IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 24, 2021, accepted January 20, 2022, date of publication February 7, 2022, date of current version February 11, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3149324

Parametric Circuit Fault Diagnosis Through Oscillation-Based Testing in Analogue Circuits: Statistical and Deep Learning Approaches

JACOB B. CLOETE[1], TINUS STANDER[2], (Senior Member, IEEE), AND DANIEL N. WILKE[1]

[1]Centre for Asset Integrity Management, University of Pretoria, Pretoria 0086, South Africa
[2]Carl and Emily Fuchs Institute for Microelectronics, Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0086, South Africa

Corresponding author: Jacob B. Cloete (u17019363@tuks.co.za)

ABSTRACT Oscillation-based testing of analogue electronic filters removes the need for test signal synthesis. Parametric faults in the presence of normal component tolerance variation are challenging to detect and diagnose. This study demonstrates the suitability of statistical learning and deep learning techniques for parametric fault diagnosis and detection by investigating several time-series classification techniques. Traditional harmonic analysis is used as a baseline for an in-depth comparison. Eight standard classification techniques are applied and compared. Deep learning approaches, which classify the time-series signals directly, are shown to benefit from the oscillator start-up region for feature extraction. Global average pooling in the convolutional neural networks (CNN) allows for Class Activation Maps (CAM). This enables interpreting the time-series signal's discriminative regions and confirming the importance of the start-up oscillation signal. The deep learning approach outperforms the harmonic analysis approach on simulated data by an average of 11.77% in classification accuracy for the three parametric fault magnitudes considered in this work.

INDEX TERMS Artificial intelligence, deep learning, fault diagnosis, machine learning, oscillation-based diagnosis, statistical learning.

I. INTRODUCTION

A significant proportion of faults in microelectronic circuits (around 80%) occur in the analogue parts of the circuit [1], necessitating extensive production testing. Circuit testing can be broadly generalised into specification-driven tests and fault-driven tests, of which the latter is more common in integrated circuit (IC) testing [2]. Fault detection may be extended to fault diagnosis, where the specific fault locations are identified, but this is complicated in analog circuits by non-linear behaviour, acceptable tolerances in components values, inefficient fault models, inaccessible nodes and measurement uncertainty [1].

In oscillation-based testing (OBT) [3]–[7], the circuit under test (CUT) is reconfigured (either using electronic switches [8], or by an external tester [5]) into an oscillator by connecting the circuit's input and output through an appropriate feedback network to create an astable feedback loop and monitoring the resulting oscillation data. This approach removes the need for a reliable or calibrated test signal, reducing test duration, complexity and cost [1]. By comparing the oscillation signal of the circuit under test to that of a fault-free circuit and various predetermined faulty circuits stored in a fault dictionary (usually generated by circuit simulation *a priori*), the faulty circuits can be identified [1].

OBT may be used to detect both catastrophic (e.g. broken bond-wires) and parametric (e.g. non-uniform oxide thickness) faults, though parametric faults are far more challenging to detect [8].

By extending the OBT procedure to diagnosing the location and extent of the fault, oscillation-based diagnosis (OBD) is possible, as first described in [9], where a simulation-generated fault dictionary approach is applied using the frequency of the first harmonic and the total harmonic distortion of the oscillating signal as features. In [10], a feed-forward artificial neural network (ANN), with the basic multi-layer perceptron (MLP) structure, is used to diagnose faults. A supervised learning procedure for capturing

