



A Novel Fault Detection, Identification and Prediction Approach for Autonomous Vehicle Controllers Using SVM

Liam Biddle¹ · Saber Fallah¹

Received: 26 January 2020 / Accepted: 21 January 2021 / Published online: 5 April 2021
© The Author(s) 2021

Abstract

Faults that develop in vehicle sensors have the potential to propagate unchecked throughout control systems if undetected. Automatic fault diagnosis and health monitoring algorithms will become necessary as automotive applications become more autonomous. The current fault diagnosis systems are not effective for complex systems such as autonomous cars where the case of simultaneous faults in different sensors is highly possible. Therefore, this paper proposes a novel fault detection, isolation and identification architecture for multi-fault in multi-sensor systems with an efficient computational burden for real-time implementation. Support Vector Machine techniques are used to detect and identify faults in sensors for autonomous vehicle control systems. In addition, to identify degrading performance in a sensor and predict the time at which a fault will occur, a novel predictive algorithm is proposed. The effectiveness and accuracy of the architecture in detecting and identifying multiple faults as well as the accuracy of the proposed predictive fault detection algorithm are verified through a MATLAB/IPG CarMaker co-simulation platform. The results present detection and identification accuracies of 94.94% and 97.01%, respectively, as well as a prediction accuracy of 75.35%.

Keywords Fault detection · Fault identification · Fault prediction · Support Vector Machine

Abbreviations

ANN	Artificial neural network
CV	Cross-validation
FDII	Fault detection, isolation & identification
IVHM	Integrated vehicle health management
<i>k</i> -NN	<i>k</i> -nearest neighbours
SVM	Support Vector Machine

1 Introduction

With the advent of autonomous vehicles, the number of sensors within cars has been significantly increased. As such applications become more reliant on a diverse set of sensors to operate, the fault tolerance and health awareness of these systems become more important. Current vehicles on the road rely on periodic maintenance and repair regimes to keep the vehicle in a safe and reliable operating range.

However, periodic maintenance and repair regimes are not ideal for autonomous vehicles as there is no mechanism to monitor the health of the various parts in-between service intervals. It is noted that an unpredicted fault in sensors can spread through each stage of autonomous driving functions, generating significant errors in the output of these functions (e.g. decision making, stability control systems, et cetera), which in turn reduce vehicle safety and may have severe consequences [1, 2].

The main approaches to developing fault tolerance can be grouped into hardware redundancy [3], model-based methods [4–13], and data-driven methods [14–26]. Hardware redundancy has been the traditional approach that relies on comparing the outputs of two identical sensors. When they disagree by a certain margin, a fault is present. This evidently can be a costly approach, which is often not economical in mass-production automotive applications. Model-based methods however typically operate by calculating residuals between measured parameters and a model of the system [3], as opposed to a physical second sensor. Model-based fault detection has seen attention in the automotive sector for systems such as navigation [4, 5], electrical power systems [6–8] and yaw moment control [9–13], where safety considerations are particularly critical. As more

✉ Saber Fallah
s.fallah@surrey.ac.uk

¹ Connected Autonomous Vehicles Lab (CAV-Lab),
Department of Mechanical Engineering Sciences, University
of Surrey, Surrey GU2 7XH, UK

sensors and systems are used, however, more complex models must be developed and implemented, which is a major weakness of model-based methods. Typically, the model-based techniques suffer from a lack of robustness against un-modelled dynamics, model uncertainties and sensor noise [9]. Additionally, existing techniques are limited to specific known faults for identification purposes [11], which for fault identification can be a weakness if not all fault types are known.

Data-driven methods instead use large amounts of data to model the vehicle and its fault conditions by creating models that capture the behaviour of complex systems which would be difficult to describe with a model-based method. The need for large amounts of data for data-driven methods is a fundamental concern. However, recent advancements in connected vehicles, distributed computing platforms and a possibility to store and analyse data arising from different sensors open up avenues for the development of intelligent fault detection, isolation and identification (FDII) systems for automotive applications using data-driven methods [2]. The effectiveness of data-driven learning techniques for fault detection and maintenance has already been demonstrated in the aircraft industry [27]. These previous bodies of work indicate that extending the fault detection and fault tolerance of autonomous vehicles to the realm of data-driven techniques can potentially provide great benefits in terms of vehicle safety.

The main approaches of data-driven methods for this purpose include Artificial Neural Networks (ANN) [14–16], k -Nearest Neighbours (k -NN) [17, 18] and Support Vector Machines (SVM) [15–26]. ANNs map input nodes to output nodes using hidden layers, with the mapping being used to determine whether new data are faulty, while k -NNs classify the data using the classification of k neighbours to determine whether new data are faulty or not [28]. Typically, both ANNs and k -NNs suffer from poor generalisation performance [16, 18]. SVMs are a statistical learning method that can identify a separating hyperplane between faulty and normal data. Using this plane to determine whether new data are faulty or not, they can provide greater generalisation capability compared to ANNs and k -NNs.

Current data-driven FDII architectures can be categorised as single fault detection for a single sensor [29], single fault detection of multi-sensor systems [30–32] and multi fault detection and identification of a single sensor [20–22]. Despite significant recent interest in data-driven approaches for fault detection and identification applications, there are still gaps remaining in the current research with regards to multi-fault identification, detection of faults in multi-sensor systems as well as condition-based predictive fault detection. As real systems comprise multiple sensors, the possibility of two or more sensors being in a fault condition is probable, so overcoming this gap is important. Similarly,

the lack of condition-based predictive fault detection is an ongoing challenge as the automotive sector moves towards a condition-based maintenance philosophy. Data-driven methods have been applied previously to predict faults based on usage data such as running hours and lifetime [2, 33], but usage-based data usually require a correlation between usage and fault types, which is not strictly true for sensor faults. By using condition-based data to detect faults, all types of faults that manifest themselves in the measured signal can be identified.

Considering the gaps in multi-sensor, multi-fault detection and identification, and condition-based fault prediction, this paper proposes a novel multi-sensor, multi-fault resilient architecture to bridge the current research gaps. The second contribution is the implementation of a novel predictive algorithm that provides an estimate of the time at which degrading performance of a sensor would be classed as a fault for the controller. The considerations made here for practical implementations would represent a step forward in addressing the current research gaps for bringing fault detection, identification and prediction into use in autonomous vehicle applications.

This paper is organised as follows. The system architecture and algorithm are proposed in Sect. 2. Section 3 develops the SVM models to implement this architecture. The simulation results of these algorithms are then presented and discussed in Sect. 4. Finally, Sect. 5 concludes the work and proposes future work outside of the scope of this paper.

2 Proposed Architecture and Algorithms

The topic explored in this paper is broadly described as Integrated Vehicle Health Monitoring (IVHM). The benefits of IVHM are improved vehicle safety, more accurate health awareness and reductions in through-life costs [34, 35]. The proposed FDII system of this paper comprises both current vehicle state awareness (fault detection and identification) and prediction to determine future vehicle health. An adequate understanding of the vehicle's current health state would allow for intelligent vehicles to reconfigure the available healthy systems to compensate for failures, or for maintenance to be directed quickly to the faulty components. Predictive fault detection allows for the condition of the vehicle systems to be predicted at a future point in time. The proposed predictive FDII system provides the opportunity for control systems to reconfigure themselves in advance of a sensor failure or fault.

The overall architecture of the proposed FDII system is shown in Fig. 1. It is comprised of Detection, Isolation, Identification and Prediction modules.

