests include renewable energy, solar cells, wind energy, power system operation and simulation, renewable energy integration and control, microgrids, smart grids, and energy storage systems.

MOHAMMED ARFAT RAIHAN CHOWDHURY received the M.Sc. degree in logistics and supply chain management from Hamad Bin Khalifa University, Doha, Qatar, in 2023. He is currently a Research Assistant. His research interests include optimization, sustainability, data science, and logistics management.

MD. SHAMIM ANOWER received the Ph.D. degree in electrical engineering from The University of New South Wales, Australia, in 2012. His field of specialization is underwater acoustic signal processing, power system stability, biomedical engineering, cyber security, and wireless communication. He has over 100 refereed publications, including journals (16 Q1 ranked in Scimago) and conference papers with an H-index of 18, i10-index of 29, and 954 citations. These cover the target sectors: energy and mining technology, quantum information, advanced digital, data science and ICT, and cyber security.

MD. MASUD RANA received the Ph.D. degree in electrical engineering from the University of Technology Sydney (UTS), in 2013. His research interests include antenna and propagation, electromagnetic fields, and communication engineering.

MD. SHAFIUL ALAM received the Ph.D. degree in electrical engineering from the King Fahd University of Petroleum and Minerals (KFUPM), in 2018. His research interests include renewable energy, ac/dc microgrids, high-voltage dc transmission systems, optimization, and machine learning for power systems.

MAHMUDUL HASAN (Graduate Student Member, IEEE) received the B.Sc. (Eng.) degree in computer science and engineering (CSE) from Hajee Mohammad Danesh Science and Technology University, Dinajpur, Bangladesh, in 2021, and the M.Sc. (Eng.) degree in CSE from the Department of CSE, Hajee Mohammad Danesh Science and Technology University, in 2023, where he is currently pursuing the Ph.D. degree in information technology (IT). He is a former Lecturer with the Department of CSE, University of Creative Technology, Chittagong, Bangladesh. He is one of the instructors of the first research BootCamp in Bangladesh. His research interests include federated learning, machine learning, deep learning, cyber security, health informatics, business intelligence, and computational sociology.

MD. SHOHANUR ISLAM SOBUJ received the B.Sc. degree in electrical and electronic engineering from Hajee Mohammad Danesh Science and Technology University, Bangladesh, in 2022. His research interests include AI, NLP, and computer vision. He has professional experience as a Machine Learning Engineer, specializing in developing AI models and algorithms. Beyond academia, he is passionate about applying machine learning techniques to solve real-world problems.

MD. BABUL ISLAM (Graduate Student Member, IEEE) received the bachelor's degree in computer science in Bangladesh and the master's degree in information engineering in China. He is currently pursuing the Ph.D. degree with the Department of Computer, Modeling, Electronic, and System Engineering (DIMES), University of Calabria (UNICAL), Rende, Italy. He has over a year of teaching experience in Bangladesh and one and a half years of industrial experience in data analytics in China. As a first author, he has published several articles. His research interests include applied machine learning, deep learning, the IoT energy efficiency, and AI healthcare.

VEERPRATAP MEENA (Member, IEEE) received the B.Tech. degree in electrical engineering from the Government Engineering College, Bikaner, India, in 2018, the M.Tech. degree in power system engineering from the Indian Institute of Technology Roorkee (IIT Roorkee), in 2020, and the Ph.D. degree in electrical engineering from the Malaviya National Institute of Technology (MNIT), Jaipur, India, in 2023. Currently, he is a Postdoctoral Fellow with Amrita Vishwa Vidyapeetham, Bengaluru, India. His research interests include electric vehicles, power systems, system modeling, controller design, interval systems, model order reduction, applications of optimization algorithms, and power electronics. He contributed to the body of knowledge by publishing several research papers in international journals and conferences. Also, he has served as the general chair, the program chair, and the program committee member of several international conferences and workshops. He has been the Chair of the IEEE Systems Council Systems Education Technical Committee, USA, since 2023. He is an Associate Editor/an Editorial Board Member of refereed journals, in particular IEEE ACCESS, *International Journal of System Dynamics Applications* (IGI Global), *International Journal of Automation and Control*, and *International Journal of Intelligent Engineering Informatics* (Inderscience Publishers).

FRANCESCO BENEDETTO (Senior Member, IEEE) has been the Chair of the IEEE Vehicular Technology Society (VTS) Italian Chapter and has been the Chair of the IEEE 1900.1 Standard Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management, since 2016. He has been the Leader of the WP 3.5 on the Development of Advanced GPR Data Processing Technique of the European COST Action TU1208 Civil Engineering Applications of Ground Penetrating Radar. He is an Associate Editor of IEEE ACCESS, an Editor of IEEE SDN Newsletter, an Associate Editor of the *AEÜ—International Journal of Electronics and Communications* (Elsevier), the Editor-in-Chief of the international journal *Recent Advances on Computer Science and Communications* (Bentham), the General Co-Chair of the IEEE 43rd International Conference on Telecommunications and Signal Processing (TSP 2020), the General Chair of the Series of International Workshops on Signal Processing for Secure Communications (SP4SC 2014, 2015, and 2016), and the Lead Guest Editor of the Special Issue on Advanced Ground Penetrating Radar Signal Processing Techniques for the *Signal Processing* Journal (Elsevier). He also served as a reviewer for several IEEE TRANSACTIONS, IET proceedings, EURASIP, and Elsevier journals, and a TPC member for several IEEE international conferences and symposia in the same fields.

Open Access funding provided by 'Università degli Studi Roma Tre' within the CRUI CARE Agreement