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

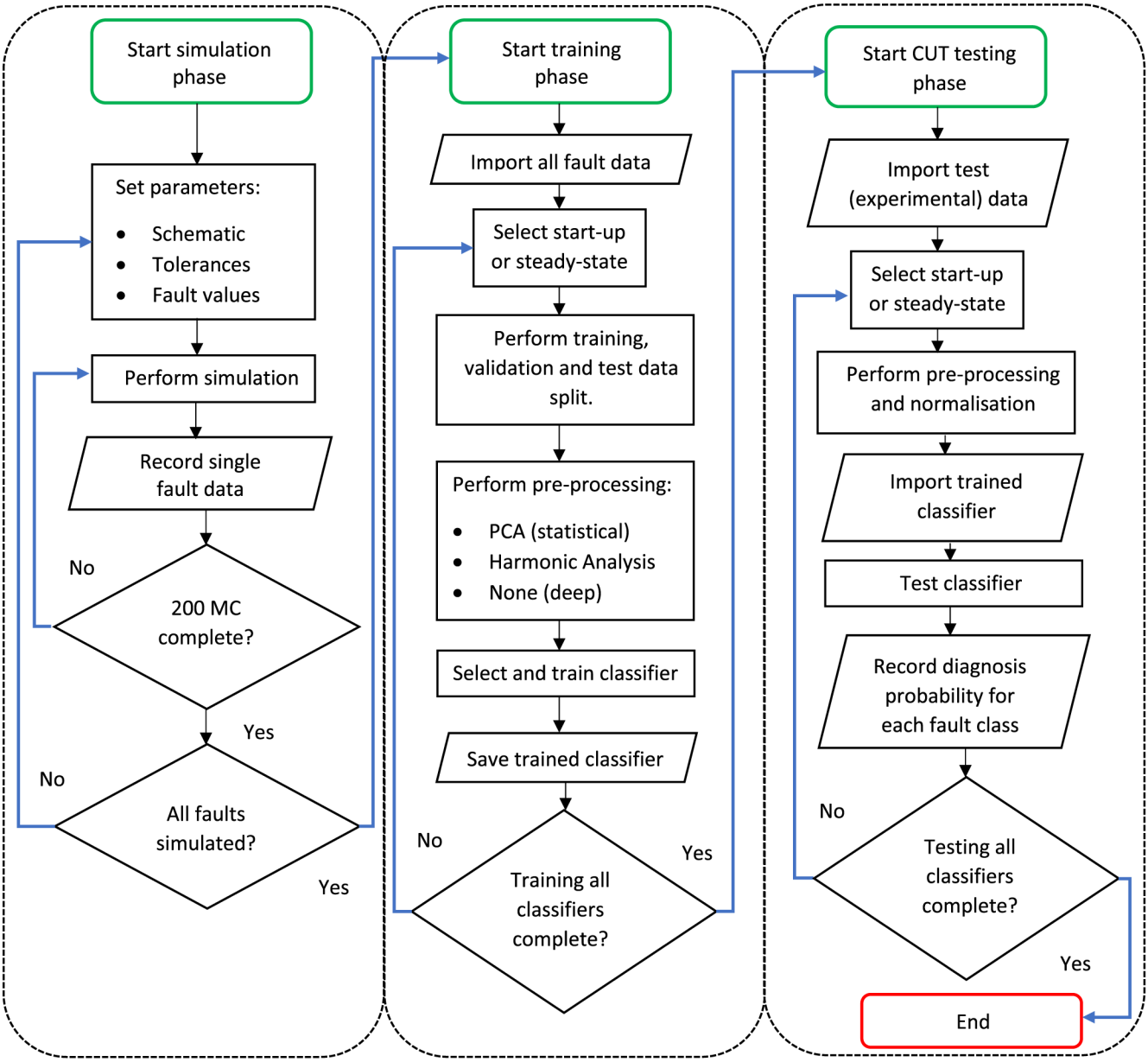


FIGURE 1. General procedure diagram.

the fault dictionary is first applied to OBD in [11], with the feature extraction extended to the first four harmonics in [2]. White noise, with an amplitude of 1% of the output signals, was added to the simulated signals to verify robustness, though the link between this introduction and probability component value variation is not presented. Other prior literature on OBD includes membership function diagnosis in [12], with the state-of-the-art work in [13] incorporating simultaneous Monte Carlo statistical variation for non-faulty component values.