The first step, detection, takes a number of independent sensor signals, applies a signal feature extraction function

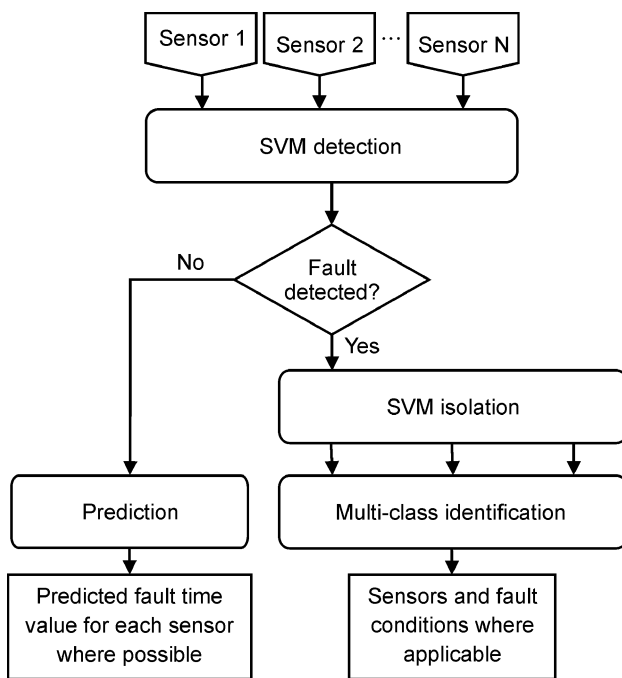


Fig. 1 Proposed FDI system architecture

for each signal, and finally combines them into a single sample. These features are used in a single class SVM model to identify whether a fault is present in the sample. By combining the signals in a single SVM model, the online performance can be improved, as opposed to using a detection SVM model for each sensor in the system. If a fault is detected, the samples are processed further. If a fault is not detected by the detection module, the sample is passed to the prediction module to establish if the sensor signal indicates degrading performance preceding a future fault.

In order to establish which of the sensors is faulty within a combined signal sample, the signal is split into segments pertaining to each sensor, which are then individually tested against its respective single-class SVM model. Each of the SVM models is trained on only the sensor it pertains to as each sensor has different characteristic features as previously discussed. This architecture is therefore resilient to multi-sensor faults. If no fault is detected in any of the sensors, the detection is assumed to be a false positive. Another advantage to this approach is the performance benefits to real-time operation. As these isolation SVM models only run when a fault is detected, the overall architecture can benefit from improved speed.

The SVM model is re-used in the prediction module to determine for non-faulty samples if the signal for that sensor is degrading. Samples within the prediction module are tested against the SVM model and a health score computed.

Statistical information of this health score is then used to make predictions as to when the sample may become faulty.

Finally, each sensor found to have a fault is then tested against a multi-class SVM model to identify which type of fault is occurring. No healthy data are passed to the identification module which means that only a single identification model is needed.

The primary outputs of the system are the sensors that are faulty for that time-sample and the fault condition identified for each faulty sensor. The secondary output is (for those sensors not in fault condition but have degraded performance) the time at which the fault is predicted to occur.

3 Development of Models

The architecture proposed has been developed further in order to identify the theoretical considerations of SVM techniques, faults and fault prediction. The models and decisions made to implement the architecture above are explored here.

3.1 SVM Models

This paper primarily analyses SVM techniques, however another common data-driven method, ANNs, has been considered by other authors. SVM techniques have been chosen for this paper given the weaknesses found with ANNs in other work. Hybrid approaches were not chosen despite the improved performance offered, as the focus of this paper is on establishing a robust and repeatable FDI approach and architecture. By choosing a single base technique and using it throughout this paper, the proposed architecture can remain flexible to future developments, allowing the models to be augmented with more sophisticated methods later.

Artificial Neural Networks are an approach that map input source nodes through hidden layers of computation nodes to output nodes [36]. This method transforms input data between layers using an activation function. The training builds connections between nodes while minimising an error function comparing computed outputs to the actual outputs. Typically, this method suffers from poor generalisation performance compared to SVMs due to the optimisation function used to train them [16]. As generalisation performance is desirable for the varied scenarios that automotive sensors are subjected to, SVMs are therefore preferable.

SVM models identify a set of support vectors within a dataset which define a separating hyperplane [36]. New data points are then classified based on which side of this hyperplane they lie. This method relies on statistical learning and can offer higher accuracies and generalisation capability due to the distinctive features that define the hyperplane [16].

For datasets that cannot be linearly separated by a hyperplane, a kernel function can be applied to transform

the feature-space into higher dimensions that are linearly separable. There are several kernel functions [37] that have been used by other authors, the most common of which are Linear [38], Polynomial [22] and Radial-Basis Function [17, 20, 26]. Generally, Radial-Basis Function was found to give more satisfactory performance for the classification of multiple sensor faults and as such has been used for this paper.

One of the most significant efforts involved with this method is identifying features of a sensor signal that are unique to a fault condition. Much work has been completed on different time-domain and frequency-domain features, illustrating the impact on classification accuracy that selection can have. It has been found that time-domain features are sufficiently detailed to give high classification accuracies for fault detection, and as such frequency-domain features are unnecessary [21, 22]. Using more features yields diminishing returns on classification accuracy, while increasing the computational time for a model to be trained and tested. As such, a compromise needs to be reached between the required classification accuracy and the computational power available [21].

Recently, modifications to the traditional SVM method have been proposed [17, 20, 22] as well as hybrid data fusion methods [15, 25, 26] which can offer higher classification accuracies in certain cases. For this paper, k -fold cross-validation (CV) has been used with the SVM models to reduce the generalisation error [22], as overfitting is a concern with this method. The dataset is divided into k sets and the model is trained k times, each time leaving out one of the sets and testing with the set which is left out. Statistical features of this process are then used to calculate the generalisation error.

3.2 Faults

There are many different types of faults that can affect sensors, and therefore not all types have been modelled exhaustively for this paper. Five distinct definitions have been modelled for this work which represents likely types of faults that can occur for the sensors in an automotive controller. Fault types can be intermittent, which are explored in this paper as it is expected to be a more difficult case to detect for SVM models.

These types of sensor faults are defined as: drift, hard-over, erratic, spike and stuck faults, shown in Fig. 2. The following definitions are used for these faults [21]:

Drift faults The sensor value deviates linearly over time from the true value.

Hard-over faults The sensor value increases to the saturation point for a short period of time.

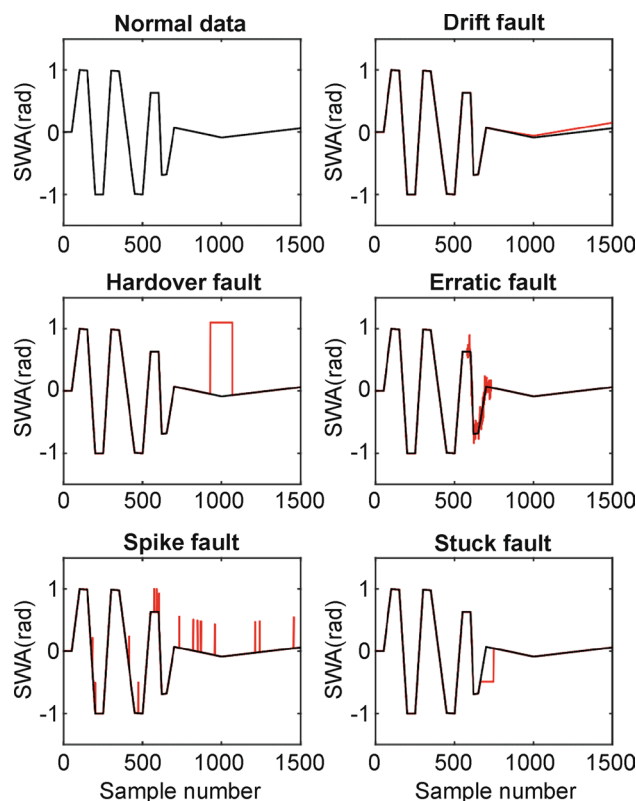


Fig. 2 Faults injected into normal signal samples of a steering wheel angle sensor

Erratic faults The sensor value varies about the true value. The magnitude of this variance can increase over time.

