

Received 10 November 2023, accepted 30 November 2023, date of publication 7 December 2023,
date of current version 15 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3340865

METHODS

Intelligent Identification of Simultaneous Faults of Automotive Software Systems Under Noisy and Imbalanced Data Using Ensemble LSTM and Random Forest

MOHAMMAD ABOUSH¹, CHRISTOPH KNIEKE¹, AND ANDREAS RAUSCH

Institute for Software and Systems Engineering, Clausthal University of Technology, 38678 Clausthal-Zellerfeld, Germany

Corresponding author: Mohammad Abboush (mohammad.abboush@tu-clausthal.de)

This work was supported by the Open Access Publishing Fund of Clausthal University of Technology.

ABSTRACT According to ISO 26262 standard, functional validation of the developed Automotive Software Systems (ASSs) is crucial to ensure the safety and reliability aspects. Hardware-in-the-loop (HIL) has been introduced as a reliable, safe and flexible test platform to enable the validation process in real-time. However, the traditional failure analysis process of HIL tests is time-consuming, extremely difficult and requires considerable effort. Therefore, an intelligent solution that can overcome the above challenges is required. Following a data-driven approach, the development of deep learning methods for fault detection and classification has gradually become a hot topic. However, despite the fruitful results, most of the previous studies were conducted for single faults without considering the simultaneous occurrence of multiple faults and ignoring the noisy conditions. In this study, based on multi-label ensemble long short term memory (LSTM) and random forest (RF) techniques, a novel method for simultaneous fault classification under noisy conditions is developed. To improve the robustness of the model against noise, a GRU-based denoising autoencoder (DAE) was implemented. Furthermore, to overcome the challenge of imbalanced data, a random undersampling algorithm was employed. By doing so, the single and simultaneous sensor faults occurring during HIL testing of ASSs can be efficiently and automatically detected and identified. To evaluate the capabilities and robustness of the proposed method, a high-fidelity gasoline engine with a dynamic vehicle system and driving environment was used as a case study. The analysis results demonstrate that the proposed model can achieve a high degree of accuracy under noise with an average detection accuracy of 99.43%. Moreover, compared to the individual methods, the proposed ensemble learning architecture with DAE provides more promising fault identification performance with improved accuracy and robustness. Specifically, the test results show that the proposed model is superior to other state-of-the-art models in identifying simultaneous faults with 91.2% F1-Score.

INDEX TERMS Automotive software systems, fault detection and diagnosis, deep learning, denoising autoencoder, LSTM, random forest, real-time simulation, fault injection, hardware-in-the-loop (HIL).

I. INTRODUCTION

IN the automotive industry, as a means of reducing the risk of vehicle accidents, considerable efforts have been devoted to the development of advanced active safety systems. Besides, Advanced Driver Assistance Systems

The associate editor coordinating the review of this manuscript and approving it for publication was Guillermo Valencia-Palomo¹.

(ADAS) have played a vital role not only in supporting driving comfort but also in improving road safety, especially in emergencies [1]. On the other hand, Verification and Validation (V&V) of such systems with a high degree of complexity and functional dependencies is a challenge [2].

As part of quality assurance, and to meet the development requirements of the functional safety standard ISO 26262, comprehensive testing activities should be performed to

ensure safety and reliability characteristics [3]. To this end, according to the V-model development approach, several test phases are defined, which are known as X-in-the-loop [4]. Some major “in-the-loop” methods of the V-Model are Model-in-the-Loop (MIL), Software-in-the-Loop (SIL), Processor-in-the-Loop (PIL), Hardware-in-the-Loop (HIL), Vehicle-in-the-Loop (VIL) and real test drives [5], [6].

The deviation in the components’ behavior from the normal state, which leads to the failure of an element, is known as a fault [7]. In automotive systems, due to the environmental and working conditions, the hardware components, i.e., sensors and actuators, are invariably prone to faults. Besides, ECUs, functional specifications, gateways, networks, vehicle subsystems, power supply and data acquisition systems are also potential points of faults in a vehicle network [8]. In the literature, the types of sequential data-related faults are termed data-centric and system-centric faults [9]. Gain, offset/bias, noise, hard-over, spike, stuck-at, packet loss, delay and drift faults are some examples of sensor faults [10]. It should be noted that the fault occurrence can be either permanent or transient, single or simultaneous faults over a specific period.

To detect unexpected faults at the system level, digital test drives with HIL simulation are introduced to serve as a validation platform for the ECU performance. In this manner, the limitations of real test drives on public roads in terms of cost, time and risk can be overcome [11], [12]. The sequential data recorded during the test execution represents the behavior of the system under test. Conventionally, to detect the unnoticed and minor faults that lead to undesired behaviors, manual inspections of the recordings are performed based on expert knowledge. Due to the heterogeneous components in the complex system architecture of the vehicles, however, vast amounts of multivariate time series data are recorded. Consequently, conventional failure analysis of test records becomes time-consuming, extremely difficult, and requires considerable efforts [13], [14]. Therefore, an intelligent solution that can overcome the above problems is required.

Current state-of-the-art discriminates between four different approaches for performing Fault Detection and Diagnosis (FDD) tasks on sequential data, namely model-based methods [15], signal-based methods [16], knowledge-based methods [17], and data-driven methods [18]. Although the model-based approach is efficient and robust under dynamic conditions, an accurate mathematical model is required, which adds more complications as the complexity of the system increases [19]. Besides the required effort, the demand for expert knowledge with extensive human intervention is a barrier to the development of knowledge-based methods [20]. Similarly, a deep understanding of the fault-free symptoms of the system is required for the development of FDD based on signal analysis methods [21].

In recent years, thanks to the introduction of advanced smart sensors and data acquisition technologies, contributing

to the provision of large amounts of data, FDD-based data-driven approach has been widely used in various fields. The main steps to develop the target model are data acquisition, feature extraction, and feature learning. As a category of data-driven approach, Machine Learning (ML) methods, e.g., Support Vector Machine (SVM) and k-Nearest Neighbor (KNN), have gained importance in recent years. However, the difficulty of manually extracting representative features of the faults is considered a drawback, especially in the presence of a large amount of data. Therefore, Deep Learning (DL)-based methods with automatic feature extraction have been extensively explored and successfully used for various applications. Moreover, the DL approach is capable of automatically learning the extracted features and establishing a nonlinear relationship between the fault symptoms and the corresponding classes. Consequently, the development of FDD model based on DL methods has gradually become a hot topic. Based on neural network architecture, several models have been proposed to perform FDD in the past decade. Deep Belief Network (DBN), Restricted Boltzmann Machine (RBM), Convolutional Neural Network (CNN), Recurrent Neural Networks (RNNs) and Autoencoders (AE) are commonly applied architectures [22], [23]. Notably, FDD-based hybrid DL methods have attracted much attention from research teams due to their great successes compared to stand-alone models [24]. Besides, several DL architectures have been proposed for time series anomaly detection and denoising the data [25], [26].

However, despite the fruitful results of the existing researches, most of the studies have been conducted for a single fault without considering the concurrent occurrence of multiple faults [27], [28]. Besides, in real-world industrial applications, non-typical dataset, i.e., imbalanced and noisy data, is considered another complicating factor [29], [30]. To fill this gap in the literature, this study proposes a novel method for simultaneous FDD, i.e., fault identification, considering the aforementioned issues. In this study, the applicability of the ensemble learning-based classifier for FDD under imbalanced and noisy conditions was investigated. The proposed model is able to automatically analyze the test records from HIL, which improves the real-time validation of ASSs during the V-cycle development process. To the best of our knowledge, this is the first study to address the problem of detecting and identifying the concurrent faults that occur during the HIL test of ASSs considering the noisy and imbalanced data. The contributions of this study can be summarized as follows:

- A novel, effective, and robust ensemble learning-based simultaneous Fault Detection and Identification (FDI) method is proposed. Specifically, a multi-label ensemble of LSTM and RF-based classifier is developed.
- To improve the performance and robustness of the developed FDI model against different levels of noise, a novel framework based on GRU-based DAE developed in the previous study [31] is adapted.

- To overcome the challenge of FDI model development in the presence of imbalanced datasets, the random undersampling algorithm is applied. Consequently, the classification accuracy of the minority classes is improved.
- To capture a realistic system behaviour in the presence of single and simultaneous sensor faults, real-time Fault Injection (FI) based on HIL simulation and a high-fidelity automotive system model is employed.
- The effectiveness of the proposed method was evaluated using real-time automotive simulation data under various noise levels, and the results were also compared to stand-alone methods.

The rest of the article is structured as follows. Section II provides an overview of the relevant literature. Section III describes the main phases of the proposed method. The dataset, implementation steps and the case study used are presented in Section IV. The results of the experimental evaluation are analyzed and discussed in Section V. Finally, Section VI outlines the conclusion and the future work.

II. RELATED WORK

This section describes related work in the field of FDD, focusing on the main contributions and drawbacks in the automotive domain considering both single and concurrent faults.

A. FAULT DETECTION AND DIAGNOSIS IN AUTOMOTIVE DOMAIN

In the last decade, along with the advances in the development of ASSs architectures, FDD strategies have attracted considerable attention from researchers. Hence, various methods for fault detection, isolation, identification and sensor state prediction have been proposed in the automotive domain.

Focusing on internal combustion engines, Jiménez et al. [32] proposed a scheme aimed at detecting and isolating a faulty fuel injector. Using an FPGA, the developed ANN-based FDD scheme has been validated in real-time. The results exhibited a remarkable performance in the classification tasks with an accuracy close to 100%. In the same field and using a HIL platform for real-time simulation, a machine learning-based system for fault detection and fault-tolerant control was proposed in [33]. The conducted comparative study of six multi-class ML models shows that the Random Forest model outperforms the other models in terms of fault identification in an air brake system with an accuracy of 91.99%. FDD for Electric Vehicles (EVs) based on DL techniques has been another hot topic in the last year. For example, in [34], LSTM-based single fault diagnosis was proposed for the induction motor of an EV. Based on the simulation model of an EV system in MATLAB/Simulink, the training dataset, including faults, has been generated by injecting short-circuit and open-circuit faults. The validation result of the proposed system using an EV prototype demonstrates the superiority of LSTM in terms of accuracy over other techniques. Meanwhile,

to address the problem of unknown fault classes and the collection of representative training data, a data-driven fault classification algorithm has been proposed in [35]. In the mentioned work, a Weibull-calibrated OSVM classifier combined with Bayesian filtering has been developed to cover seven different single types of engine faults. As a case study, a real internal combustion engine has been used to demonstrate the classification performance of unknown faults on sequential data. However, to generate the residual in the proposed work, a high-precision mathematical model is required, which in turn increases the cost and the complexity. In the context of developing FDD for autonomous vehicles, Biddle et al. have proved in [36] that the employment of SVM from ML techniques can ensure high accuracy in detection, isolation and identification with an efficient computational burden for multiple faults in multi sensors. To evaluate the proposed algorithm, the MATLAB/IPG CarMaker co-simulation platform was used considering five individual sensor faults, namely drift, hard-over, erratic, spike and stuck fault. However, the real-time constraints of the system behaviour in the presence of the faults remained unaccounted for dataset generation. Furthermore, the robustness of the developed model against sensor noise was not analyzed, which opens the door for further improvements.

The modification of conventional DL architecture shows better fault detection performance in various fields [37]. For example, to ensure the safety of a railway vehicle system, a DL-based fault detection method was proposed in [38]. In the proposed work, the bidirectional LSTM-DAE network was modified to overcome the challenge of the unavailability of data sets under a faulty state, which is necessitated to determine the added noise level. Although the performance of the modified BiLSTM is superior over other models, i.e., autoregression model, LSTM and BiLSTM-DAE, the system state under concurrent faults was not considered.

Despite of the rapid development of real-time testing platforms, research on developing an intelligent system able to detect, isolate and identify faults during the development process is still in the early stages. For example, as an improvement of the embedded system testing process, Scharoba et al. [39] have proposed a proximity-based anomaly detection system using ML techniques to automatically evaluate the test runs and identify the faulty behaviour. The method has been developed based on the historical test records so that deviations from the normal behaviour of the test object can be detected. In the aforementioned study, a drive controller under development was used as a case study to evaluate the proposed framework. Despite the obvious superiority of the proposed anomaly detection method, the identification of single and simultaneous faults has not been considered in this work. Concerning the same area, the problem of classifying sensor faults during the V-cycle development process was researched by Abboush et al. [40]. They proposed a novel DL architecture for identifying system-level fault types using a combined CNN and LSTM network.

The generation of a faulty dataset was achieved by injecting the faults in real time using a HIL platform. To validate the performance of the proposed model, a gasoline engine with entire vehicle dynamic models has been employed. The evaluation results exhibit high performance of the hybrid DL techniques compared to the stand-alone methods with an accuracy of 98.88% and 98.85%, respectively.