This study advances the state-of-the-art in OBD of parametric faults in three ways:

1) To verify the efficacy of ANN-based OBD under more realistic circumstances, all non-faulty passive component tolerances are included in the simulation for the first time in this work, by means of a generalised Monte Carlo sampling. This stands in contrast to current techniques that use ANN's to diagnose faults [2], [11], where the assumption is that all non-faulty passive components are fixed at their nominal value. This is a far more accurate representation of real-world test conditions [14], [15] than has been considered previously.
2) Component faults are, for the first time, diagnosed directly from time-series data. This time-series

diagnosis is performed both with a statistical learning approach [16], and a deep learning approach [17]. Both approaches are compared to the harmonics analysis currently used in OBD [2], [11], [12] and class activation maps (CAMs) used to highlight which data are more crucial in correct classification. This is a significant finding in selecting of suitable data, potentially reducing the resources required in collecting data that does not contribute to classification accuracy.

3) The time-series data is applied to evaluate the start-up region of the oscillation signal as a classification feature for the first time, which is impossible with the current state-of-the-art harmonics approach.

This paper is organised as follows:

In Section II, the general procedure of simulating the CUT, training the classifiers and classifying the parametric faults is presented. Section III outlines the training and evaluation methodology for the harmonics approach while the training and evaluation methodology for the statistical learning approach is presented in Section IV. Section V provides a brief literature review of time-series deep learning, then presents the training and evaluation methodology for the deep learning approach. Section VI summarises the results and presents a comparison of all three approaches. Conclusions are offered in Section VII.

II. PROCEDURE

The general procedure, showing the simulation, training, and classification steps, is presented in Fig. 1. This figure has three phases, namely the simulation phase (described in Sections II.A and II.B), training phase (described in Sections II.C and II.D) and CUT testing phase (described in Section II.D). The process starts with SPICE netlists of the CUT in OBT mode (i.e. feedback applied), with parametric variations defined for both nominal variation and injected faults. As the generality of the trained model is significant in practical engineering [18], this work relies on Monte Carlo analysis in circuit simulation to estimate measurement data in volume production. This approach has been proven reliable and is widely used in OBT [14] and elsewhere in semiconductor engineering [15].

A. CIRCUIT UNDER TEST

This work could be applied to various fields in semiconductor manufacturing. High-Pass filters commonly used in baseband circuits are used for communications [19] and at the input of digital-to-analog converters for spectrum shaping [20]. Active filters are also commonly selected as circuits under test in other OBT and OBD research [2], [11], and are once again selected here to make the work better comparable to prior works.

The simulation procedure requires a schematic of the CUT, along with the component tolerances and fault values. The CUT is a second-order high-pass Butterworth filter with −3 dB cut-off at 1 kHz and high-frequency gain of 1.075.

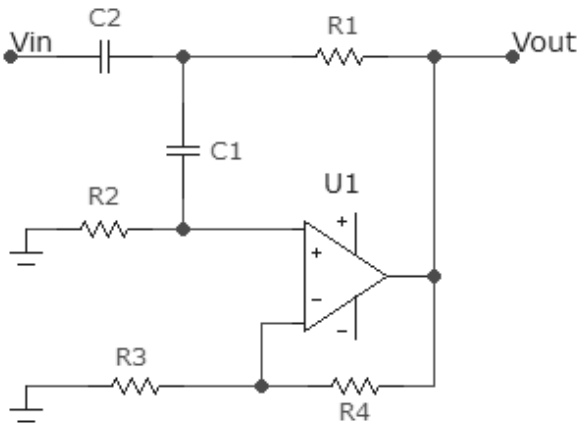


FIGURE 2. Device under test in normal operation mode.

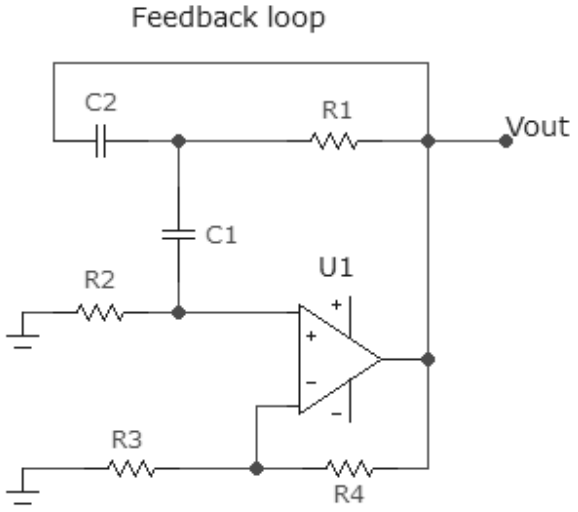


FIGURE 3. Device under test in oscillation mode.