Spike faults The sensor value is significantly above the true value for a single point. The density of spike faults within the signal can increase over time.

Stuck faults The sensor value remains at a fixed value for a short period of time.

3.3 Feature Selection and Extraction

Feature selection is important in order to capture sufficient defining features about a signal, specifically the characteristics of a fault present in a signal sample. In this paper, this step is performed using a feature extraction algorithm applied to each signal sample. The features are then combined in a single combined feature vector for use in the SVM models.

Figure 3 shows how the signals are combined in the feature extraction algorithm. No weighting or normalisation is performed at this stage to accommodate the differences in sensor signal characteristics. Instead, the SVM model standardises the combined feature vectors with respect to

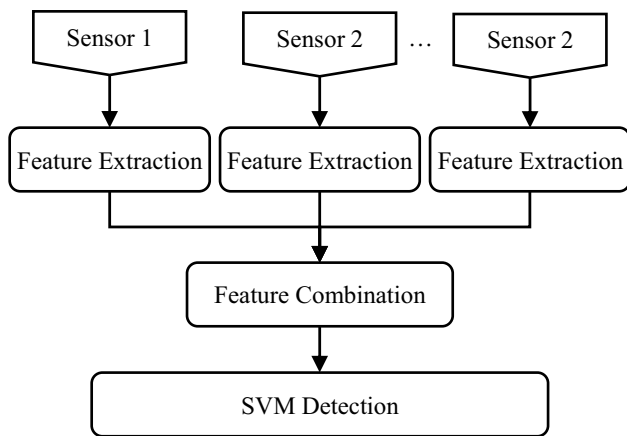


Fig. 3 Combined feature vector creation

Table 1 Time-domain feature definitions used for N data points in a sample x [21]

$f_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$	$f_{SRA} = \left(\frac{1}{N} \sum_{i=1}^N \sqrt{ x_i } \right)^2$
$f_{KV} = \frac{1}{N} \sum_{i=1}^N \left(\frac{x_i - \mu_i}{\sigma} \right)^4$	$f_{SV} = \sum_{i=1}^N \left(\frac{x_i - \mu_i}{\sigma} \right)^3$
$f_{PPV} = \max(x) - \min(x)$	$f_{CF} = \frac{\max(x)}{f_{RMS}}$
$f_{IF} = \frac{\max(x)}{\frac{1}{N} \sum_{i=1}^N x_i }$	$f_{MF} = \frac{\max(x)}{f_{SRA}}$
$f_{SF} = \frac{\max(x)}{\sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}}$	$f_{KF} = \frac{\frac{1}{N} \sum_{i=1}^N \left(\frac{x_i - \mu_i}{\sigma} \right)^4}{\left(\frac{1}{N} \sum_{i=1}^N x_i^2 \right)^2}$

both extracted feature and sensor. This ensures that the differences in sensor signal characteristics do not impact the performance of the system.

As has been previously explored [21], increasing the number of time-domain features which describe the signal yields diminishing increases in classification accuracy. Therefore, a feature space utilising time-domain features is used to train and test the SVM algorithms, to give sufficient detection accuracy whilst simplifying the system. Frequency-domain features were not considered as the time-domain features are sufficient to differentiate between faulty and normal signals.

Table 1 shows the feature definitions used for the fault types as defined in Ref. [21]. These features are Root Mean Square (f_{RMS}), Square Root of Amplitude (f_{SRA}), Kurtosis Value (f_{KV}), Skewness Value (f_{SV}), Peak-to-Peak Value (f_{PPV}), Crest Factor (f_{CF}), Impulse Factor (f_{IF}), Marginal Factor (f_{MF}), Shape Factor (f_{SF}) and Kurtosis Factor (f_{KF}).

Algorithm 1. Feature Extraction Pseudocode
function Feature Extraction(data)

Sampling Rate = N

for $i = 1 : \text{end}(\text{data})$
 $X_i = \text{data}(i : I + N)$

Feature 1 = $f_1(X_i)$

...

Feature $k = f_k(X_i)$

if $X_i = \text{faulty}$

Label = 1

Fault Type = 1,2,..., N

else

Label = 0

Fault Type = 0

End

Output(Features, Label, Fault Type)

The pseudocode for this implementation is shown in Algorithm 1. The output of this function is a matrix of features, where each row is a single k -datapoint sample, with each column being the value of the feature defined in Table 1.

3.4 Fault Detection

Fault detection is one of the primary objectives of this FDII system. As any number of sensors can be in a fault condition, the sensor signals are combined and an SVM model uses the features extracted from all sensor signals to determine if any of the sensors are in a fault condition. This is beneficial as it reduces the complexity of the architecture into a single SVM model for detection, rather than independent detection models for each sensor. As the number of sensors used increases, the benefits of this approach become more apparent.

The main objective of fault detection is to determine whether the overall functional system is faulty or safe to use. The criticality of the fault cannot be investigated at this step as the detection model can only report that there is a fault present. In order to reconfigure the system or direct maintenance efforts, the signal is split into respective sensor signals. Each signal is then passed to the fault isolation module to determine which specific sensors are in a fault condition.

3.5 Fault Isolation

Fault isolation is necessary to identify which sensors are reporting faults. It is possible to perform independent fault detection on each sensor instead of fault detection on the entire combined signal feed, however this adds computational load to the overall FDII system as explored in Sect. 1. Independent fault detection is also evidently less feasible when the number of sensors is increased. For this paper, the isolation module only tests the sensor if a fault is detected, which reduces computational load.

Each sensor type has an SVM model, as each sensor has different characteristic ‘normal’ signal types, even after standardisation. These models are tested against new samples which are determined by the detection model to be faulty. This approach allows for any number of sensors to be declared faulty for a particular sample, which ensures the architecture is multi-sensor, multi-fault tolerant. Any sensor-specific samples which are determined to be faulty are then passed to the fault identification module to determine which type of fault is occurring for that particular sensor.

3.6 Fault Identification

Once a fault has been detected and isolated, it then needs to be identified as a particular fault type. This is important as different fault types have different impacts on vehicle reconfigurability. For example, a drift fault may not be serious enough to cause a significant reduction in vehicle safety, whereas a hard-over fault would be.

To determine which type of fault is present, the signal for each sensor is used in a multi-class SVM model, with one class for each fault type. There is an additional implied class (normal data), though it is never trained or tested for. It is assumed that all data which reach the identification stage should be faulty data. This is opposed to other methods [21, 22], which use the multiclass model to both detect and identify a fault in a single sensor. This is expected to give slightly higher identification accuracies as faulty data cannot be misclassified as normal data.

The benefit of performing fault identification on each of the three sensors rather than on the combined signal is that it reduces the number of classes needed to represent every

possible combination of sensor faults. As multi-sensor failure tolerance is the aim of this paper, this approach is more appropriate.

As the normal signal is not tested for, only a single multi-class identification model is trained with the data from all three sensors, and then each sensor is tested individually using the same model for each sensor.

3.7 Fault Prediction

One area of interest is the use of SVM models to predict faults. Within this paper, the methods of Ref. [17] have been modified and extended to provide a time position at which the fault is predicted to occur. This brings many benefits to automotive applications, especially in autonomous vehicles which lack a human driver who can actively detect faults in the vehicle systems. If degrading performance is detected and a fault predicted to occur before the end of a driving scenario, a higher-level vehicle control system could reconfigure itself [3] or adjust its mission profile to continue meeting its safety requirements.