In line with the above observations from previous works, the developed intelligent methods have covered various systems in the automotive domain. However, despite the proposed models' high performance in terms of accuracy, the diagnosis problem of the simultaneous faults occurrence under noisy and imbalanced data conditions is not addressed. Therefore, in our proposed work, we attempt to bridge this gap in the literature by proposing a novel method capable of detecting and classifying concurrent faults for real-time testing of an ASS during the V-cycle development process.

B. DETECTION AND DIAGNOSIS OF CONCURRENT FAULTS IN AUTOMOTIVE AND OTHER DOMAINS

The availability of the datasets containing the faulty behaviour paved the way for using a data-driven approach to address the issue of the simultaneous fault diagnosis. For example in [41], Asgari et al. have proposed a hybrid FDD framework focusing on the cooling systems in data centres as a target application. In the proposed strategy, one-class SVM and Nonlinear AutoRegressive Exogenous (NARX) have been used for the detection phase, while two DL techniques, i.e., 2D-CNN and LSTM, have been employed for the diagnosis task. Seven different types of faults related to pumps and fans along with their combinations have been considered. Besides, the effect of adding noise to the training dataset has been analysed with different standard deviations, i.e., 0.1, 0.5, 2 and 3. Based on the F1-Score and accuracy as evaluation metrics, the experiments show the robustness and the ability of the proposed model, with 100% accuracy in detection and diagnosis. However, the limitations of one-class SVM in terms of runtime computation impose restrictions on its application in real-time. In addition, the simulation model of the target system has been used to simulate the system behaviour under normal and faulty conditions without considering the real-time constraints. Addressing the same challenge, Li et al. [42] have shown in their proposed study that multi-label classification based on DL techniques can provide significant results for FDD of solid oxide fuel cell (SOFC) systems. The significance of the work lies in the fact that the simultaneous faults are not required, only faulty data with individual faults. For feature extraction, PCA techniques have been used. Furthermore, multi-class SVM techniques have been employed for the classification of nine different fault classes. However, despite the high performance of single fault classification with F1-Score of 96.4%, the validation results for concurrent faults are considered not satisfying with F1-Score of 84.93%. Besides the high computational time required for the classification, i.e., 26.1 seconds, the dataset has been generated in a simulation environment

without accounting for real-time constraints. Therefore, the applicability of the proposed model in real-time application should be further investigated. In the automotive domain, one of the first examples of addressing the diagnosis problem of simultaneous-engine faults using a probabilistic committee machine was presented in [43]. The proposed intelligent FDD system exhibits good diagnostic performance with an accuracy of 92% and 81.49% for single and simultaneous faults, respectively. Considering a real 4-cylinder in-line engine as a case study, three different signal patterns of a real engine, including 15 types of faults, were used to train and validate the model. In the same context, but for a different application domain, fuzzy logic-based fault detection and isolation of multiple and unknown faults for a continuously-stirred tank heating system has been developed in [44]. Besides, wavelet transform techniques have been adopted to deal with the noisy data in the measurements. By doing so, the robustness of the developed model against noise during the diagnosis process of multiple faults has been ensured with a high accuracy of 100%. However, to develop such a system, expert knowledge with a deep understanding of the domain and physical behaviour is required, which is a challenge in complex software systems. Similarly, considering the noisy seismic data, a residual deep neural network coupled with the IIR Wiener filter denoising method is proposed in [45]. The proposed method shows high performance not only in denoising and reconstructing the data, but also in detecting the abnormal signals in the records. By doing so, less computational resources, effort and time are required for the detection and denoising process. However, other diagnosis tasks, i.e., fault identification and localization, have not been covered. Finally, an architecture for a health monitoring system considering multiple faults and sensors in autonomous vehicles has been proposed by Safavi et al. in [46]. To address the task of fault detection, isolation and identification, DNNs with multiple classes and 1D CNNs have been employed. Real sensor measurements have been used to validate the performance of the proposed methodology. In this study, four types of sensor faults have been considered, namely drift, hard-over, erratic and spike faults. Although the proposed system presents a good performance with a detection accuracy of 99.84% and an identification accuracy between 73% and 100%, the robustness of the model with respect to the noise has not been considered. Furthermore, a normal data distribution with one standard deviation was statically used to generate the faulty data. Table 1 outlines an overview of the related works highlighting the key aspects of the proposed work in comparison to other related works. Specifically, the table presents the approach used in the related works, the application domain, the dataset used for development, the target faults to be identified and the evaluation of the work in terms of performance and robustness.

To conclude, despite proposing novel FDD models with remarkable achievements, detecting and identifying the concurrent faults in the presence of imbalanced data and

TABLE 1. Overview of the related work.

Reference	Techniques	Application	Dataset	Real-Time Constraints	Fault	Review
[32]	ANNs	Isolating the damaged injector of an internal combustion engine	Real word dataset	yes	Four single faults	Accuracy: High 100% Implementation: Real vehicle Robustness: -
[33]	RF	Identifying the faults of air brake system	Real-time simulation dataset	yes	Two single faults	Accuracy: Acceptable 91.99 % Implementation: Real-time simulation Robustness: Considered
[34]	LSTM	FDD of induction motor of an EV	Simulation dataset	No	Four single faults	Accuracy: High 97% Implementation: EV prototype Robustness: -
[35]	OSVM with Bayesian filtering	Classifying the unknown faults of an internal combustion engine	Real engine dataset	Yes	Seven single faults	Accuracy: Low 85% Implementation: Engine test bench Robustness: -
[36]	SVM	Detecting, isolating, identifying and predicting the faults of autonomous vehicle	Simulation dataset	No	Five single faults	Accuracy: High 97.01% Implementation: Co-simulation MATLAB/IPG CarMaker Robustness: -
[40]	CNN-LSTM	Single FDC for HIL tests during ASSs development	Real-time simulation dataset	Yes	Nine single faults	Accuracy: High 98.8% Implementation: HIL test benches Robustness: -
[41]	OCSVM-NARX, 2D-CNN-RNN	FDD of cooling systems in data centers	Simulation dataset	No	Seven fault with its combinations	Accuracy: High 100% Implementation: Simulation model Robustness: Considered
[42]	PCA-multi-class SVM	Classifying the faults of solid oxide fuel cell (SOFC) systems	Simulation dataset	No	Nine faults with its combinations	Accuracy: Low 84.93% Implementation: Simulation model Robustness: -
[43]	PCM	Diagnosis simultaneous faults of engine	Real engine dataset	Yes	Fifteen faults with its combinations	Accuracy: Low 81.49% Implementation: Real vehicl Robustness: Considered
[44]	Fuzzy logic	FDI of continuously-stirred tank heater system	Simulation dataset	No	Three faults with its combination	Accuracy: High 100% Implementation: Simulation model Robustness: Considered
[46]	DNN and 1D CNNs	Detecting, isolating, identifying and predicting the faults of autonomous vehicle	Real word dataset	Yes	Four fault with its combinations	Accuracy: Acceptable 89% Implementation: Real vehicle Robustness: -
Proposed Model	DAE, LSTM-RF	FDI for HIL tests during the development process of ASSs	Real time simulation dataset	Yes	Five faults with its combinations	Accuracy: High 91.2% Implementation: HIL test benches Robustness: Considered

existing noise in the measurements has not been sufficiently explored. Moreover, the coverage of the fault types and the generation of the faulty data considering the real-time system behaviour under faulty conditions should be accounted for. The novelty of the proposed work is thus to fill the gap in the literature by developing a robust DL-based single and simultaneous FDI model that accounts for the noisy and imbalanced data of HIL tests. Furthermore, a real-time FI framework and a high-fidelity vehicle system were utilised to collect representative datasets for developing the target model.

III. METHODOLOGY

The proposed architecture aims at detecting and identifying single and simultaneous faults in the sensors of ASSs under noisy and imbalanced data conditions. The proposed model is intended to be used during the development phase of ASSs, i.e., real-time system validation using the HIL platform. By doing so, the failure analysis process, during the mentioned testing phase, can be improved. Consequently, the FDI of sensor faults is achieved in an efficient manner, reducing the time and effort during the analysis process. The proposed FDI architecture consists of four main phases, namely data acquisition, data preparation, data denoising and feature learning, as shown in Figure 1.

A. DATA ACQUISITION

To ensure the reality of the captured system behaviour under different conditions, a real-time HIL simulation platform

is employed. In this way, the interaction of the developed Electronic Control Unit (ECU) with other system components, e.g., other ECUs, real sensors and actuators, controlled equipment and the in-vehicle network, is accurately captured in real-time. Besides, many representative and relevant test kilometers can be performed with low costs and high safety compared to real test drives. Thanks to the logging system in the HIL platform, the target system's variables are recorded as multivariate time series data during the virtual test drive. In this study, the main elements of the HIL system are the HIL simulator, the developed target ECU, real wheel and pedals, CAN bus communication and the real-time FI framework.

As a result of system execution under non-faulty conditions, i.e., desired/standard behaviour, healthy data samples can be acquired. To this end, the sensor and actuator signals accessed via CAN bus are recorded. Owing to the HIL simulator with high-fidelity automotive simulation models, high-quality datasets can be collected considering the real-time constraints. Besides, employing the real-time FI framework proposed in a previous work [47], representative faulty and healthy data are generated in real-time. For this purpose, the target system is executed under faulty conditions, i.e., in the case of random sensor/actuator fault occurrence. Thanks to the aforementioned framework, the faults can be injected programmatically via the CAN bus without changing the original system architecture. Thus, back-box execution of both ECU and plant in real-time is ensured. Noteworthy, as a precondition of FI process, three attributes should be specified, namely FI time, fault types and

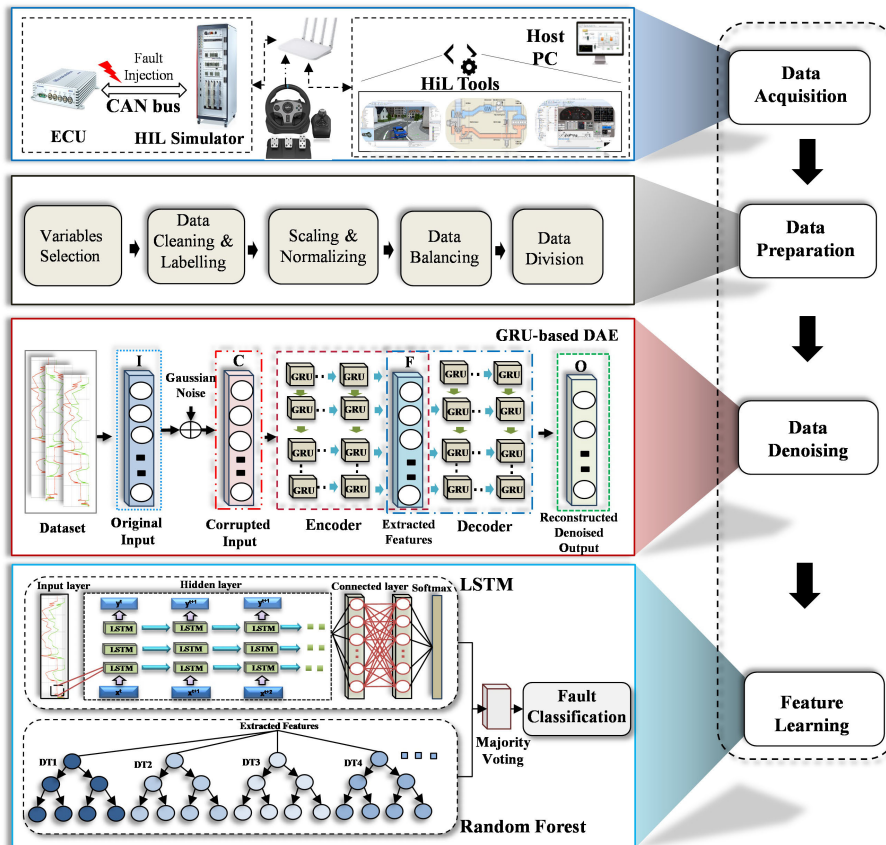


FIGURE 1. Proposed ensemble learning-based simultaneous FDI methodology.

fault locations. Based on the system architecture components, the potential location of the occurrence of the faults can be identified, e.g., sensor, actuator, network or controller [8]. Whereas various sequential data-related malfunctions, e.g., gain, offset, hard-over, stuck-at, delay, noise, packet loss, drift and spike fault can be injected as fault types [48]. The faults can be injected either permanently or temporarily, and may occur individually or simultaneously as multiple faults over a period. Therefore, the timing and duration of FI play a critical role in the generated representative faulty data. For example, transient faults produce imbalanced data with a different ratio between faulty to healthy samples. Notably, faulty data with simultaneous faults are generated by injecting two different faults simultaneously into different locations. Various factors potentially cause the occurrence of faults within or between the system components. Some examples of direct causes of faults are dirty or damaged sensors, aging, corrosion, vibration, electromagnetic interference, improper calibration, and weak batteries [49]. Notably, other factors, such as bumpy roads and driving uncertainties, can also cause anomalies without necessarily producing faults.