The CUT is shown in normal operation in Fig. 2. The CUT is reconfigured, either using electronic switches [8], or by an external tester [5], into an oscillator by connecting the circuit's input and output thereby creating a redundant positive feedback loop as shown in oscillation setup in Fig. 3. All components have tolerances of 5%, which marks the 3σ standard deviation. The Op-Amp used for the simulation is the LT1013, and the nominal values used for each component are shown in Table 1.

B. SIMULATION PROCEDURE

The filter is simulated in Micro-Cap 12 [21] in both the normal (to verify the performance characteristics) and oscillation configurations. The components are allocated the specified tolerances. For each class (fault), the Monte Carlo simulations produce 200 oscillation time-series datasets. For example, when fault D1A is simulated, the R1 component value is changed to 12 kΩ and all non-faulty components are assigned

TABLE 1. General parametric fault setup.

Defect label	Component	Nominal value	Fault value
BASE	-	-	-
D1A	R1	10kΩ	10kΩ + (α × 10kΩ)
D1B	R1	10kΩ	10kΩ - (α × 10kΩ)
D2A	R2	10kΩ	10kΩ + (α × 10kΩ)
D2B	R2	10kΩ	10kΩ - (α × 10kΩ)
D3A	R3	10kΩ	10kΩ + (α × 10kΩ)
D3B	R3	10kΩ	10kΩ - (α × 10kΩ)
D4A	R4	750Ω	750Ω + (α × 750Ω)
D4B	R4	750Ω	750Ω - (α × 750Ω)
D5A	C1	22nF	22nF + (α × 22nF)
D5B	C1	22nF	22nF - (α × 22nF)
D6A	C2	22nF	22nF + (α × 22nF)
D6B	C2	22nF	22nF - (α × 22nF)

*The α indicates the fault size (0.2, 0.5, or 0.9)

random values according to the Gaussian distributions of 5% (3σ) to account for component tolerances. The general parametric fault setup is shown in Table 1. The first 5 ms of the signal is recorded at a 0.5 μs timestep, producing 10 000 datapoints for each Monte Carlo dataset. The resulting 2 MHz sample rate is significantly higher than the Nyquist rate of the 5th harmonic of the fundamental oscillating tone, which is usually around 100 kHz.

C. FEATURE EXTRACTION

The classification performance when using the start-up oscillating signal is compared to the steady-state of the signal for non-harmonic approaches. For the start-up case, the first 100 μs of the signal is used for classification. When considering the steady-state case, a peak-finding algorithm is used to determine the amplitude of each oscillation. When the difference in amplitude between two consecutive peaks is less than 1%, the signal is considered steady-state. Three consecutive peaks are then used for the steady-state time-series. Different feature extraction and pre-processing methods are used for each approach (harmonic, statistical or deep learning). In harmonic analysis the magnitudes of the spectral components of the harmonic oscillator are used. Statistical learning uses principal component analysis (PCA) while for the deep learning, the time-series data is used directly. These pre-processing steps are discussed in Section III.A, Section IV.A and Section V.A.

D. TRAINING AND DIAGNOSIS

The data is randomly split into training, validation, and test sets, with ratios as described in Sections III, IV and V. After this, the required pre-processing and regularisation steps are performed on set. In OBD, the primary interest is in having a model (e.g. statistical or deep learning) or strategy (e.g. signal processing) that outputs classes or labels (e.g. parametric fault labels) when fed time-series data or harmonic data. The following three definitions are applied in this process:

Definition 1 Formally, let (X, y) be a training instance given T observations $X = [X_{(1)}, X_{(2)}, X_{(3)} \ldots, X_{(T)}]$, which is the time-series of features extracted of each parametric fault.

Definition 2 In addition, a discrete class variable y which takes k possible values $y = [y(1), y(2), y(3) \ldots, y(k)]$, are the parametric fault labels.