Within the detection SVM model developed for this paper, the separating hyperplane or ‘decision-boundary’ can be tested with new samples to identify a fault that degrades sensor performance over time. The distance between this decision boundary and the new sample can be used to identify how ‘healthy’ and how ‘faulty’ the signal is. The smaller the value, the closer to the decision boundary the sample lies. Negative values represent members of the opposing class.

In the implementation of Ref. [17], the distance was normalised using a logistic function in order to clip the significant variations of healthy/faulty samples and to focus on the transition between the two. The values of the health score would therefore be bounded between 0 and 1, with the threshold of normality lying at 0.5. Although trend analysis was suggested to be possible in Ref. [17], for this paper it was not found to be sufficient to allow for predictions to be made. As such, a 5-point trailing moving average is also applied to smooth the transition and help consistent predictions be made. The pseudocode for this is shown in Algorithm 2.

Algorithm 2. Health Score and Prediction Pseudocode

function Health Score(Detection SVM Model, Signal)

Features = Feature Extraction(Signal)

[Label, X] = Detection SVM Model(Features)

$$\text{Health Score} = \frac{1}{1+e^X}$$

Average Score = Moving Average(Health Score)

if Average Score(i) < Threshold $m = i$ $n = m + 2$ Trend = Linear Line(Average Score($m : n$))**end****while** (gradient(Trend) < Set Point && Trials < N) $N = N + 1$ $m = m + 1$ $n = m + 2$ Trend = Linear Line(Average Score($m : n$))**End****Output**(Health Score, Trend)

For fault prediction, three types of faults are applicable: drift, erratic and spike. This paper develops this prediction method on erratic as it is thought to be the more serious and most difficult to detect given its highly variable nature.

4 Simulation Results and Discussion

In order to test the proposed architecture and algorithms, a proprietary controller for autonomous vehicles proposed in Refs. [39, 40] has been used. The controller consists of three input signals: driving torque, brake pressure and steering wheel angle. Two output signals have been considered to analyse the effect of faults on vehicle control performance: linear velocity (v_x) and yaw moment (M_z). All simulations have been run using a co-simulation MATLAB/IPG Car-Maker environment.

The dataset generated to train and test the proposed FDII system comprised of 3 million data points, with each point containing the signal amplitude for each sensor in a 10 ms interval. This represents 500 min of simulated data sampled at 100 Hz. The ratio between training and testing data has been set at 80% training to 20% testing. This has been set high deliberately to ensure a sufficient variety of training scenarios are captured. This will help reduce the generalisation error discussed in Sect. 3.1.

Testing began with a small trial dataset of 300,000 data points in order to tune and test without significant processing time. The accuracies calculated are the number of correct classifications over the total number of classifications. The steps taken to simulate this architecture are detailed in Algorithm 3.

Algorithm 3. Methodology for Reproduction

-
1. Number and type of sensors and controller selected
 2. SVM models produced for each module defined in the overall FDII Architecture
 3. Initial trial dataset of 300,000 data points simulated in IPG-CarMaker for tuning the SVM models
 4. Faults injected into the trial dataset
 5. SVM model trained using trial dataset with 10-fold Cross-Validation
 6. Simulation dataset of 3,000,000 data-points simulated
 7. Faults injected into the simulation dataset
 8. Optimal parameters selected for each model based on earlier trials
 9. SVM models trained on 80% of the simulation dataset
 10. Architecture tested using the remaining 20% of simulation dataset
-

4.1 Data Collection

Three simulated driving scenarios are used to build each dataset using the IPG CarMaker software. One typical of urban driving, one of highway driving and one of parking. The driving scenarios have been duplicated, representing multiple runs of each scenario to fill the specified number of data points. All scenarios are then combined to fill the datasets. The selection of driving scenarios is intended to capture the variety of sensor outputs that may be seen in normal conditions, and to explore the effect these scenarios have on SVM accuracies. This is important in order to ensure the model does not predict normal driving data as faulty.

The trial dataset was generated by limiting the duration of the simulations such that the total amount of data was approximately 300,000 data points (50 min). The simulation dataset is generated with the same simulation setup, but with the duration limited such that the total amount of data was approximately 3,000,000 (500 min). Faults are artificially induced using MATLAB in both datasets with identical fault parameters.

The artificially generated faults were injected into the three sensor signals, namely driving torque, brake pressure,

Table 2 Fault parameters

Fault	Amplitude	Gradient	Max length (data points)
Drift	N/A	$\pm \frac{1}{\text{Scenario Length}}$	N/A
Hard-over	$1.1 \times \text{signal FSO}$	N/A	$0.5 \times \text{scenario length}$
Erratic	$\pm 0.2 \times \text{signal FSO}$	N/A	$0.4 \times \text{scenario length}$
Spike	$+0.5 \times \text{signal FSO}$	N/A	N/A
Stuck	N/A	N/A	100

and steering wheel angle. The injection of each fault incorporates pseudo-randomness to simulate the variability of each fault condition. Intermittency of fault conditions has been included for hard-over, erratic, spike and stuck to simulate real-world fault conditions. Half of the scenarios had a fault injected into them to provide sufficient training data.

For the driving scenarios which have had a fault injected, the fault is introduced after the 25th percentile of one or more of the three sensor signals. The type and location of fault are introduced using a uniform distribution to get an equal distribution of faults. Normal distributions have been used to simulate the amplitude of faults where applicable. An error of 20% of Full-Scale Output (FSO) has been defined as faulty for the erratic and drift faults.

The fault parameters shown in Table 2 have been selected to give a suitable representation of real-world faults for the simulated data. Uniform distributions are used to incorporate randomness into drift fault gradient, hard-over fault length, erratic fault length and stuck fault length. The erratic fault amplitude uses a normal distribution with a standard deviation of 0.2.

4.2 Detection Performance

For detection, a 30-feature single class SVM model has been trained, with 10 features extracted per sensor. As can be seen from Fig. 4, the prediction closely matches the actual faults that were injected into the simulated dataset, confirming the accuracy of the detection SVM model. The accuracies of the model for the detection of different injected faults are presented in Table 3.

As the table shows, the model detects each fault with a minimum accuracy of 88.42% for drift faults and a maximum accuracy of 99.97% for hard-over faults. However, these accuracies are likely higher than would be observed using real sensor data as the data in the normal class does not contain any noise. The assumption made during this simulation is that data pre-treatment is sufficient to minimise the noise in the signals. Should noise be present or inadequately filtered out by signal pre-processing, it is expected that the performance of the detection module will degrade. The

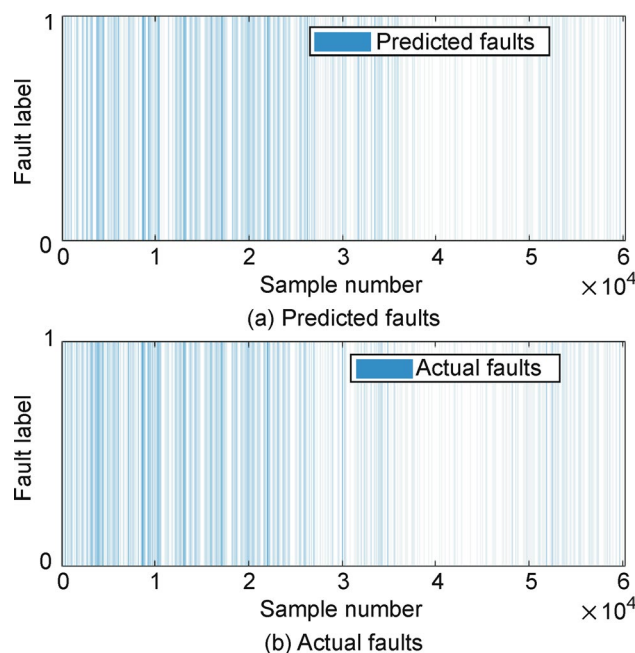


Fig. 4 Detection classification results from 30-feature SVM model with **a** predicted faults and **b** actual faults

detection performance for each fault type would be driven mainly by the signal features and how close they are to the signal features of noisy data. For example, the signal features of erratic faults and noise are similar, so the performance for erratic detection is expected to be reduced.