B. DATA PREPARATION

Once the representative dataset is acquired, the data is pre-processed in the data preparation phase. By mitigating

the irrelevant data and correcting the missing samples at this stage, the training process is improved by reducing the computational cost and avoiding overfitting [50]. Basically, the collected data is pre-processed through various steps, i.e., variable selection, data cleaning, data labeling, scaling and normalization, balancing and data division.

Since the entire vehicle system is simulated with a high-fidelity simulation model, a large number of system variables can be captured. Therefore, it is essential to select the variables that play a critical role in determining the state of the system, i.e., healthy or faulty state. In this study, various system-level sensor signals were selected to serve as input for the targeted FDI model. Engine speed, engine torque, vehicle speed, throttle position, engine temperature, intake manifold pressure and rail pressure are the main variables of the FDI model. In this way, the model can be trained using relevant healthy and faulty features. Following that, the dataset is filtered and cleaned so that data quality is improved eliminating any negative factors during the data generation process. Specifically, the cleaning process aims at removing any outliers and compensating the missing samples in the dataset. Since the FDI model is developed based on a supervised learning approach, data labeling takes place before the training process. During this step, the classes are assigned to the corresponding data for classification

purposes. However, in the case of simultaneous faults, two classes should be considered at the same time in the labeling process. Therefore, in this study, a multi-class, multi-label approach [51] has been considered so that the potential fault combinations can be covered. However, to avoid the challenge of manual labeling, a data dictionary process has been developed to automatically identify all possible fault combinations. By doing so, a unique numeric label with a specific index can be created for each possible fault type pair. Noteworthy, in the ASSs, the sensors signals have different ranges of values. Therefore, using Z-Score normalization function, the variables' amplitudes are normalized and scaled uniformly to the range [0,1]. Mathematically, the scaling process of the input values is presented in [52].

Another complicating factor for the training process is imbalanced data, where the ratio between the faulty and healthy samples is disproportionate. In this case, the trained classifier might be biased toward the majority class, resulting in a poor prediction performance [53]. To tackle this problem, different approaches have been proposed in the literature, i.e., augmentation-based, feature learning-based and classifier design-based approach [54]. In this study, the random undersampling technique has been utilized to overcome the imbalanced data challenge. The core idea behind this technique is to remove randomly selected instances from the majority class until the required class balance is reached [55]. Due to the availability of a sufficient number of healthy samples in the HIL tests, the reduction of the dataset by the mentioned balancing technique has no negative impact on the training process. The motivation behind selecting the technique among other balancing techniques, e.g., random over-sampling and SMOTE, is that they may lead to overfitting by not matching the original time series patterns. Finally, the balanced data is split into a training, a validation and a testing subset. Specifically, 80% of the data is used for the training process, while the rest, 10% each, is used for the validation and testing process.

C. DAE-BASED DATA DENOISING

In real-world applications, noisy data with uncertainty patterns play a negative role in FDI process based on a data-driven approach. Noteworthy, most of the developed models are based on clean historical simulation data under experimental laboratory conditions. Recently, Denoising Autoencoder (DAE) has been introduced as a powerful method to overcome the noisy data challenge [56]. DAE model is constructed based on four different layers, i.e., input layer, corrupted layer, hidden layers, and output layer. The architecture of the hidden layer consists of two main modules, namely encoder and decoder. To utilize the denoising function of DAE, the original input data (I) should be corrupted with a certain noise level, e.g., Gaussian noise. The encoder aims to map the input data into a lower dimensional representation, i.e., a latent code, in which the extracted features (F) from the corrupted input data (C) are stored. The

reconstruction of the extracted features into the original data space, on the other hand, is done by the decoder module, known as reconstructed output (O). By doing so, a compact representation of the input data can be used for various tasks such as dimensionality reduction, data denoising or generative modelling. Mathematically, Equation (1) and (2) represent the encoder and decoder modules, respectively.

$$F = f_{enc}(WC + b) \quad (1)$$

$$O = f_{dec}(W'F + b') \quad (2)$$

where f_{enc}, f_{dec} denote an activation function of the encoder and the decoder, respectively. W and W' are the weight of encoder and decoder, respectively. (b) and (b') are the offset vectors of the encoder and decoder, respectively.

The training approach of the DAE model is driven by the idea of minimizing the reconstruction loss between the input data (I) and the reconstructed output (O). As a result, the encoder and decoder modules are trained to represent and reconstruct the data effectively without noise. Hence, the performance of the DAE can be measured based on the reconstruction error, known as loss function, as can be shown in Equation (3)

$$L(I, O) = ||O - I||^2 \quad (3)$$

Autoencoders can be designed based on various architectures, e.g., DNN-based AE, CNN-based AE or LSTM-based AE. According to the remarkable achievements of GRU-based DAE compared to the other AE variants [57], in this study, GRU-based DAE has been utilized to enhance the robustness of the proposed FDI model against noise. Besides, the ability to meet the requirements of real-time applications with low resources and fast inference time were the motivation to consider GRU cell. Thanks to the internal structure of GRU, i.e., gate mechanisms, the information flow can be controlled, resulting in less computational effort and training parameters compared to the LSTM cell. The mathematical representation of the GRU cell equations is presented in [57].

D. ENSEMBLE LEARNING-BASED FAULT DETECTION AND IDENTIFICATION

Once the data is denoised using DAE, the correlation between the features of time series data is leveraged in such a way that the respective fault type is identified. The importance of this phase lies in the fact that different fault features can belong to two classes at the same time, which in turn, leads to misclassification in case of the complex dependent patterns. Therefore, to avoid the biased classification accuracy of the individual classifier, ensemble learning [58] is used in the proposed architecture. To this end, ensemble learning based on LSTM and RF classifiers is used. In particular, the voter mechanism is adopted to select the predicted output with high probability from both classifiers. By doing so, the deviation of the classification accuracy can be mitigated, and the fault identification process can be improved.

LSTM has been proven to be good not only in constructing long-term relationships in sequential data, but also in

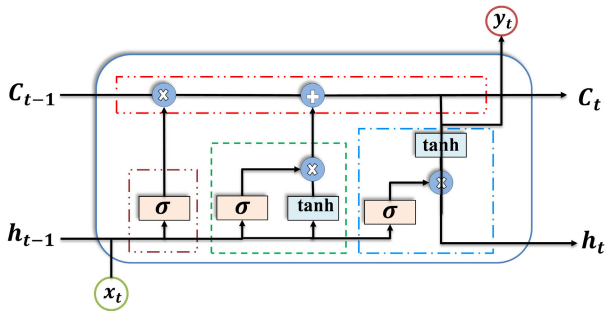


FIGURE 2. Internal structure of LSTM cell.

addressing the vanishing and exploding gradient problem of RNN. Besides, LSTM has shown high performance on large datasets for supervised classification problems compared to other techniques, e.g., CNN, MLP and GRU [59]. Furthermore, LSTM is considered a powerful technique for processing complex nonlinear data in a higher dimensional noisy space, providing high accuracy, reliability and effectiveness for FDI [59]. Therefore, due to the application scope and the data in our study, LSTM was selected for the development of the FDI model. The core structure of the model is the LSTM cell, which is constituted based on three gates, namely input, forget and output gate, as shown in Figure 2.

Besides the mentioned gates, a memory, known as Cell state, is employed to allow LSTM network to retain information over long sequences. The internal structure of the cell enables the information flow to be precisely regulated. Specifically, it determines how much new information to discard, retain or reinforce. Subsequently, based on the gating mechanism and memory cell, the tasks involving long-range dependencies can be effectively addressed. Equation (4) represents the output of the input gate. Whereas, the output of the forget gate is determined based on Equation (5), where $f_t \in 0, 1$, i.e., 0 for removing the data and 1 for retaining it. Two different activation functions are utilized to update the hidden state, i.e., (σ) and (\tanh). The output of current state C_t is determined by equation (7), where C_{t-1} represents the output of the previous state (memory unit).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_g) \quad (4)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (5)$$

$$d_t = \tanh(W_d \cdot [h_{t-1}, x_t] + b_d) \quad (6)$$

$$C_t = (f_t \times C_{t-1}) + (g_t \times d_t) \quad (7)$$

notable, the gate's value depends on the hidden state h_{t-1} and input x_t . Finally, the output cell can be mathematically represented by equation (9), where, the output gate is shown in equation (8)

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (8)$$

$$h_t = O_t \times \tanh(C_t) \quad (9)$$

As the ensemble learning method involving multiple decision trees, Random Forest (RF) is introduced as an

efficient technique for classification problems that avoid the problem of overfitting [60]. Among the various ML classifiers, RF outperforms other techniques, such as KNN and SVM, in terms of performance and computation time [61]. In addition to its resilience against overfitting, it shows outstanding performance in processing a large set of features [62]. Being simple, fast, robust and able to handle missing data, the RF classifier has achieved remarkable success in various applications [63], [64]. However, despite the mentioned benefits, the main drawback of RF is the inability to extract the features and capture the temporal dependencies present in time series data. Therefore, in this study, the representative features extracted by LSTM are employed and fed to the RF, ensuring high predictive performance. The core idea of RF is to aggregate the predictions of all trees formed based on three types of nodes, i.e., root, decision, and leaf node. The final decision is made based on the majority voting mechanism among the trees. Mathematically, the prediction output can be presented according to equation (10).

$$T = \text{Mode}T_1(x), T_2(x), \dots, T_i(x) \quad (10)$$

where the predicted output is represented by T , Mode is the majority voting operation, and $T_i(x)$ represents the prediction of the i -th Decision Tree (DT) in the forest.

The construction of the RF tree depends on the bagging technique, where multiple bootstrap samples are created from the original training dataset increasing the stability and the accuracy [65]. The training process is performed separately for each individual tree, using a random subset of the data and features with the same ratio. The training process is completed once the predefined criterion is reached, e.g., maximum depth or minimum number of samples in a leaf node.

Finally, the prediction results of the LSTM and the RF are combined, and based on the majority-voting method [66], the final classification decision with the highest count is made using the collective decision. As a result, by benefiting from both classifiers, the predictive performance and robustness are improved with less sensitivity to overfitting. Besides, model uncertainty can be addressed by considering the prediction decision of each classifier.

IV. CASE STUDY AND EXPERIMENTAL IMPLEMENTATION

In this section, the details of the case study used, including the system architecture, the platform setup and the implementation steps, are presented. Furthermore, the main phases and steps of the development of the proposed FDI model are described, i.e., data generation, data pre-processing, model training and testing.

A. SYSTEM ARCHITECTURE OF THE CASE STUDY

According to the Automotive Safety Integrity Level (ASIL) of ISO 26262 standard [3], the failures in the engine management system are classified as C to D being considered as very critical with high severity class. Therefore, the

gasoline engine system has been selected as a case study to demonstrate the applicability of the proposed method and validate the performance of the developed DL-based FDI model. Thus, to validate the target FDI model and demonstrate its capability, two system models provided by dSPACE have been utilized, i.e., ASM Gasoline Engine and ASM Vehicle Dynamics with traffic [67]. Notably, the mentioned models have been modified and integrated together so that a digital test drive is enabled to ensure the comprehensive characteristics of the engine system.

At the software level, the engine system has been modelled and simulated in the MATLAB/Simulink environment with high fidelity so that comprehensive characteristics of the system behavior can be captured. More specifically, the detailed subsystems of the engine with their components and connections were considered in the model architecture. The main subsystems of the engine system are air path system, fuel system, piston engine system, exhaust system and cooler system, as can be observed in Figure 3. Besides, using Model-based Design (MBD) approach, the interaction between the target systems with their environment has been considered by modelling the powertrain, vehicle dynamics systems as well as the driving environment. By doing so, the main characteristics of the entire vehicle, e.g., longitudinal driving, vehicle resistances, transmission and driver characteristics can be obtained. Finally, the control algorithm, i.e., System Under Test (SUT), which is directly connected to the controlled plant in a closed loop control, has been designed as a behaviour model referred to as SoftECU.