Definition 3 The task of classifying time-series data consists of learning a classifier on training dataset. A training dataset S is a set of n training instances: $S = [(X_{(1)}, y_{(1)}), (X_{(2)}, y_{(2)}), \ldots, (X_{(n)}, y_{(n)})]$ such that n < T. The classifier maps from the space of possible inputs X to a probability distribution over the classes / labels $P = [y_{(1)}, y_{(2)}, y_{(3)} \ldots, y_{(k)}]$. The fault with the maximum probability is the diagnosed fault. This result is recorded and by comparing the recorded result to the simulated fault, accuracy is determined as a percentage of correctly diagnosed faults.

E. LEARNING METHODOLOGIES

Before presenting the details each approach in this work, it is prudent to briefly review prior approaches. Previous OBD studies [2], [11], [12] all fall within the supervised category because labelled data can be generated during the simulation and is used to explicitly create a classification model by using the known input and output of each sample. This model can then be used to classify new unseen and unlabeled samples to perform OBD.

These classification approaches all require different levels of feature engineering. Signal processing requires the analyst to conduct all preprocessing and feature engineering. This is, the most traditional and prevalent fault testing approach. Statistical learning, on the other hand, only requires the analyst to control preprocessing and hyper-parameters of the feature engineered linear models and remains unexplored in OBD. Machine learning requires the analyst to control the pre-processing and hyper-parameters of feature engineered non-linear models. The OBD work in [2], [11] falls in this class.

The structure of the deep learning models [2], [17], [22], used in this study, are shown in Table 2, with filter length is indicated in brackets. All models use categorical cross-entropy as a loss function. The structure of the Inception model shown in Table 2 is extended as seen in [22]. The various other hyperparameters, such as learning rate, learning rate reduction parameters, the use of dropout, softmax layers at the end of the network, input and output layer size, etc. are adjusted between datasets, but otherwise follow the state-of-the-art as outlined in the source references above.

TABLE 2. General structure of deep learning models.

Model	Module 1	Module 2	Module 3	Activation	Optimzer
MLP	500	500	500	ReLU	Adamax
FCN	128 (8)	256 (5)	128 (3)	ReLU	Adamax
ResNet	64 (8,5,3)	128 (8,5,3)	128 (8,5,3)	ReLU	Adamax
Inception	64 (1,3,5)	96 (1,3,5)	128 (1,3,5)	ReLU	Adamax

III. HARMONIC ANALYSIS

This section aims to reproduce the harmonic analysis of [2], [11]. In the first rounds of analysis, static idealised values

of non-faulty components are maintained, with Monte Carlo variation (200 samples) applied only to the faulty component. Non-faulty component value variations are then introduced to demonstrate the shortcomings of using threshold detection techniques, used for parametric faults in [8], and the need to explore new techniques.

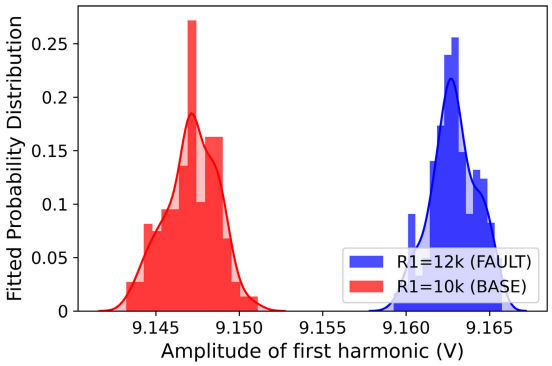


FIGURE 4. Probability density distribution when a tolerance is applied to only the faulty component.

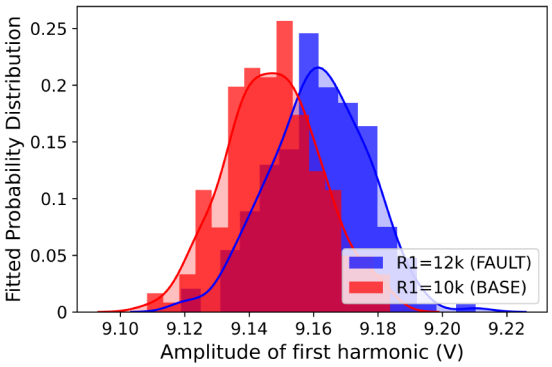


FIGURE 5. Probability density distribution when tolerances are applied to all components.