The accuracy for the spike fault class was found to be higher than that reported elsewhere in Refs. [21, 22]. This has been investigated and the potential reason would be the smaller sample size of 10 data points compared to the other approaches using 1000 data points per sample. This means that the spike faults can represent a larger proportion of the sample for a given spike fault density, and therefore have more distinctive features. No discernible accuracy reductions are observed as a result of the smaller sample size in other fault classes.

Drift faults typically are harder to detect at their initiation, as they can be typical of normal sensor behaviour. This is shown here with drift faults being the hardest fault types to detect, however, the accuracy is still appreciably high.

Stuck faults show a lower accuracy as one characteristic feature of stuck faults is the absence of noise presented after fault initiation. Due to the lack of noise in the simulations, stuck faults are not as easily distinguished from normal data. This is a weakness of simulating sensor data. It does however highlight that for sensors that lack significant characteristic noise, stuck faults may be difficult to detect.

Hard-over faults are noticeably easy to detect as they have significantly abnormal features. With a signal gradient approaching infinity, no normal driving scenario could

Table 3 Detection accuracies for the trial and full datasets (%)

	Overall accuracy	Drift	Hard-over	Erratic	Spike	Stuck
Trial	94.54	81.73	99.49	92.76	99.75	96.80
Simulation	94.94	88.42	99.97	89.66	99.96	96.54

produce this type of sensor output for the sensors used in this paper. This is useful as hard-over faults have a very high potential to reduce driving safety.

In addition to the three input sensors, two system outputs, linear velocity (V_x) and yaw moment (M_z), are considered to gauge whether it is more effective to detect faults using the system outputs rather than sensor inputs. From a qualitative investigation of the outputs, the features appear less prominent in the system outputs due to the number of sensors used and the fixed parameters in the controller. As such, a reduction in accuracy was expected. The approach to detection was the same as with the input sensors, but with only 20 features used (10 for each signal).

While not explicitly explored, it is understood that an approach focused on the controller outputs would suffer from variable correlation, where the isolation of a fault to a specific component becomes more difficult due to the effect a fault in a single component has over the controller. By using an architecture focused on the sensors themselves, the impact of this is minimised.

Figure 5 shows the performance of the model when trained and tested against the two outputs of the system. Although it represents the faults injected relatively adequately, the overall accuracy is 79.33%, lower than previously with the input sensors. Most significantly, the hard-over accuracy is found to be 75.92%, compared with 100% for the previous model. This illustrates how distinctive features of the faults become subtler and harder to recognise with an SVM model after being propagated through a controller. This is an area where further work is needed to establish whether the losses in detection accuracies by using controller system outputs can be mitigated by feature selection, or whether this is a characteristic of performing fault detection on controller system outputs.

4.3 Isolation Performance

In the isolation module, the faults are detected for each sensor independently in order to isolate where the fault is occurring. Three 10-feature single class SVM models have been trained, one for each sensor type. An overall detection accuracy of 97.42% is observed, slightly higher than the overall accuracy of the 30-feature SVM model. The accuracies are presented in Table 4.

The most significant finding from this simulation is that each sensor has varying accuracies both overall and for each fault type. This appears to be mitigated by the inclusion of

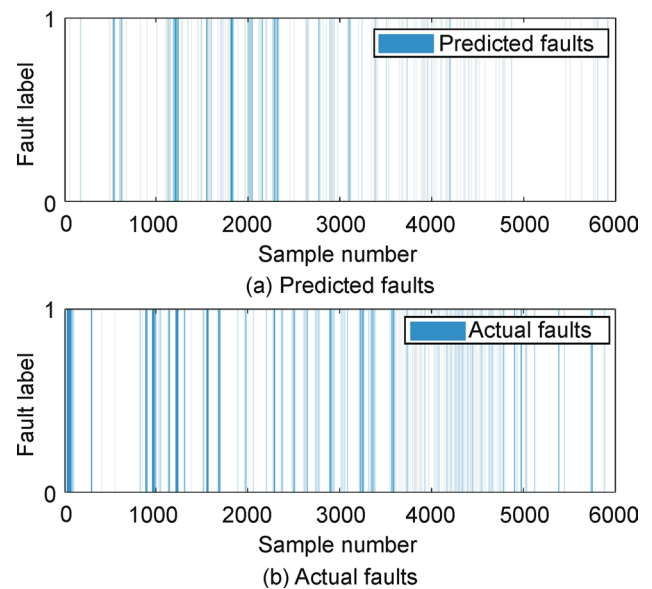


Fig. 5 Detection classification results for 20-feature SVM model of outputs v_x and M_z with **a** predicted faults and **b** actual faults

more data but remains an issue. This is mainly due to the fact that the normal data types for the sensors are unique to that type of sensor. For instance, the data for the steering wheel angle sensor typically exhibit lower gradients than that of the driving torque, even after normalisation in the SVM model. This explains the difficulty that the SVM model has in detecting a fault in the steering wheel angle sensor for drift faults, as a drift fault has very similar time-domain features to that of a normal signal for that particular sensor.

4.4 Identification Performance

For identification, the sensor which has been isolated in the previous step is tested to determine which type of fault is present. The dataset has been used to train a 10-feature, 5-class SVM model, with each class representing one of the fault types previously modelled. As found with isolation, identification accuracy varies depending on which sensor is tested. This is shown from the results in Fig. 6, with an overall accuracy of 97.01% observed. The breakdown of accuracies is presented in Table 5.

The identification accuracies vary between 99.10% and 73.49%, with brake pressure having the highest accuracies. Driving torque is poorer with several of the lowest accuracies, especially for spike fault identification. Spike faults

Table 4 Isolation detection accuracies for each sensor type, Driving Torque (DT), Braking Pressure (BP), Steering Wheel Angle (SWA) (%)

	Overall accuracy	Drift	Hard-over	Erratic	Spike	Stuck
<i>Trial</i>						
DT	97.80	93.18	99.49	96.46	99.75	98.99
BP	97.89	95.79	99.75	96.46	99.66	97.05
SWA	95.66	80.72	99.75	98.57	100.00	97.39
<i>Simulation</i>						
DT	97.83	93.59	99.91	96.28	99.97	99.35
BP	97.80	96.98	99.98	98.26	98.57	95.20
SWA	96.62	86.25	100.00	97.41	99.98	99.33