At the hardware level, in our study, dSPACE MicroAuto-Box II is used as Rapid Control Prototype (RCP) to emulate the functionality of the real ECU and to execute the control algorithms. The mentioned RCP (DS1401 Base Board) has 900 MHz processor, 6th Gen.Intel®Core™ i7-6822EQ, 16 MB memory and 340 ms boot time for 3 MB application. dSPACE SCALEXIO, in turn, is employed as a real-time simulator to comprehensively and accurately simulate the complex controlled system, i.e., the gasoline engine with the vehicle dynamics. The sensors and the actuators' signals between the real-time simulator and the MicroAutoBox are transmitted via a CAN bus, while the connection to the host PC is established via an Ethernet. To enable the digital test drive considering the user's behaviour, the real wheel and pedal are connected to the HIL system. By doing so, the driving scenarios can be performed either automatically by the machine or by the user based on the defined requirements. A virtual driving environment with dynamic traffic has been designed and modelled using ModelDesk. Besides, a 3-D visualization of the environment has been enabled by MotionDesk, as can be seen in Figure 4.

Before executing the model on the target machine, the model's parameters should be set. For this purpose, the ModelDesk tool is used to specify both the internal and the external system specifications according to the user's requirements. The core specifications of the selected case study are listed in [47].

Thanks to the property of the MBD approach of generating the code of complex embedded systems from the model, the target application can be automatically deployed on the target hardware. To be specific, the generated model code of the control system and the plant are loaded into the ECU and the HIL simulator, respectively. Once the application is available for execution, the driving tests are configured based on the user's requirements. In this step, the driving scenarios and the driving mode are defined. Besides, the ControlDesk tool can be used to perform online parameterization, instrumentation, controller calibration and recording of the measurements. Noteworthy, the SoftECU model and the real ECU of the target case study allow the execution in two modes, i.e., simulation/offline mode and real-time/online mode.

B. REPRESENTATIVE DATASET GENERATION

As a golden run behaviour, the system is executed in the real-time under fault-free conditions. By doing so, the healthy dataset, including the sensors and actuators signals, can be collected, representing the normal behaviour of the SUT. Using the ControlDesk tool, "city" and "highway" scenarios have been selected from the list of driving scenarios. The test scenario used to collect the data under healthy and faulty conditions is illustrated in Figure 5a and 5b as vehicle and engine behaviour under fault-free conditions, respectively.

On the other hand, the faulty dataset has been generated by injecting several types of faults, individually and simultaneously, into the sensor signal during the real-time execution using a real-time FI framework. FI attributes have been specified so that the representative and realistic permanent and transient sensor faults during the driving cycle are injected. Thus, the effect of critical faults causing the failure at the system level in real-time is captured.

Aiming at injecting most of the sensor faults occurring in the time series data, five fault types with their combinations have been considered in this study. In concrete terms, the considered fault types are Gain, Stuck-at, Noise, Drift and Delay faults. The occurred faults in the Accelerator Pedal Position (APP) and the engine speed sensor (RPM) can have a serious impact on the vehicle behaviour in terms of safety [68]. Therefore, APP and RPM sensors have been selected as potential fault locations in the target system. As the objective of our target FDI model is to cope with imbalanced data, transient faults have been covered in this study.

To analyze and capture the system behaviour under simultaneous faults, the combination of the aforementioned faults types have been injected into the target location mutually. To be demonstrated, in each experiment, two different fault types, e.g., Gain and Noise, have been injected into APP and RPM sensor for a specific duration, respectively. Thus, the single and simultaneous faults have been injected for a certain duration and deactivated again after a short time. Based on the selected driving scenario, the faults have been injected for 170-330 sec. In total, Besides the healthy class, 15 different classes from 15 FI experiments as single and combination

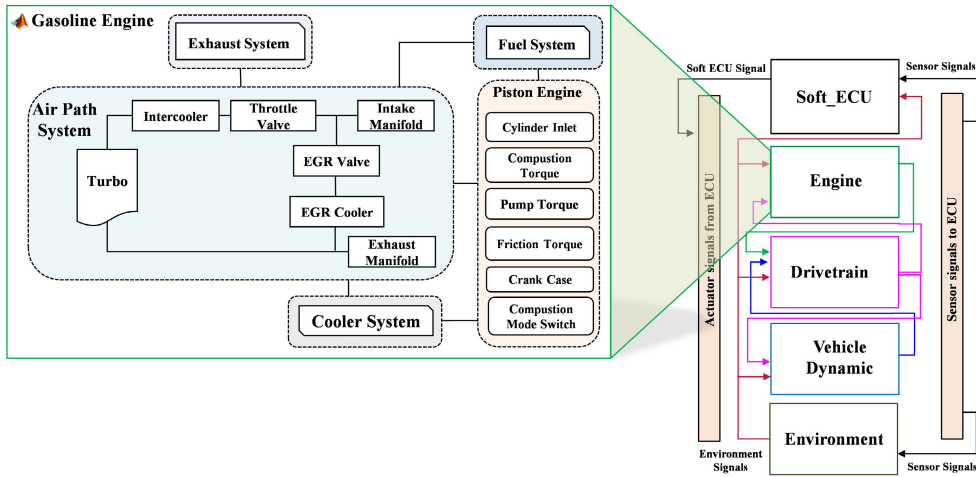


FIGURE 3. System architecture of the used case study.



FIGURE 4. HIL-based digital test drive platform.

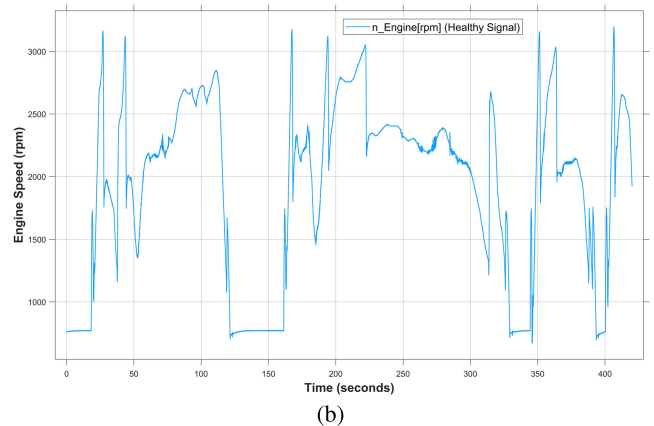
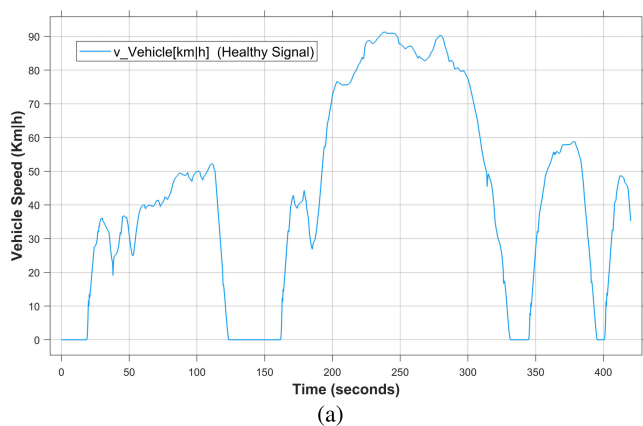


FIGURE 5. Selected test drive scenario as a desired behaviour (golden run). (a) Vehicle system behavior under fault-free conditions. (b) Engine system behavior under fault-free conditions.

faults have been collected. The illustration of the system behaviour under simultaneous faults can be seen in Figure 6. Specifically, the effect of injecting stuck-at and delay

faults simultaneously on the engine speed, engine torque, engine temperature and vehicle speed is demonstrated in Figure 6 a,b,c and d, respectively.

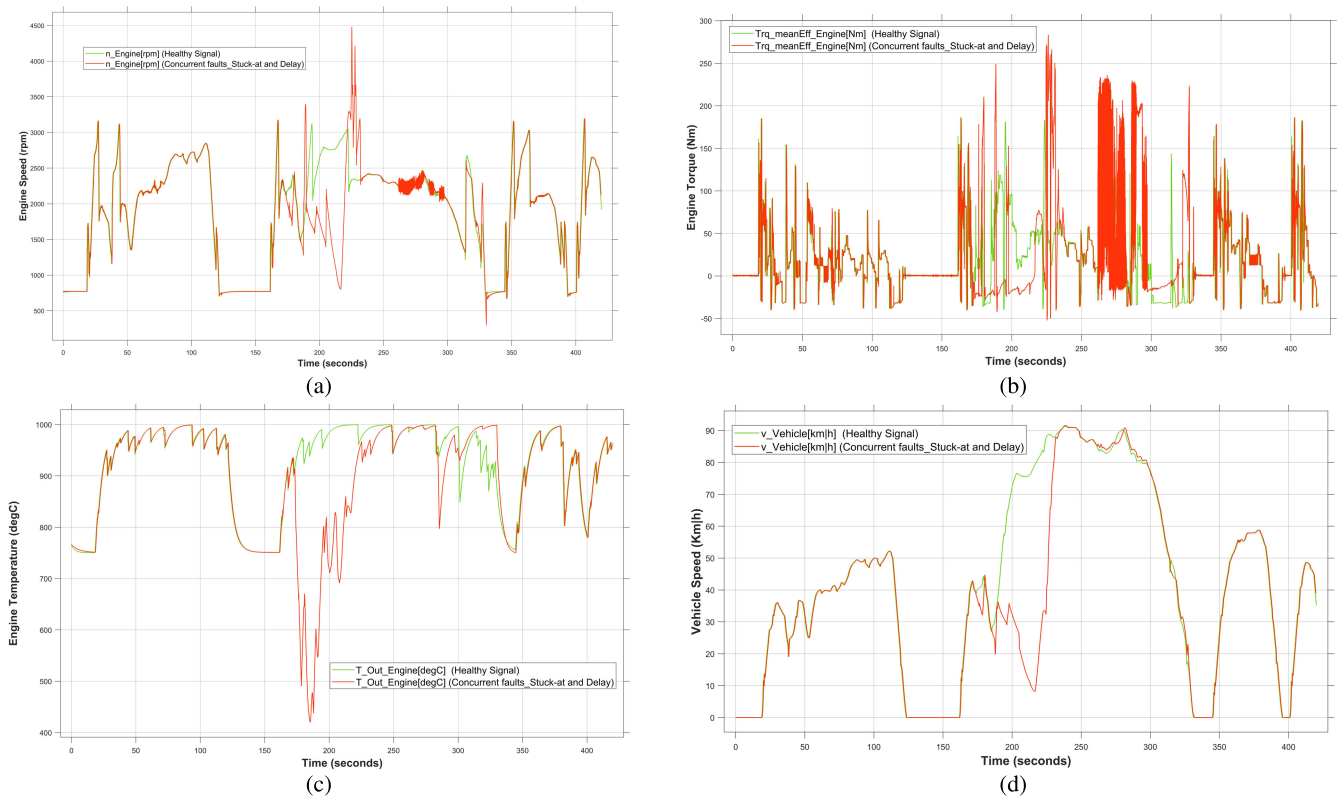


FIGURE 6. The effect of transient simultaneous faults on the system behaviour. (a) Engine Speed behaviour under stuck-at and delay faults. (b) Engine torque behaviour under stuck-at and delay faults. (c) Engine temperature behaviour under stuck-at and delay faults. (d) Vehicle Speed behaviour under stuck-at and delay faults.

C. DATASET DESCRIPTION AND PREPROCESSING

To facilitate the steps of data preprocessing, the recorded data is saved in a CSV format. Beside the healthy data samples, as a result of 15 experiments of FI, 16 CSVs have been collected including faulty data. The sampling time in all experiments is 0.001 sec. The total number of data samples from the data collection phase is 44.800.000, with 2.800.000 samples for each experiment.

As explained in the methodology section, once the data is collected from the sensors, the pre-processing phase takes place. This phase aims to clean and format the collected data so that the target model can be trained efficiently. The higher the quality of the data, the more efficient the performance of the model resulting from the training. To this end, several techniques were applied to our dataset. Firstly, in addition to removing the outliers and duplicate samples in the dataset, the missing values were also identified and treated to clean the data. Next, the data is visualized and analyzed using the Simulation Data Inspector from MATLAB so that the data distribution, patterns and relationships between the variables can be defined. Since the selection of features plays a crucial role in the performance of the classification tasks by the trained model, the most important system variables that contribute to the performance of the model were selected. The main variables considered in this study are throttle position [%], engine temperature [degC], mean effective engine torque

[Nm], engine speed [rpm], intake manifold pressure [Pa], rail pressure [bar] and vehicle speed [Km/h], as shown in Figure 7. Data labelling is the next step where the categorical variables are coded into numerical representations. In this study, a Label Power Set (LPS) based on a multi-label multi-class strategy was used to perform the labeling process. By doing so, all possible combinations of fault types can be represented in pairs. In addition, the Z-Score function is used to normalize and scale the values within the same range. Imbalanced data is also handled by applying the random subsampling technique. Finally, before splitting the data, the data is standardized so that the numerical characteristics have a standard deviation of 1 and a mean of 0. The collected data has been distributed into three portions, i.e., training, validation and testing part, 80% of the collected dataset has been assigned for the training process, whereas, 20% has been assigned for the validation and testing phase, respectively. The detailed distribution of the collected dataset is illustrated in Table 2. To train the DAE, a certain noise level is added to the processed data in order to obtain the corrupted samples as input for the encoder module.

D. TRAINING AND OPTIMISATION

The implementation steps of the proposed method have been carried out using Google Colab, in which TensorFlow framework [69] is used with the Python programming

A	B	C	D	E	F	G	H	
Time	Pos_Throttle[%]	T_Out_Engine[degC]	Tq_meanEff_Engine[Nm]	n_Engine[rpm]	p_InMan[Pa]	p_Rail[bar]	v_Vehicle[km/h]	
1	0	766.1549711	1.00166817	773.430293	99828.20251	119.9560256	2.88739E-43	
2	0.0010023	100	766.1533018	1.001458454	772.922836	99828.30085	119.9572804	2.85851E-43
3	0.0019988	100	766.1519049	1.000509247	772.3807066	99828.40558	119.9582441	2.82993E-43
4	0.0029966	100	766.1504	0.99992058	771.7953642	99828.51573	119.958902	2.80163E-43
5	0.0039949	100	766.1487788	0.99713677	771.1613802	99828.63206	119.9592137	2.77361E-43
6	0.0050007	100	766.147031	0.994949456	770.5228867	99828.75945	119.9592149	2.74588E-43
7	0.006002	100	766.1451458	1.008464126	770.0810469	99828.89622	119.9588339	2.71842E-43
8	0.007004	100	766.1431257	1.018724629	769.8828704	99829.03592	119.9581527	2.69123E-43
9	0.0080013	100	766.1410291	1.0283194	769.7506943	99829.18178	119.9577883	2.66426E-43
10	0.0090023	100	766.1389869	1.038856349	769.8515836	99829.20757	119.9563308	2.63768E-43
11	0.0100007	100	766.1367643	1.048446959	770.1004704	99829.23369	119.9539556	2.6113E-43
12	0.0110003	100	766.1346663	1.051186337	770.4847562	99829.22221	119.9546881	2.58510E-43
13	0.0119994	100	766.1326339	1.056609454	770.9837285	99829.17644	119.9539395	2.55934E-43
14	0.013	100	766.130694	1.060770541	771.573572	99829.09983	119.9535491	2.53374E-43
15	0.0139996	100	766.1288692	1.063543846	772.2276007	99828.99686	119.9534831	2.5084E-43

FIGURE 7. Collected dataset from real-time HIL simulation.

TABLE 2. Description of the generated dataset with the fault classes.

Fault ID	Fault Type	Fault Duration (sec)	Training Samples	Testing Samples
H	Healthy	-	2240000	280000
F1	Gain	166–335	2240000	280000
F2	Stuck-at	179–328	2240000	280000
F3	Noise	174–340	2240000	280000
F4	Delay	170–318	2240000	280000
F5	Drift	176–333	2240000	280000
F1F2	Gain & Stuck-at	164–323	2240000	280000
F1F3	Gain & Noise	169–315	2240000	280000
F1F4	Gain & Delay	175–330	2240000	280000
F1F5	Gain & Drift	174–324	2240000	280000
F2F3	Stuck-at & Noise	176–334	2240000	280000
F2F4	Stuck-at & Delay	180–340	2240000	280000
F2F5	Stuck-at & Drift	171–338	2240000	280000
F3F4	Noise & Delay	166–325	2240000	280000
F3F5	Noise & Drift	177–338	2240000	280000
F4F5	Delay & Drift	173–342	2240000	280000

language. The DL model has been trained based on the balanced training data which is fed from the preprocessing phase. Following the pre-processing and balancing of the data, model development is carried out in four phases, namely model design, model training, model validation and evaluation, as illustrated in Figure 8. In the model design phase, the parameters and configurations of the network architecture are identified. Besides, the initialization of the model’s hyperparameters takes place. Specifically, the number of layers, epoch, learning rate, batch size, noise level, activation function and the optimizer are specified.

Noteworthy, in case of training DAE, in addition to the mentioned hyperparameters, Gaussian noise levels are determined. In our case the target model has been trained based on four levels of noise, i.e., 3%, 6%, 8% and 10%, added to the original dataset.

On the other hand, RF training requires determining the number of decision trees, the maximum depth of each decision tree and the random number generator. Besides, the “out-of-bag (OOB)” error estimation and “warm start” features are enabled. Once the training process is initiated, the loss function is calculated and the model performance is tracked. Based on the propagated loss, the internal model parameters are updated accordingly. Figure 9 illustrates the training process curve of the DAE, LSTM and RF. Noteworthy, the performance of the trained model highly depends on the defined hyperparameters. Therefore, the trained model is optimized by tuning the hyperparameters, known as the model optimization process. To this end, using validation data, the model’s performance is evaluated to check whether or not the convergence has been achieved.

TABLE 3. Optimal hyperparameters of LSTM architecture.

Hyperparameter	Optimal Values
Hidden Layer Size	512
Epochs	60
Batch Size	1024
LSTM Layers	3
Learning Rate	0.0005
Activation Function	tanh
Optimizer	adam

Specifically, certain hyperparameters are tuned in such a way that the model accuracy is improved based on the optimized architecture. In our case, a grid search mechanism has been employed so that different combinations of hyperparameters tuning are performed. The core idea of the technique is to establish a grid of hyperparameters’ values with the potential combinations. Then, based on validation cross, the model’s performance is analysed for each combination enabling the selection of the optimal values that provide the best performance. However, due to the model’s complexity, the trade-off between the computational cost of the training and the generalization issue should be considered during the implementation. Detailed specifications of the selected optimal hyperparameters of the proposed LSTM are presented in Table 3.

V. RESULTS AND DISCUSSION

In this section, the experimental results of the proposed model are discussed and analyzed. In particular, using a test dataset, the effectiveness of the proposed FDI model is demonstrated in terms of accuracy. Furthermore, the superiority of the proposed architecture compared to stand-alone techniques is presented. To evaluate the detection and identification performance of the proposed model, three evaluation metrics are used in this study, i.e., precision, recall and F1-Score [70]. On the other hand, Mean Square Error (MSE) [71] is used to evaluate the performance and effectiveness of the GRU-based DAE.

A. GRU-BASED DAE PERFORMANCE

To demonstrate the anti-noise capability of the proposed model, it was evaluated under different Gaussian noise levels, i.e., 3%, 6%, 8% and 10%. In Figure 10, the performance of the developed model in reconstructing the original data under different levels of noise can be clearly observed. Specifically, at low noise levels, the developed model shows high performance in terms of minor reconstruction error with 0.023 MSE, which gradually increases at higher levels. This means that almost no information is lost in the denoising and reconstructing process. It is worth noting that even at the highest noise level of 10%, the denoising performance of the model is still acceptable, with an MSE of 0.0618. Compared to other structures of DAE, e.g., ANN-based DAE, the superiority of the proposed model can be observed with a low MSE. Besides the noise-free level, the reconstruction error of the proposed model is significantly lower with 0.0234, 0.0421, 0.0504 and 0.0618 MSE, at noise levels 3%,

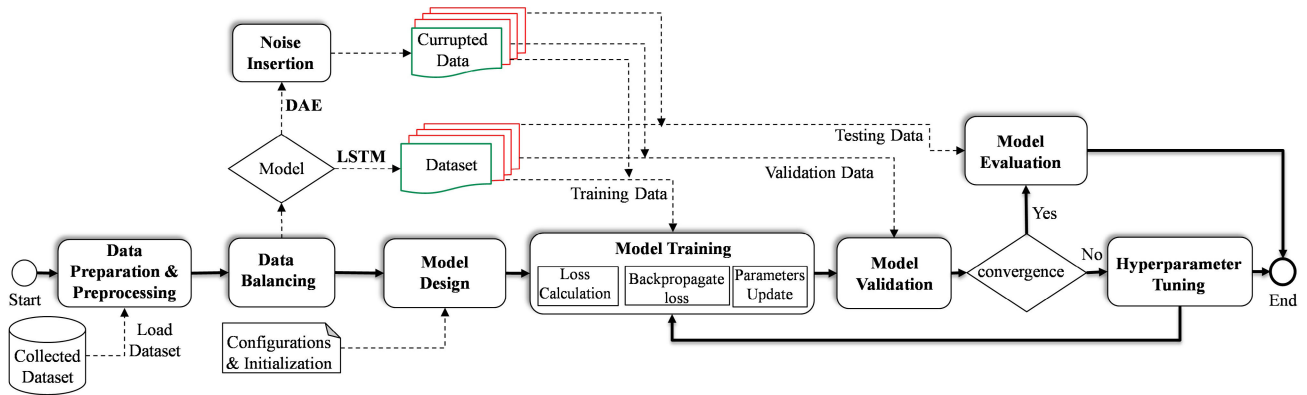


FIGURE 8. Flowchart of model training and optimization.

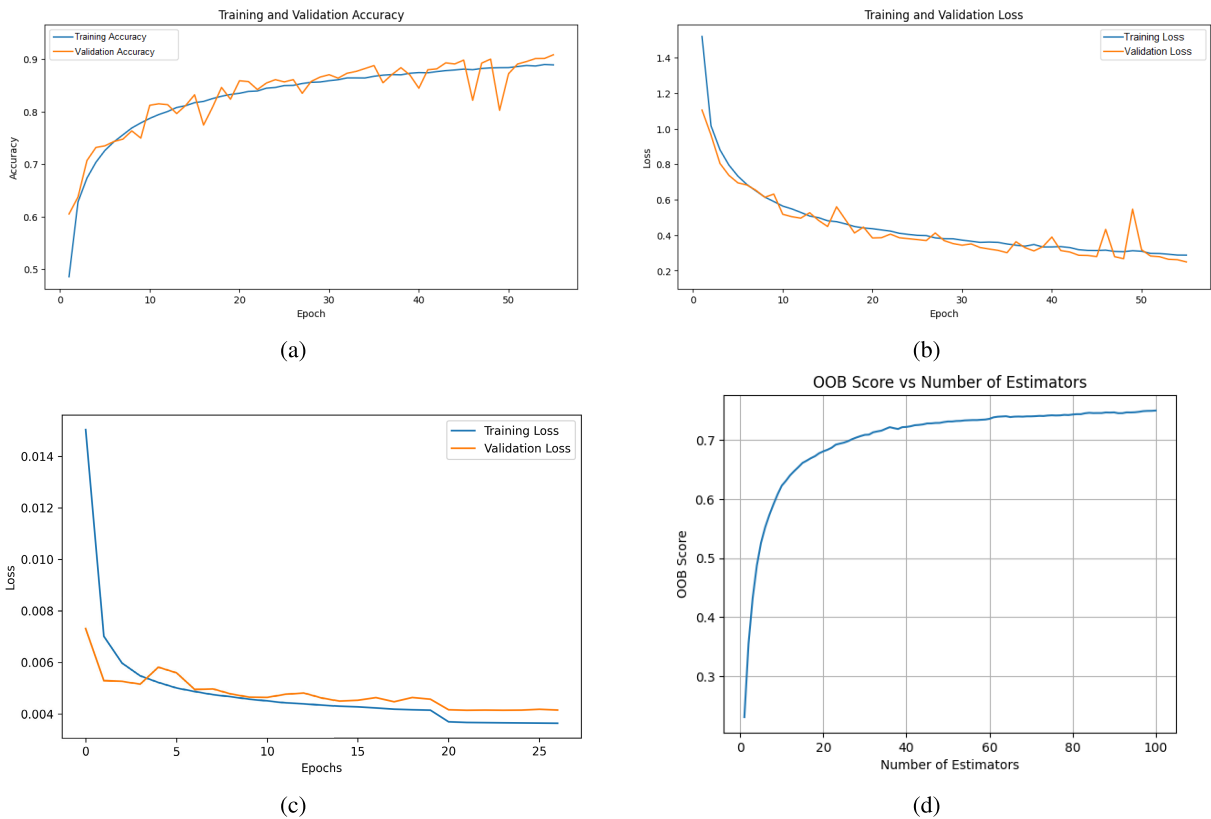


FIGURE 9. Hyperparameter optimization results. (a) Training and validation accuracy of LSTM. (b) Training and validation Loss of LSTM. (c) Training and validation Loss of GRU-DAE. (d) OOB Score of RF.