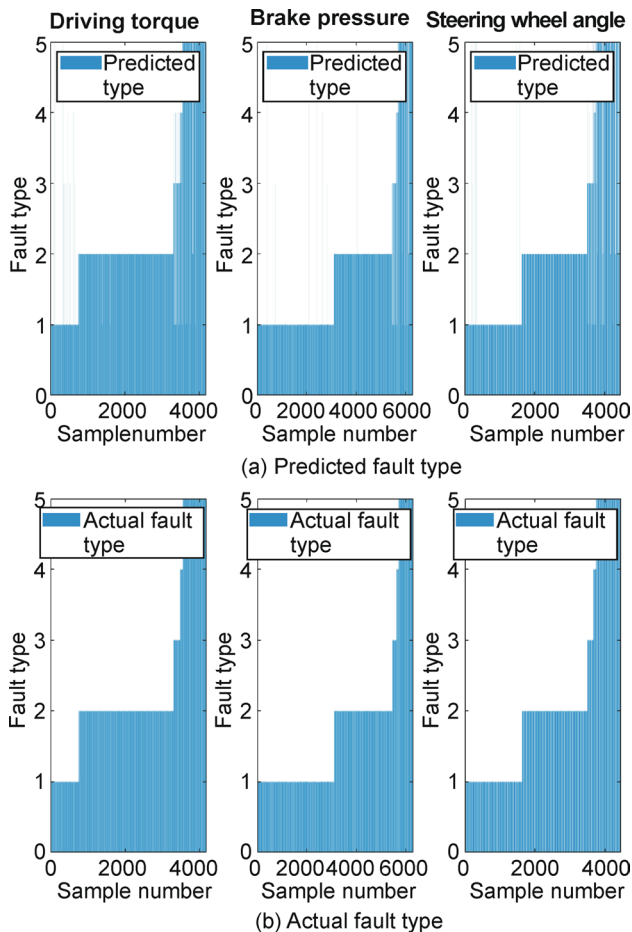


Fig. 6 Identification results for all three sensors combined with **a** predicted fault type and **b** actual fault type

are the hardest faults to identify for all sensors, often being misclassified as erratic faults and vice versa. Increasing the amount of training data improved the accuracy for these two fault classes. The reason can be that the spike and erratic faults modelled are too similar to be distinguished easily with the chosen feature set, and more data is needed to accurately predict these two faults. Therefore, future work may be in identifying either a dataset or a feature set that provides

Table 5 Identification accuracy (%)

	Overall accuracy	DT	BP	SWA
<i>Trial</i>				
Drift	93.82	81.58	96.10	95.59
Hard-over	85.47	91.77	100.00	77.73
Erratic	54.41	44.44	66.67	55.00
Spike	90.91	75.00	100.00	100.00
Stuck	93.82	81.58	96.10	95.59
<i>Simulation</i>				
Drift	97.23	93.97	98.59	96.15
Hard-over	98.65	99.10	98.99	97.63
Erratic	82.17	75.65	89.61	82.84
Spike	79.09	73.49	81.01	84.48
Stuck	97.23	93.97	98.59	96.15

a better compromise between the identification accuracy for all fault types.

Remark 1 The models for the detection, isolation and identification are tuned in order to reduce misclassification error. The two main parameters of the models which have been tuned are the box constraint and kernel scale parameters. Tuning is performed on a smaller trial dataset (300,000 data points) and then applied to the larger simulation dataset as the process is very time-consuming.

To minimise the CV loss of the model, Bayesian Optimisation was performed with 30 function evaluations using the ‘OptimizeHyperparameters’ function within MATLAB. The default tenfold was used with the CV method for tuning. The trial datasets were run using the default box constraint and kernel scale values to validate the improvement in CV loss.

Table 6 shows the results of the optimisation function for the 30-feature detection model. As can be seen, a reduction in the CV loss is observed. It is worth noting here that a compromise may be achieved between training time and CV-loss, however, the models have been optimised for accuracy. To illustrate the impact of training time, for the trial data the

Table 6 Tuning parameters-trial detection SVM

	CV loss/%	C	σ	Training time/s
Trial	5.01	1	1	8.8
	4.83	171.39	2.0301	76.1

training time increases by a factor of 10 for the indicated 0.18% reduction in CV-loss.

Remark 2 The performance of the SVM models is found to be satisfactory during the testing component of the simulations. As the simulations are able to run faster than the time period of the dataset tested, the architecture can be run in real-time. However, the training component involves large variability depending on the parameters set. Training times of up to 30 min are observed on this relatively small dataset. The workstation used for these simulations has a 3.7 GHz Hex-core CPU and 16 GB of Memory. Memory is not found to be a limiting factor, however, the training regularly uses 100% of a single CPU core. Parallel processing may improve training time but is not explored in this paper.

4.5 Fault Prediction

To analyse the accuracy of prediction, the prediction algorithm has been tested for 10,000 degrading erratic faults. Erratic faults were selected as they were perceived as the most difficult fault type to predict. The overall prediction accuracy is found to be 75.35%. The prediction accuracy here is expressed as the difference between the predicted fault time and the actual fault time for a 20% error, expressed as a percentage of the time in which the fault initiates and degrades to an actual fault.

For this paper, a maximum of 100 predictions are attempted to ensure that the fault is always predicted even if it is not accurately predicted in the first trials. Multiply the attempted amount if the gradient of prediction is found to be too shallow for a prediction to be made. This has been done to only detect faults that are predicted to be degrading fast enough to be of interest.

Table 7 shows an extract of the testing data for an erratic data sample. As can be seen, the distance between the new sample and the hyperplane reduces as the fault error increases, however not linearly. An average normalised score of 0.7 and above was found to exclusively represent healthy samples, therefore this was set as the threshold below which a degrading fault is predicted to be occurring.

Improvements of accuracy can be gained by using a larger number of health scores to predict, however, this reduces the time between the prediction being made and the fault occurring. The parameter choices made for this paper give a good compromise between how close the prediction is to the

Table 7 Sample of hyperplane distances of an erratic sample for approximate error values

Approx. (~) sample error/%	Distance to hyperplane	Normalised health score
0	0.99	0.71
~5	0.97	0.71
~10	0.87	0.70
~15	0.64	0.65
~20	-0.43	0.39

actual fault time, retaining as much time before the prediction occurs as possible.

Figure 7 shows an example of a fault prediction. While this example shows excellent prediction accuracy, the variance was found to be an issue as shown in Fig. 8. Although 24.63% of predictions have accuracies over 90% and give a close approximation to the fault initiation time, 57.44% of predictions had accuracies between 60% and 90%. This is found to be due to a combination of factors, the most significant being the inaccuracies in the detection model for degrading faults. This causes the gradient of the degradation to be too variable to accurately predict. While training the models using degrading faults is explored, it is found that the initiation of degrading faults was classified as normal data and therefore caused the same variable gradient. Additionally, the variability of the erratic fault was a significant problem to try and accommodate, as the signal would often appear to recover as can be seen in Fig. 7. This causes the linear detection to be insufficient to capture the full behaviour of the degradation, demonstrating the challenges that real sensor faults pose to fault prediction.

There were a small number of predictions that failed even after 100 trials, which were included as having 0% accuracy. This is likely unavoidable, although could be improved by tuning the trailing moving-average used. Centre-point moving-averages were considered for this paper and show a small improvement in accuracy; however this reduced the time between the prediction being made and the fault initiating, reducing the value of the fault prediction in the first place. Trailing moving-averages are therefore thought to be a more appropriate method of smoothing the health scores for prediction. As can be seen from Fig. 8, only 0.73% of predictions completely failed, and 67.68% of predictions showed accuracies in the 70–100% range.

There is also a possibility for refining a prediction over a longer time period, which was not explored in this paper. Moreover, it is thought that while the time-period for prediction used in this approach is short, the same approach can apply to fault types that degrade over much longer time-periods.