6%, 8% and 10%, respectively. Thus, the effectiveness of the DAE structure can be proven by the validation results not only in reconstructing the original data without loss, but also in denoising the data with high performance. However, the higher the level of the added noise, the higher the error in reconstructing and denoising the original data.

B. DETECTION AND CLASSIFICATION RESULTS USING ENSEMBLE LSTM-RF

The performance of developed ensemble classifiers with optimized architectures has been evaluated using a testing dataset in terms of precision, recall and F1-Score. Fault

identification performance of the model under various faults classes, i.e., single and concurrent faults, is illustrated in Figure 11.

In the case of single faults, i.e., gain, stuck-at and noise, it can be emphasized that the achieved identification performance of the proposed model is obviously high with a score above 97% in all evaluation metrics. However, due to the complexity of the pattern and corresponding features of delay and drift faults, the classifiers' performance decreases to reach F1-score of 92.57% and 82.92%, respectively. The poor sensitivity performance in the mentioned types with recall values of 88.33% and 83.33% is caused by the high

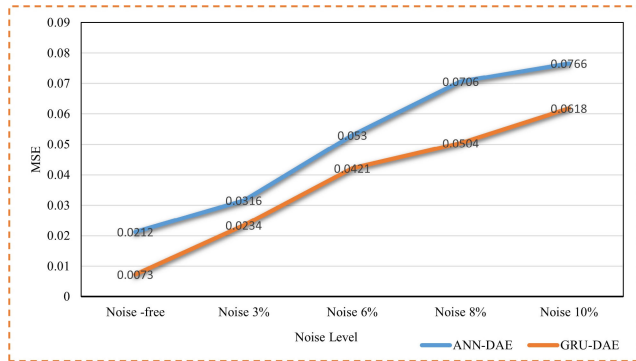


FIGURE 10. GRU-DAE performance under various level of noise.

rate of falsely identified faulty samples as negative instances. Nevertheless, remarkable achievement can be observed by detecting the faulty state of the system with a harmonic average 99.43% F1-Score.

Based on the classification results of the concurrent faults, the effectiveness of the ensemble learning can be demonstrated with an over 92% accuracy as an average of precision, recall and F1-Score. The highest accuracy was recorded for the identification of the gain-noise and gain-delay classes, with F1-Scores of 98.65% and 97.92%, respectively. However, on the other hand, the performance of the classifier drops slightly and settles at around 90% F1-Score for the other classes. The worst accuracy value of F4F5 is caused by the healthy samples being misclassified as faulty behaviour, which is a so-called false alarm. In summary, the applicability of the proposed model in detecting and identifying composite faulty behaviour with high reliability has been demonstrated. Even at low values of the evaluation metrics, the performance of the model is still satisfactory within an acceptable range.

As a graphical representation of the FDI performance for each fault type, the ROC curve is used. This allows the relationship between the rate of true positives (TPR) and the rate of false positives (FPR) to be plotted. Figure 12a and Figure 12b show to what extent the model is able to distinguish between types of faults, individually and simultaneously. For example, the ROC curve of the class 0 (gain fault) with ROC of 0.99 indicates that the gain fault can be accurately identified by the model with a probability of 99%. On the other hand, the chance of the model to correctly identify the concurrent faults, for example, F1F2, is 93%. The better recognition performance arises from the ROC value tending towards the upper left corner.

C. MODEL ROBUSTNESS AGAINST NOISE

To evaluate the FDI performance of the proposed model against noise, test data with different noise levels is used. Thanks to the proposed GRU-DAE as the primary step before the FDI phase, the proposed classifier shows remarkable performance under noisy conditions. From the evaluation results in Figure 13, the high accuracy of the proposed FDI

model in terms of precision, recall and F1-score can be noticed.

Specifically, the model shows high accuracy with over 94% F1-Score under the first two noise levels, i.e., 1%, 4%. This robustness is achieved by applying DAE to reconstruct and denoise the original data with a very low MSE. However, increasing the noise level has a negative effect on the model performance. Consequently, the scores of the evaluation metrics decrease to about 91% at a noise level of 15%. Nevertheless, the model's ability to cope with the challenge of noise is still satisfactory at high levels, i.e., 20% noise, with an accuracy of 89% F1-Score. Thus, it can be concluded that the robustness of the proposed FDI model to noise has been improved by the denoising process using GRU-DAE. Finally, it is worth mentioning that the applicability of the proposed model has been investigated with two automotive case studies, i.e., a gasoline engine system and a vehicle dynamic system. Both case studies show high performance of the model with low reconstruction error. Besides the dataset1 obtained from automated test drives, the test dataset2 acquired from manual digital test drives has been used to evaluate the proposed model. Specifically, it was shown that the proposed model performed well in denoising and reconstructing data with MSE of 0.0421 and 0.0955 based on dataset1 and dataset2, respectively.

D. CLASSIFICATION RESULTS COMPARED TO STAND-ALONE ALGORITHMS

The superiority of the proposed ensemble learning is demonstrated by comparing the classification results obtained by our target model with those obtained by single methods. Specifically, the performance of the LSTM-RF, LSTM and RF models for each class, including single and concurrent faults, are compared in terms of precision, recall and F1-Score, as shown in Table 4, 5 and 6, respectively. The comparison results in Table 4 indicate that the ensemble models outperform the single models with an average precision of 93.57%. Whereas the average precision of LSTM and RF are 92.3% and 70.16% respectively. It is evident that the number of false positives in the proposed method has been reduced by considering the decisions of LSTM and RF. However, in some classes such as delay fault, the LSTM shows a better performance in identifying the class, which in turn improves the identification performance of the F3F4 class. The normalized confusion matrix is presented in Figure 14. By the confusion matrix, the identification performance of the proposed model based on the testing dataset is demonstrated. It can be noted that some data samples of the delay fault were incorrectly identified as simultaneous stuck-at and delay faults. A similarly poor prediction performance was achieved with class F2F3. The reason for this lies in the similarity of the signal characteristics of the mentioned fault class. In addition, a few samples of class F1F2 were misclassified by the model as class F3F4. On the other hand, the highest identification accuracy was found for the class F1F3, i.e., gain and noise faults.

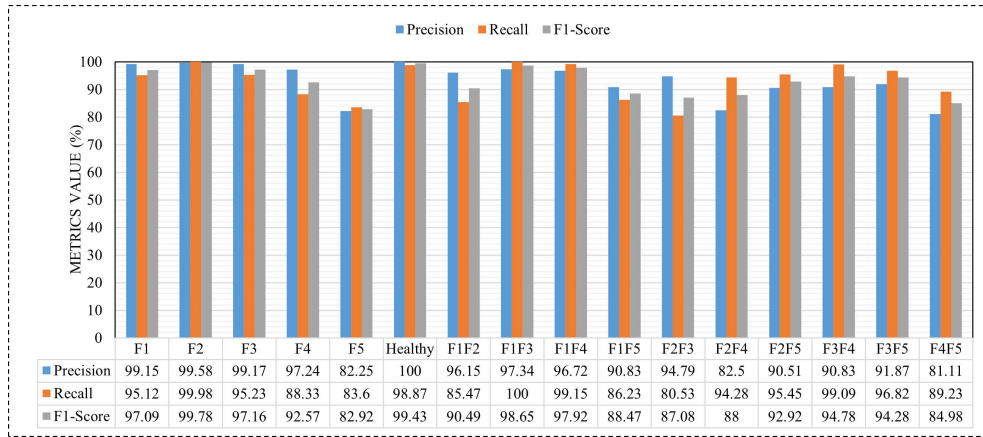


FIGURE 11. Testing results of the proposed model for single and concurrent faults.

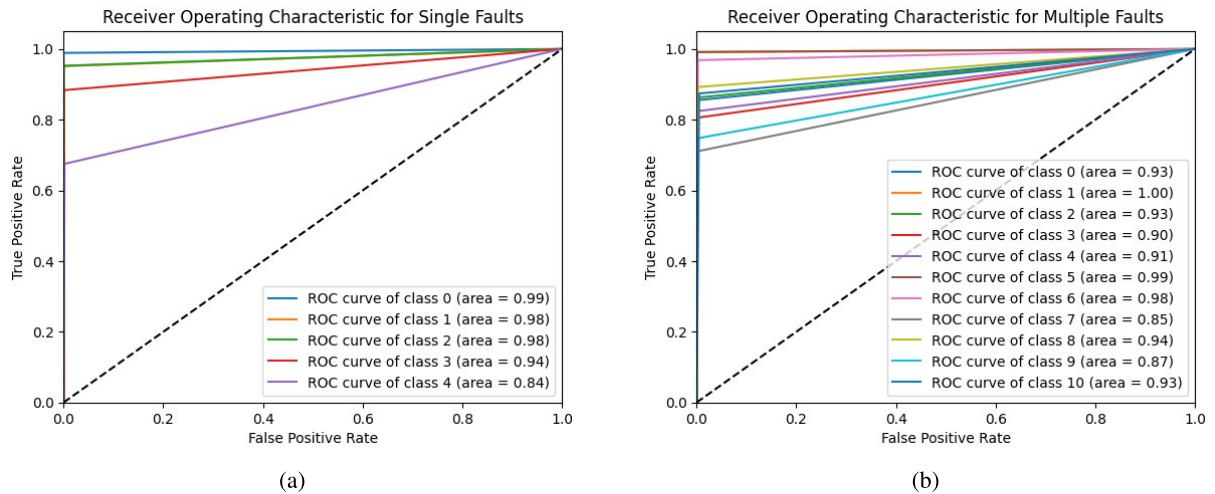


FIGURE 12. Fault identification performance in terms of AUC-ROC curve. (a) AUC-ROC curve of the proposed FDI model for single faults. (b) AUC-ROC curve of the proposed FDI model for concurrents faults.

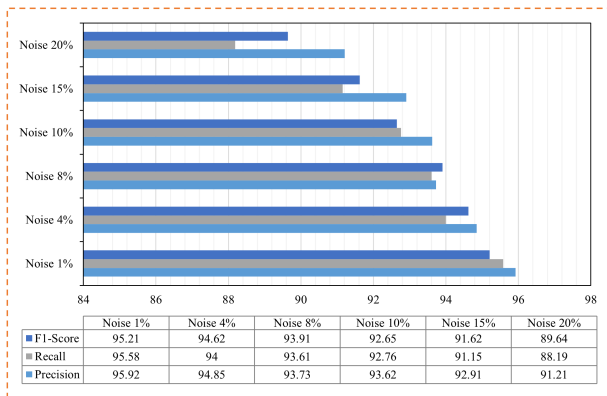


FIGURE 13. Fault identification performance of the proposed model under various levels of noise.

Similarly, the sensitivity of the proposed model to identifying all the faulty features correctly outperforms the stand-alone methods. As shown in Table 5, the recall value of the ensemble learning is above 95% in most classes.

TABLE 4. Precision score of the proposed FDI model compared to individual methods (%).

ID	Type	LSTM	RF	LSTM-RF
H	Healthy	99.99%	99.88%	100%
F1	Gain	96.94%	45.8%	99.15%
F2	Stuck-at	97.78%	85.98%	99.17%
F3	Noise	89.95%	53.65%	97.24%
F4	Delay	84.97%	70.24%	82.25%
F1F2	Gain-Stuck-at	94.58%	73.42%	96.15%
F1F3	Gain-Noise	97.98%	89.47%	97.34%
F1F4	Gain-Delay	88.65%	54.16%	90.83%
F2F3	Stuck-at-Noise	90.66%	53.09%	94.79%
F2F4	Stuck-at-Delay	90.65%	67.60%	90.51%
F3F4	Noise-Delay	83.95%	78.51%	91.87%

Moreover, even for complex faulty patterns, e.g., F4, F1F4 and F2F3, the proposed model shows better performance than LSTM and RF. The harmonic mean between recall and precision concludes that most fault classes can be accurately detected and identified by our proposed method, as shown in Table 6. Besides the individual classes, the proposed model

TABLE 5. Recall score of the proposed FDI model compared to individual methods (%).

ID	Type	LSTM	RF	LSTM-RF
H	Healthy	98.77%	96.05%	98.87%
F1	Gain	92.33%	61.85%	95.12%
F2	Stuck-at	95.67%	78.63%	95.23%
F3	Noise	89.62%	71.54%	88.33%
F4	Delay	70.22%	65.89%	83.6%
F1F2	Gain-Stuck-at	86.38%	86.06%	85.47%
F1F3	Gain-Noise	98.64%	91.07%	100%
F1F4	Gain-Delay	84.37%	55.32%	86.23%
F2F3	Stuck-at-Noise	79.83%	46.87%	80.53%
F2F4	Stuck-at-Delay	96.24%	79.33%	95.45%
F3F4	Noise-Delay	97.92%	77.86%	96.82%

TABLE 6. F1-Score of the proposed FDI model compared to individual methods (%).