Fig. 7 Linear fault prediction with 98.46% accuracy

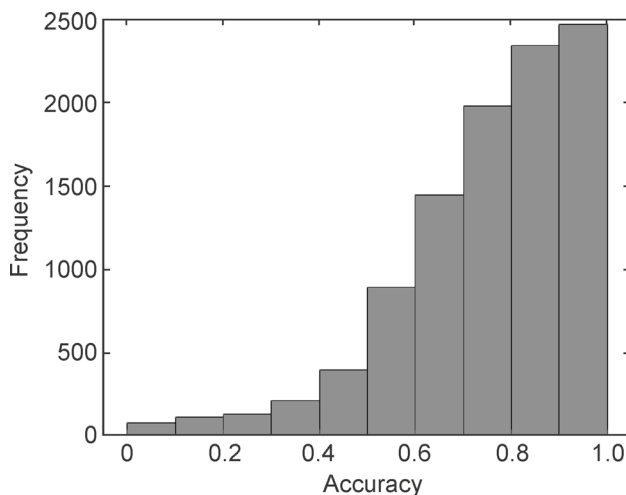
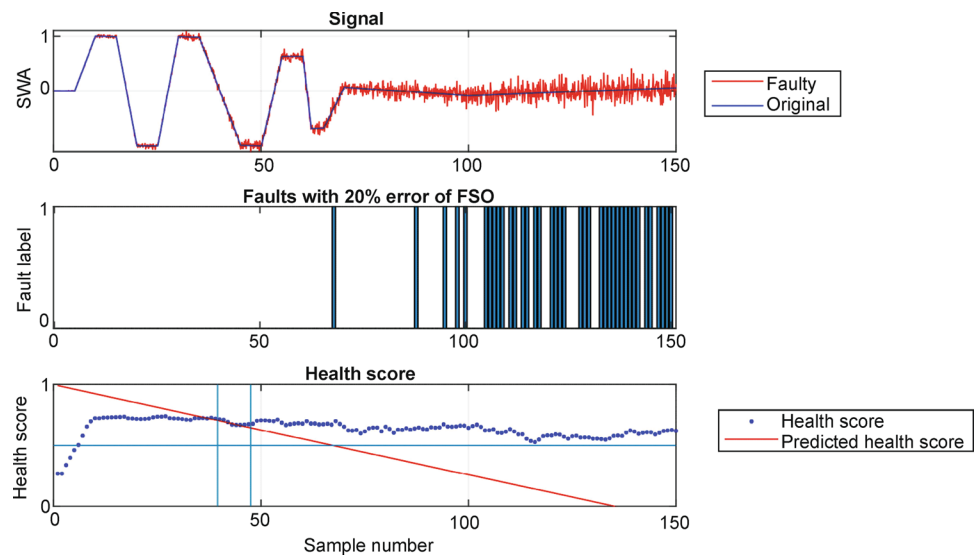


Fig. 8 Histogram showing frequency distribution of prediction accuracies

The predictive algorithm was also validated against the two other applicable fault types for degrading faults: drift and spike faults. For spike faults, the definition in Sect. 3.2 is used, where the density of the spike faults increases over time. The parameters tuned with erratic faults were re-used to identify how general this algorithm is. For drift faults, the accuracy is reduced to 56.97%. For spike faults, the algorithm failed to make a prediction for 74.77% of faults, with an overall accuracy of 14.86%.

For comparison, other more established methods of prediction can achieve higher accuracies such as 97.1% for a method using ANN models [41]. It is worth noting however that the reduction in accuracy in the method used in this paper can be balanced against the relative benefits. Fault

prediction using this algorithm does not rely on more symptomatic fault prediction such as vibration analysis or physical parameter observations, and so can be more appropriate for sensor and control applications.

Further work for this approach would mainly involve identifying coefficients to the logistic function to improve the normalisation, or the exploration of different normalisation functions, allowing for a more linear trend in the health scores. Different approaches to the prediction algorithm should also be explored which would allow for a more generalised algorithm that can predict multiple fault types. The relatively simple trailing moving average could also be replaced with a more sophisticated method, as well as non-linear predictions.

5 Discussion and Future Work

Key observations have been made on the application of SVM models in multi-sensor control systems, most notably showing that different sensors have different detection and identification accuracies when exposed to particular fault types such as drift faults. This observation will be a major consideration for SVM models applied to different multi-sensor control systems subject to a wide variety of faults. This paper does not consider the full subset of faults however, such future work could be extended to include a broader definition of faults based on a wider survey of likely sensor system faults. One key weakness identified is that simulation of data without consideration of noise can affect the validity of the accuracies obtained. Notably, erratic faults may be much harder to detect for a signal should noise be present. On the other hand, stuck faults should see an improvement in accuracy should noise be considered. Overall, the accuracies

achieved by the FDII system proposed here show high accuracies of 94.94%, 97.42%, 97.01% for detection, isolation and identification, respectively. Other authors have found similar overall classification accuracies (88.13%) with the same fault set tested here, with similar methods [21].

For this paper, only a single data-driven method approach was used. As observed in the literature, generally hybrid approaches can offer significant improvements in detection and identification accuracy given the relative benefits of each method. For this approach, it is most likely that future work in replacing the detection and isolation component with an ANN model will provide significant improvements in accuracy, as the lowest accuracies observed are for the detection module. The isolation module can also be greatly simplified by the ability of the ANN to isolate faults to particular sensors.

For detection and isolation, the selection of parameters to improve generalisation performance allows new types of faults to be distinguished from healthy signal samples. However, it is worth noting that for identification, the multi-class SVM model used is not resilient to new fault types as the model has a single class for each fault type and will mis-identify new types. Other approaches in future work should be explored to improve this weakness.

Training of the SVM models is completed offline with no consideration given to updating this over time. Online adaptive training can provide improved classification accuracy, especially where parameters are subject to drift over time due to changing conditions. Vehicles operate in an inherently unpredictable environment, so the ability to adapt to changes in the environment would be a significantly beneficial focus for future work.

This paper also proposes a novel algorithm for fault prediction in sensing systems of autonomous vehicles. The predictive algorithm uses condition-based data to give the best possible estimate of the time at which a degrading fault will emerge in a sensor. As opposed to time or usage-based approaches used by other authors, this approach is applicable to all types of fault regardless of their relationship with time or usage-based data. The prediction algorithm shows great promise, with a relatively simple implementation resulting in a prediction accuracy of 75.35%. However, unsatisfactory performance is observed when the algorithm developed for erratic faults is applied to drift and spike faults. This highlights an area for improvement where future work can improve this prediction accuracy by using more sophisticated trend analysis and data analytics. Developing a more generalised prediction scheme that can be applied to several different fault types is a key goal of future work. In addition, it is noted that the simple trend analysis results in lower performance compared to other methods [41].

Finally, while not considered in detail here, as IVHM systems typically allow for reconfiguration, the outputs from

the FDII system could be used to inform a higher-level vehicle control system that can reconfigure the available healthy components to continue the current vehicle mission. Analytical redundancy for this purpose has already been examined in model-based methods. While the models may not be sufficient to detect faults, they could be used to recover some level of vehicle functionality using available healthy sensors. Future work on comprehensive IVHM systems that meet more levels of autonomy will evidently need to examine a wider variety of systems and provide more valuable outputs such as this to the overall vehicle systems, in order for commercial applications to be successful.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Shen, T., Kilic, A., Reuss, H.: A new system diagnostic method for powertrain of automated electric vehicles. In Proceedings of the 2018 21st international conference on intelligent transportation systems (ITSC), Maui, 4–7 Nov. 2018
2. Dhall, R., Solanki, V.: An IoT based predictive connected car maintenance approach. *Int. J. Interact. Multimedia Artif. Intel.* **4**, 16 (2017)
3. Hwang, I., Kim, S., Kim, Y., Seah, C.: A survey of fault detection, isolation, and reconfiguration methods. *IEEE Trans. Control Syst. Technol.* **18**, 636–653 (2010)
4. Huang, W., Su, X.: Design of a fault detection and isolation system for intelligent vehicle navigation system. *Int. J. Navig. Observ.* **2015**, 1–19 (2015)
5. Yang, L., Li, Y., Wu, Y., Rizos, C.: An enhanced MEMS-INS/GNSS integrated system with fault detection and exclusion capability for land vehicle navigation in urban areas. *GPS Solut.* **18**, 593–603 (2013)
6. Alavi, S., Fekriasl, S., Niyakan, S., Saif, M.: Fault detection and isolation in batteries power electronics and chargers. *J. Energy Storage.* **25**, 100807 (2019)
7. Scacchioli, A., Rizzoni, G., Salman, M., Li, W., Onori, S., Zhang, X.: Model-based diagnosis of an automotive electric power generation and storage system. *IEEE Trans. Syst. Man Cybern. Syst.* **44**, 72–85 (2014)
8. Wang, Y., Tian, J., Chen, Z., Liu, X.: Model based insulation fault diagnosis for lithium-ion battery pack in electric vehicles. *Measurement* **131**, 443–451 (2019)
9. Pous, N., Gingras, D., Gruyer, D.: Intelligent vehicle embedded sensors fault detection and isolation using analytical redundancy