ID	Type	LSTM	RF	LSTM-RF
H	Healthy	99.38%	97.93%	99.43%
F1	Gain	94.58%	52.63%	97.09%
F2	Stuck-at	96.71%	82.14%	97.16%
F3	Noise	89.78%	61.32%	92.57%
F4	Delay	76.90%	68%	82.92%
F1F2	Gain-Stuck-at	90.30%	79.24%	90.49%
F1F3	Gain-Noise	98.31%	90.26%	98.65%
F1F4	Gain-Delay	86.46%	54.85%	88.47%
F2F3	Stuck-at-Noise	84.90%	49.79%	87.08%
F2F4	Stuck-at-Delay	93.36%	73.00%	92.92%
F3F4	Noise-Delay	90.40%	78.18%	94.28%

TABLE 7. Comprehensive analysis of FDI performance with different classifiers in terms of F1-Score (%).

Model	Single Fault	Simultaneous Faults
SVM	81.17%	77.93%
DT	79.48%	73.84%
MLP	90.66%	88.5%
1D-CNN	92.7%	87.44%
LSTM-RF	94.82%	91.2%

TABLE 8. Comparison between the results of the proposed method and other related works.

Reference	Method	Accuracy
[42]	PCA-multiclass SVM	84.93%
[43]	PCM	81.49%
[46]	DNN and 1D-CNNs	89%
[72]	MLBCL-EE	87.57%
[73]	HML-KNN	85.07%
[74]	SSAE	87.61%
Proposed work	DAE, LSTM-RF	91.2 %

the complex relationship between fault classes, especially in the presence of concurrent faults. In contrast, in our model, the decision is made based on a voting algorithm that consider the results of each model. Several other ML-based classifiers were implemented and evaluated on the test dataset. To this end, four classifiers were selected, i.e., SVM, DT, MLP and 1D-CNN. Table 7 shows the performance of the proposed model compared to the conventional classifiers in terms of F1-Score. The values shown in the table indicate the average fault identification accuracy of mentioned models in the case of individual and simultaneous faults. The aforementioned methods were evaluated using a test dataset with healthy and faulty data samples. Despite the good performance of DL architecture-based classifiers, i.e., 1D-CNN and MLP, the ensemble LSTM-RF classifier can provide superior performance compared to other traditional classifiers. In table 8, the performance of the proposed model in terms of accuracy is compared with those obtained previously in other related works. It can be concluded that the proposed method has achieved a significant improvement in the performance of simultaneous FDI compared to other methods. Moreover, due to the model’s robustness against noise, the proposed method can also be used for FDI problems in different systems from other domains.

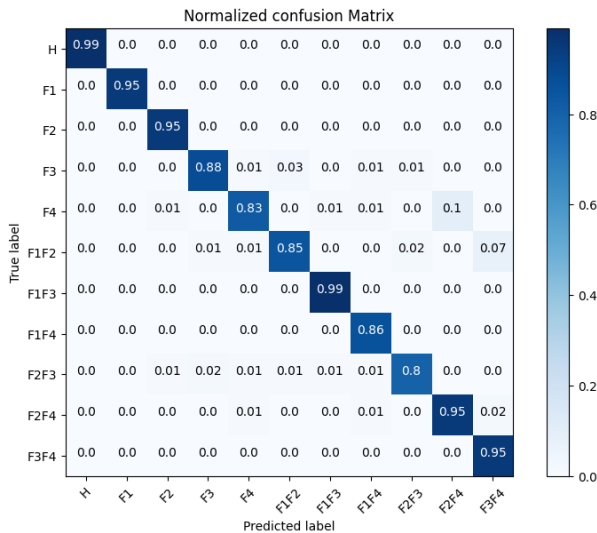


FIGURE 14. Confusion matrix of the proposed model with normalization.

has achieved high performance in the case of F1F2, F1F3 and F2F4 with F1-Score of 90.49%, 98.65% and 92.92%, respectively. On the other hand, the identification of the faulty state of the system in the case of delay fault still needs improvement. The reason for this drawback is the similarity between the healthy and faulty features used to develop the model. Conclusively, the performance of the individual method-based classifiers is significantly worse than that of our proposed model for the same training and testing data. This is due to the inability of the traditional method to capture

E. COMPUTATIONAL COMPLEXITY ANALYSIS

One of the main concerns in the development of DL models is the computational cost, especially in the case of having large amounts of data. Therefore, the required training and inference time of our target models is evaluated considering the specifications of the training platform, i.e., Google Colab, as shown in Table 9. An acceptable training time is required to develop a GRU-based DAE with an average computation time of 4445 seconds. Similarly, the computing time required for the testing process of data reconstruction and denoising is very low at 0.339 sec. On the other hand,

TABLE 9. Training and testing times of the proposed FDI model.

Model	Training Time (s)	Testing Time (s)
GRU-DAE	4445.17	0.339
LSTM-RF	5319.48	4.85

for the development of the proposed ensemble models, the training time required is 5319.48 sec. While for the testing of 10% of the total dataset, 4.85 sec is required. It is clear that due to the ensemble process of decision-making, a considerable amount of testing time is required. The higher the number of classifiers for ensemble learning, the higher the computation time required for training and testing. However, considering that the features are extracted automatically, the computational time is acceptable compared to the traditional ML method, which requires additional time for manual feature extraction. On the other hand, advances in computational resources can address this drawback and balance the trade-off between high performance and development time. Moreover, considering other simple DL architectures, e.g., GRU-based classifiers, paves the way for further improvements with less computational time. Notably, the problem of the computational cost of model development due to the size of dataset can be solved by improving the proposed method, so that the identification of compound faults can be dependent on the data of the individual faults.

VI. CONCLUSION

To address the classification problem of the concurrent faults during real-time validation of ASSs using HIL simulation, an ensemble learning-based method is proposed in this article. In particular, multi-label ensemble LSTM and Random Forest models have been developed. Unlike the conventional methods that focus on single faults, the aim of the study is to detect and identify single and simultaneous sensor faults at the system level, considering noisy and imbalanced data conditions. GRU-based DAE is adopted in this study to ensure the reliability and robustness of the proposed model against noise. Besides, to cope with the challenge of FDI model development in the presence of imbalanced data, a random undersampling algorithm is employed. Notably, real-time FI based on HIL simulation is utilized to analyze the critical faults and to collect the representative dataset. To validate the effectiveness and applicability of the proposed method, a high-fidelity model of a gasoline engine system is used as a case study, considering the entire vehicle dynamic system with its environment. According to the average value of the quantitative evaluation metrics, the single faults were classified with 94.82% F1-Score, 93.52% recall and 96.23% precision. Another promising finding was that the proposed model can accurately identify the concurrent faults with 91.2% F1-Score, 92.62% recall and 90.26% precision. Using the same dataset, the analysis results prove that the FDI accuracy of the single and simultaneous faults is significantly improved by our ensemble models compared to the traditional

single classifier. Thanks to the GRU-based DAE, the original data can be effectively reconstructed and denoised with a MSE of less than 0.05 at a noise level of 8%. This, in turn, contributes to improving the performance of the proposed model even in the presence of a high noise level, which shows outstanding performance with 92.65% F1-Score at 10% noise level. Besides, the model shows high detection performance with an average accuracy of 99.43%. All in all, the employment of DAE with an ensemble prediction model can provide a reliable and robust FDI for the real-time validation process of ASSs during HIL tests compared to individual methods. As a result, not only the safety and reliability of the target systems can be enhanced, but also the effort and time during the development process can be reduced.

In the future, the proposed model can be improved in terms of performance accuracy and computational time. Furthermore, the applicability of other simple DL architectures to build the ensemble classifiers for other development phases of ASSs can be investigated. Finally, the adaptability and applicability of the proposed model to FDI problems of systems from other industrial domains, e.g., railway and aviation, can be further investigated.

REFERENCES

- [1] J. Golias, G. Yanniss, and C. Antoniou, "Classification of driver-assistance systems according to their impact on road safety and traffic efficiency," *Transp. Rev.*, vol. 22, no. 2, pp. 179–196, Jan. 2002.
- [2] A. Vogelsang, "Feature dependencies in automotive software systems: Extent, awareness, and refactoring," *J. Syst. Softw.*, vol. 160, Feb. 2020, Art. no. 110458.
- [3] ISO 26262-10:2012-Road Vehicles—Functional Safety—Part 10: Guideline on ISO 26262. Accessed: Feb. 14, 2023. [Online]. Available: <https://www.iso.org/standard/54591.html>
- [4] G. Tibba, C. Malz, C. Stoermer, N. Nagarajan, L. Zhang, and S. Chakraborty, "Testing automotive embedded systems under X-in-the-loop setups," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2016, pp. 1–8.
- [5] V. Garousi, M. Felderer, Ç. M. Karapıçak, and U. Yılmaz, "Testing embedded software: A survey of the literature," *Inf. Softw. Technol.*, vol. 104, pp. 14–45, Dec. 2018.
- [6] E. Bringmann and A. Kr, "Model-based testing of automotive systems," in *Proc. Int. Conf. Softw. Test., Verification, Validation*, Apr. 2008, pp. 485–493.
- [7] R. Isermann and P. Ballé, "Trends in the application of model-based fault detection and diagnosis of technical processes," *Control Eng. Pract.*, vol. 5, no. 5, pp. 709–719, May 1997.
- [8] A. Theissler, "Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection," *Knowl.-Based Syst.*, vol. 123, pp. 163–173, May 2017.
- [9] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava, "Sensor network data fault types," *ACM Trans. Sensor Netw.*, vol. 5, no. 3, pp. 1–29, May 2009.
- [10] U. Saeed, S. U. Jan, Y.-D. Lee, and I. Koo, "Fault diagnosis based on extremely randomized trees in wireless sensor networks," *Rel. Eng. Syst. Saf.*, vol. 205, Jan. 2021, Art. no. 107284.
- [11] G. Sievers, C. Seiger, M. Peperhowe, H. Krumm, S. Graf, and H. Hanselmann, "Driving simulation technologies for sensor simulation in SIL and HIL environments," in *Proc. DSC*, 2018, pp. 127–130.
- [12] J. Zhou, R. Schmied, A. Sandalek, H. Kokal, and L. del Re, "A framework for virtual testing of ADAS," *SAE Int. J. Passenger Cars - Electron. Electr. Syst.*, vol. 9, no. 1, pp. 66–73, Apr. 2016.
- [13] A. Theissler, "Detecting anomalies in multivariate time series from automotive systems," Ph.D. thesis, School Eng. Des., Brunel Univ., London, U.K., 2013.