- and nonlinear transformations. *J. Control Sci. Eng.* **2017**, 1–10 (2017)
10. Hernández-Alcántara, D., Tudón-Martínez, J., Amézquita-Brooks, L., Vivas-López, C., Morales-Menéndez, R.: Modeling, diagnosis and estimation of actuator faults in vehicle suspensions. *Control Eng. Pract.* **49**, 173–186 (2016)
 11. You, S., Cho, Y., Hahn, J.: Model-based fault detection and isolation in automotive yaw moment control system. *Int. J. Automot. Technol.* **18**, 405–416 (2017)
 12. Rahimi, A., Dev Kumar, K., Alighanbari, H.: Fault detection and isolation of control moment gyros for satellite attitude control subsystem. *Mech. Syst. Signal Process.* **135**, 106419 (2020)
 13. Wikström, A.: Yaw rate and lateral acceleration sensor plausibilisation in an active front steering vehicle. Dissertation, Linköping University (2006).
 14. Nieto González, J.: Vehicle fault detection and diagnosis combining an AANN and multiclass SVM. *Int. J. Interact. Des. Manuf. (IJIDeM)* **12**, 273–279 (2017)
 15. Hu, J., Huang, T., Zhou, J., Zeng, J.: Electronic systems diagnosis fault in gasoline engines based on multi-information fusion. *Sensors*. **18**, 2917 (2018)
 16. Niu, X., Yang, C., Wang, H., Wang, Y.: Investigation of ANN and SVM based on limited samples for performance and emissions prediction of a CRDI-assisted marine diesel engine. *Appl. Therm. Eng.* **111**, 1353–1364 (2017)
 17. Diez-Olivan, A., Pagan, J., Khoa, N., Sanz, R., Sierra, B.: Kernel-based support vector machines for automated health status assessment in monitoring sensor data. *Int. J. Adv. Manuf. Technol.* **95**, 327–340 (2017)
 18. Andre, A., Beltrame, E., Wainer, J.: A combination of support vector machine and k-nearest neighbors for machine fault detection. *Appl. Artif. Intel.* **27**, 36–49 (2013)
 19. Banerjee, T., Das, S.: Multi-sensor data fusion using support vector machine for motor fault detection. *Inf. Sci.* **217**, 96–107 (2012)
 20. Aoudi, W., Barbar, A.: Support vector machines: a distance-based approach to multi-class classification. In: Proceedings of the 2016 IEEE international multidisciplinary conference on engineering technology (IMCET), Beirut, 2–4 Nov. 2016
 21. Jan, S., Lee, Y., Shin, J., Koo, I.: Sensor fault classification based on support vector machine and statistical time-domain features. *IEEE Access*. **5**, 8682–8690 (2017)
 22. Jan, S., Koo, I.: Sensor faults detection and classification using SVM with diverse features. In: Proceedings of the 2017 international conference on information and communication technology convergence (ICTC), Jeju Island, 8–10, Oct. 2017
 23. Nalepa, J., Kawulok, M.: Selecting training sets for support vector machines: a review. *Artif. Intel. Rev.* **52**, 857–900 (2018)
 24. Wang, F., He, K., Liu, Y., Li, L., Hu, X.: Research on the selection of kernel function in SVM based facial expression recognition. In: Proceedings of the 2013 IEEE 8th conference on industrial electronics and applications (ICIEA), Melbourne, 19–21 June 2013
 25. Realpe, M., Vintimilla, B., Vlacic, L.: Sensor fault detection and diagnosis for autonomous vehicles. *MATEC Web of Conferences*. **30**, 04003 (2015)
 26. Anaissi, A., Khoa, N., Rakotoarivelo, T., Alamdari, M., Wang, Y.: Adaptive online one-class support vector machines with applications in structural health monitoring. *ACM Trans. Intel. Syst. Technol.* **9**, 1–20 (2018)
 27. Atyabi, A., Zadeh, S.M., Nefti-Meziani, S.: Current advancements on autonomous mission planning and management systems: an AUV and UAV perspective. *Ann. Rev. Control.* **46**, 196–215 (2018)
 28. Shakhnarovich, G., Darrell, T., Indyk, P., Dietterich, T., Jordan, M.: *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. MIT Press, Cambridge (2006)
 29. Ren, L., Xu, Z., Yan, X.: Single-sensor incipient fault detection. *IEEE Sens. J.* **11**, 2102–2107 (2011)
 30. El-Koujok, M., Benammar, M., Meskin, N., Al-Naemi, M., Langari, R.: Multiple sensor fault diagnosis by evolving data-driven approach. *Inf. Sci.* **259**, 346–358 (2014)
 31. El-Koujok, M., Benammar, M., Meskin, N., Al-Naemi, M., Langari, R.: Multiple sensor fault diagnosis for non-linear and dynamic system by evolving approach. Proceedings of the IEEE 2012 prognostics and system health management conference (PHM-2012 Beijing). (2012).
 32. Liu, J.: Data-driven fault detection and isolation for multimode processes. *Asia Pac. J. Chem. Eng.* **6**, 470–483 (2011)
 33. Prytz, R.: machine learning methods for vehicle predictive maintenance using off-board and on-board data. Dissertation, Halmstad University (2014)
 34. Dong, Y., Xia, T., Fang, X., Zhang, Z., Xi, L.: Prognostic and health management for adaptive manufacturing systems with online sensors and flexible structures. *Comput. Ind. Eng.* **133**, 57–68 (2019)
 35. Prajapati, A., Roy, B., Prasad, R.: A state of art review of integrated vehicle health management system. In: Proceedings of the 2018 3rd international conference for convergence in technology (I2CT), Pune, 27–30 Oct. 2018
 36. Davies, E.: *Computer vision-principles, algorithms, applications, learning*. Elsevier, London (2018)
 37. Cuesta, H.: *Practical data analysis*. Packt Publishing, Birmingham (2013)
 38. Li, L., Yu, S., Xiao, W., Li, Y., Huang, L., Zheng, X., Zhou, S., Yang, H.: Sequence-based identification of recombination spots using pseudo nucleic acid representation and recursive feature extraction by linear kernel SVM. *BMC Bioinf.* **15**, 340 (2014)
 39. Fallah, S., Khajepour, A., Fidan, B., Chen, S., Litkouhi, B.: Vehicle optimal torque vectoring using state-derivative feedback and linear matrix inequality. *IEEE Trans. Vehic. Technol.* **62**, 1540–1552 (2013)
 40. Taherian, S., Montanaro, U., Dixit, S., Fallah, S.: Integrated trajectory planning and torque vectoring for autonomous emergency collision avoidance. In: Proceedings of the 2019 IEEE intelligent transportation systems conference (ITSC), Auckland, 27–30 Oct. 2019
 41. Sahu, A., Palei, S.: Fault prediction of drag system using artificial neural network for prevention of dragline failure. *Eng. Fail. Anal.* **113**, 104542 (2020)