- [14] C. V. Jordan, F. Hauer, P. Foth, and A. Pretschner, "Time-series-based clustering for failure analysis in hardware-in-the-loop setups: An automotive case study," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSREW)*, Oct. 2020, pp. 67–72.
- [15] R. Isermann, "Model-based fault-detection and diagnosis—status and applications," *Annu. Rev. Control*, vol. 29, no. 1, pp. 71–85, Jan. 2005.
- [16] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757–3767, Jun. 2015.
- [17] W. Li, H. Li, S. Gu, and T. Chen, "Process fault diagnosis with model- and knowledge-based approaches: Advances and opportunities," *Control Eng. Pract.*, vol. 105, Dec. 2020, Art. no. 104637.
- [18] P. Jieyang, A. Kimmig, W. Dongkun, Z. Niu, F. Zhi, W. Jiahai, X. Liu, and J. Ovtcharova, "A systematic review of data-driven approaches to fault diagnosis and early warning," *J. Intell. Manuf.*, vol. 34, no. 8, pp. 3277–3304, Dec. 2023.
- [19] D. Gonzalez-Jimenez, J. del-Olmo, J. Poza, F. Garramiola, and P. Madina, "Data-driven fault diagnosis for electric drives: A review," *Sensors*, vol. 21, no. 12, p. 4024, Jun. 2021.
- [20] D. L. Nuñez and M. Borsato, "An ontology-based model for prognostics and health management of machines," *J. Ind. Inf. Integr.*, vol. 6, pp. 33–46, Jun. 2017.
- [21] X. Dai and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2226–2238, Nov. 2013.
- [22] F. Zhang, N. Saeed, and P. Sadeghian, "Deep learning in fault detection and diagnosis of building HVAC systems: A systematic review with meta analysis," *Energy AI*, vol. 12, Apr. 2023, Art. no. 100235.
- [23] S. Qiu, X. Cui, Z. Ping, N. Shan, Z. Li, X. Bao, and X. Xu, "Deep learning techniques in intelligent fault diagnosis and prognosis for industrial systems: A review," *Sensors*, vol. 23, no. 3, p. 1305, Jan. 2023.
- [24] Y. Shen and K. Khorasani, "Hybrid multi-mode machine learning-based fault diagnosis strategies with application to aircraft gas turbine engines," *Neural Netw.*, vol. 130, pp. 126–142, Oct. 2020.
- [25] M. I. Radaideh, C. Pappas, J. Walden, D. Lu, L. Vidyaratne, T. Britton, K. Rajput, M. Schram, and S. Cousineau, "Time series anomaly detection in power electronics signals with recurrent and ConvLSTM autoencoders," *Digit. Signal Process.*, vol. 130, Oct. 2022, Art. no. 103704.
- [26] N. Iqbal, "DeepSeg: Deep segmental denoising neural network for seismic data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 7, pp. 3397–3404, Jul. 2023.
- [27] S. A. A. Taqvi, H. Zabiri, L. D. Tufa, F. Uddin, S. A. Fatima, and A. S. Maulud, "A review on data-driven learning approaches for fault detection and diagnosis in chemical processes," *ChemBioEng Rev.*, vol. 8, no. 3, pp. 239–259, 2021.
- [28] J. Chen, L. Zhang, Y. Li, Y. Shi, X. Gao, and Y. Hu, "A review of computing-based automated fault detection and diagnosis of heating, ventilation and air conditioning systems," *Renew. Sustain. Energy Rev.*, vol. 161, Jun. 2022, Art. no. 112395.
- [29] P. Peng, W. Zhang, Y. Zhang, H. Wang, and H. Zhang, "Non-revisiting genetic cost-sensitive sparse autoencoder for imbalanced fault diagnosis," *Appl. Soft Comput.*, vol. 114, Jan. 2022, Art. no. 108138.
- [30] X. Liu, Q. Zhou, J. Zhao, H. Shen, and X. Xiong, "Fault diagnosis of rotating machinery under noisy environment conditions based on a 1-D convolutional autoencoder and 1-D convolutional neural network," *Sensors*, vol. 19, no. 4, p. 972, Feb. 2019.
- [31] M. Abboush, C. Knieke, and A. Rausch, "GRU-based denoising autoencoder for detection and clustering of unknown single and concurrent faults during system integration testing of automotive software systems," *Sensors*, vol. 23, no. 14, p. 6606, Jul. 2023.
- [32] A. Namigtle-Jiménez, R. F. Escobar-Jiménez, J. F. Gómez-Aguilar, C. D. García-Beltrán, and A. C. Téllez-Anguiano, "Online ANN-based fault diagnosis implementation using an FPGA: Application in the EFI system of a vehicle," *ISA Trans.*, vol. 100, pp. 358–372, May 2020.
- [33] R. Raveendran, K. B. Devika, and S. C. Subramanian, "Brake fault identification and fault-tolerant directional stability control of heavy road vehicles," *IEEE Access*, vol. 8, pp. 169229–169246, 2020.
- [34] 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.
- [35] D. Jung, "Data-driven open-set fault classification of residual data using Bayesian filtering," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 5, pp. 2045–2052, Sep. 2020.
- [36] L. Biddle and S. Fallah, "A novel fault detection, identification and prediction approach for autonomous vehicle controllers using SVM," *Automot. Innov.*, vol. 4, no. 3, pp. 301–314, Aug. 2021.
- [37] Y. Chen, M. Rao, K. Feng, and G. Niu, "Modified varying index coefficient autoregression model for representation of the nonstationary vibration from a planetary gearbox," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, 2023.
- [38] Y. Chen, G. Niu, Y. Li, and Y. Li, "A modified bidirectional long short-term memory neural network for rail vehicle suspension fault detection," *Vehicle Syst. Dyn.*, vol. 61, no. 12, pp. 3136–3160, Dec. 2023.
- [39] S. Scharoba, K.-U. Basener, J. Bielefeldt, H.-W. Wiesbrock, and M. Hübner, "Towards machine learning support for embedded system tests," in *Proc. 24th Euromicro Conf. Digit. Syst. Design (DSD)*, Sep. 2021, pp. 166–173.
- [40] M. Abboush, D. Bamal, C. Knieke, and A. Rausch, "Intelligent fault detection and classification based on hybrid deep learning methods for hardware-in-the-loop test of automotive software systems," *Sensors*, vol. 22, no. 11, p. 4066, May 2022.
- [41] S. Asgari, R. Gupta, I. K. Puri, and R. Zheng, "A data-driven approach to simultaneous fault detection and diagnosis in data centers," *Appl. Soft Comput.*, vol. 110, Oct. 2021, Art. no. 107638.
- [42] S. Li, H. Cao, and Y. Yang, "Data-driven simultaneous fault diagnosis for solid oxide fuel cell system using multi-label pattern identification," *J. Power Sour.*, vol. 378, pp. 646–659, Feb. 2018.
- [43] P. K. Wong, J. Zhong, Z. Yang, and C. M. Vong, "Sparse Bayesian extreme learning committee machine for engine simultaneous fault diagnosis," *Neurocomputing*, vol. 174, pp. 331–343, Jan. 2016.
- [44] A. R. Ramos, C. D. Acosta, P. J. R. Torres, E. I. S. Mercado, G. B. Baez, L. A. Rifón, and O. Llanes-Santiago, "An approach to multiple fault diagnosis using fuzzy logic," *J. Intell. Manuf.*, vol. 30, no. 1, pp. 429–439, Jan. 2019.
- [45] A. Othman, N. Iqbal, S. M. Hanafy, and U. B. Waheed, "Automated event detection and denoising method for passive seismic data using residual deep convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–11, 2022.
- [46] S. Safavi, M. A. Safavi, H. Hamid, and S. Fallah, "Multi-sensor fault detection, identification, isolation and health forecasting for autonomous vehicles," *Sensors*, vol. 21, no. 7, p. 2547, 2021.
- [47] M. Abboush, D. Bamal, C. Knieke, and A. Rausch, "Hardware-in-the-loop-based real-time fault injection framework for dynamic behavior analysis of automotive software systems," *Sensors*, vol. 22, no. 4, p. 1360, Feb. 2022.
- [48] T. Muhammed and R. A. Shaikh, "An analysis of fault detection strategies in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 78, pp. 267–287, Jan. 2017.
- [49] J. A. Crossman, H. Guo, Y. L. Murphey, and J. Cardillo, "Automotive signal fault diagnostics—Part I: Signal fault analysis, signal segmentation, feature extraction and quasi-optimal feature selection," *IEEE Trans. Veh. Technol.*, vol. 52, no. 4, pp. 1063–1075, Jul. 2003.
- [50] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.
- [51] C.-M. Vong, P.-K. Wong, and W.-F. Ip, "A new framework of simultaneous-fault diagnosis using pairwise probabilistic multi-label classification for time-dependent patterns," *IEEE Trans. Ind. Electron.*, vol. 60, no. 8, pp. 3372–3385, Aug. 2013.
- [52] S. Gopal Krishna Patro and K. Kumar Sahu, "Normalization: A preprocessing stage," 2015, *arXiv:1503.06462*.
- [53] P. Bedi, N. Gupta, and V. Jindal, "I-SiamIDS: An improved siam-IDS for handling class imbalance in network-based intrusion detection systems," *Int. J. Speech Technol.*, vol. 51, no. 2, pp. 1133–1151, Feb. 2021.
- [54] T. Zhang, J. Chen, F. Li, K. Zhang, H. Lv, S. He, and E. Xu, "Intelligent fault diagnosis of machines with small & imbalanced data: A state-of-the-art review and possible extensions," *ISA Trans.*, vol. 119, pp. 152–171, Jan. 2022.
- [55] T. Hasanin and T. Khoshgoftaar, "The effects of random undersampling with simulated class imbalance for big data," in *Proc. IEEE Int. Conf. Inf. Reuse Integr. (IRI)*, Jul. 2018, pp. 70–79.
- [56] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 1096–1103.

- [57] J. Cowton, I. Kyriazakis, T. Plötz, and J. Bacardit, "A combined deep learning GRU-autoencoder for the early detection of respiratory disease in pigs using multiple environmental sensors," *Sensors*, vol. 18, no. 8, p. 2521, Aug. 2018.
- [58] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Frontiers Comput. Sci.*, vol. 14, no. 2, pp. 241–258, 2020.
- [59] 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, Sep. 2020.
- [60] P. Kar, S. Banerjee, K. C. Mondal, G. Mahapatra, and S. Chattopadhyay, "A hybrid intrusion detection system for hierarchical filtration of anomalies," in *Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2018*, vol. 1. Singapore: Springer, 2019, pp. 417–426.
- [61] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, Dec. 2002.
- [62] M. Belgiu and L. Drăguț, "Random forest in remote sensing: A review of applications and future directions," *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 24–31, Apr. 2016.
- [63] S. Ma, M. Chen, J. Wu, Y. Wang, B. Jia, and Y. Jiang, "Intelligent fault diagnosis of HVCB with feature space optimization-based random forest," *Sensors*, vol. 18, no. 4, p. 1221, Apr. 2018.
- [64] M. A. Marins, B. D. Barros, I. H. Santos, D. C. Barrionuevo, R. E. V. Vargas, T. de M. Prego, A. A. de Lima, M. L. R. de Campos, E. A. B. da Silva, and S. L. Netto, "Fault detection and classification in oil wells and production/service lines using random forest," *J. Petroleum Sci. Eng.*, vol. 197, Feb. 2021, Art. no. 107879.
- [65] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, and Y. Si, "A data-driven design for fault detection of wind turbines using random forests and XGboost," *IEEE Access*, vol. 6, pp. 21020–21031, 2018.
- [66] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Inf. Fusion*, vol. 6, no. 1, pp. 63–81, Mar. 2005.
- [67] *Automotive Simulation Models (ASM)*. Accessed: Feb. 22, 2023. [Online]. Available: <https://www.dspace.com>
- [68] P. Koopman, "A case study of Toyota unintended acceleration and software safety," Pittsburgh, PA, USA, Sep. 2014, pp. 3–6.
- [69] `tf.keras.utils.normalize` | *TensorFlow Core v2.7.0*. Accessed: Feb. 22, 2023. [Online]. Available: <https://www.tensorflow.org>
- [70] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," 2020, *arXiv:2010.16061*.
- [71] H. Shao, H. Jiang, H. Zhao, and F. Wang, "A novel deep autoencoder feature learning method for rotating machinery fault diagnosis," *Mech. Syst. Signal Process.*, vol. 95, pp. 187–204, Oct. 2017.
- [72] Y. Wu, W. Jin, Y. Li, and D. Wang, "A novel method for simultaneous-fault diagnosis based on between-class learning," *Measurement*, vol. 172, Feb. 2021, Art. no. 108839.
- [73] X. Ma, Y. Hu, M. Wang, F. Li, and Y. Wang, "Degradation state partition and compound fault diagnosis of rolling bearing based on personalized multilabel learning," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.
- [74] Z. Zhang, S. Li, Y. Xiao, and Y. Yang, "Intelligent simultaneous fault diagnosis for solid oxide fuel cell system based on deep learning," *Appl. Energy*, vols. 233–234, pp. 930–942, Jan. 2019.



MOHAMMAD ABOUSH received the M.Sc. degree in mechatronic engineering from Siegen University, Siegen, Germany, in 2018. Since 2019, he has been a Research Associate with the Institute for Software and System Engineering, Clausthal University of Technology, Clausthal-Zellerfeld, Germany, where he is currently pursuing the Ph.D. degree. His current research interests include the verification and validation of automotive software systems, real-time hardware-in-the-loop (HIL)

simulation, intelligent fault detection and diagnosis, fault injection test, reliability and safety analysis, and machine learning. His work focuses on the development of intelligent failure analysis methods based on machine learning approach for real-time validation of automotive software systems.



CHRISTOPH KNIEKE received the degree in computer science from TU Braunschweig, Germany, the Diploma degree in computer science, in 2004, the Ph.D. degree, in 2011, and the Habilitation degree, in 2019. During this time, he was already a Research Assistant. Furthermore, he was a Lecturer with the TU Clausthal. In particular, he is interested in model-based systems engineering (MBSE), domain-specific modeling languages, software product line engineering, and architecture evolution. Within these fields, he was also the Leader of several research and development projects.



ANDREAS RAUSCH received the Ph.D. degree from the University of Munich, in 2001, under Prof. Dr. Manfred Broy. He is currently the Director of the Institute for Software and Systems Engineering, TU Clausthal, where he was the Head of the Chair for Software Systems Engineering, from 2007 to 2018. Until 2007, he was the Head of the Chair for Software Architecture, University of Kaiserslautern. His research interests include the field of software engineering focuses on software architecture, model-based software engineering, and process models, with more than 140 publications worldwide. He is involved in the management of and participation in a number of international, European, and national research projects.

• • •