A. TRAINING METHODOLOGY

The simulation and training are performed according to the general procedure shown in Fig. 1. The data is mean-centered and then randomly split in training, validation and testing sets in a ratio of $\frac{1}{3} : \frac{1}{3} : \frac{1}{3}$. During pre-processing a Fourier transform is performed on the time-series of the steady-state oscillating signal. The amplitude and frequency of 5 harmonic components [2] are collected, reducing the number of dimensions significantly. An MLP with the same hyper-parameters and structure as seen in [2] is trained on the training and validation datasets.

B. EVALUATION METHODOLOGY

The difference in approaches is illustrated in Figs. 4 and 5 for a D1A injected fault. In Fig. 4, the approach in [2], [11], with the addition of component tolerances for only the faulty component, clearly separates the distributions in oscillating signal amplitude between faulty and non-faulty circuits. This enables simple threshold detection as a classifier. In the more realistic scenario followed in this work (Fig. 5), the faulty and non-faulty conditions are more challenging to distinguish. These overlapping probability density distributions, similar to those seen in [8], show the limitations of linear threshold techniques for parametric faults. The overall classification results when using the Harmonics approach is compared to the overall results when using the statistical learning and deep learning approaches in Section VI and V.

IV. STATISTICAL LEARNING

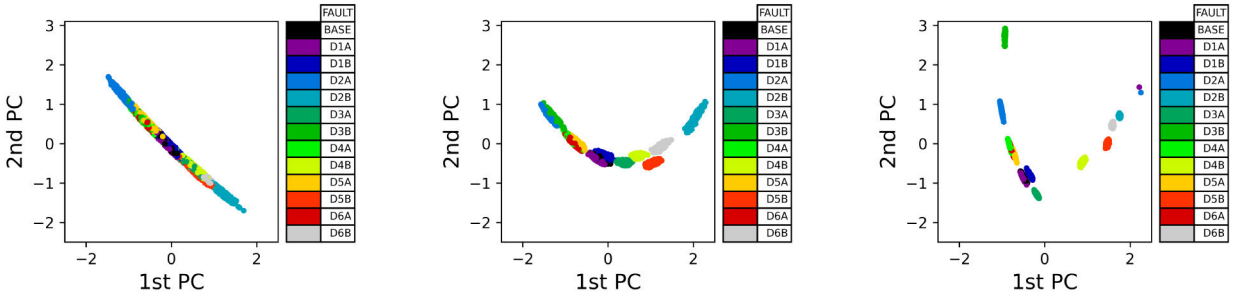
PCA is investigated as a preprocessing technique, instead of harmonic analysis, to explore its efficacy in classifying parametric faults when non-faulty component tolerances are included. PCA is a widely used dimensionality reduction technique used in machine learning, that creates an orthogonal projection of the dataset onto a lower dimensional linear space (principal subspace), in a manner that maximises the variance of the projected data [23]. This application of PCA falls under the class of latent variable models [24].

A. TRAINING METHODOLOGY

Both the start-up and steady-state data is mean-centered and then randomly split in training and testing sets in a ratio of $\frac{1}{2} : \frac{1}{2}$. Fig. 7 shows the variance explained by each principal component for the three datasets. It is clear that most of the variance can be explained by the first two principal components for all the datasets. Therefore, the first two principal components are selected as features for the latent space. The PCA latent space for each fault dataset is shown in Fig. 6. These three figures clearly show how the latent space becomes less entangled as the size of the faults increase. The two-dimensional latent space allows the use statistical learning classifiers to segment the 2D feature space which allows classification of new test data. This study considers 8 established statistical classification algorithms [25], namely, K-Nearest Neighbours (KNN), Radial Basis Function Support Vector Machines (RBF-SVM), Decision Trees, Random Forest, Adaboost with Random Forest as the base, Extremely Randomised Trees (ExtraTrees) [26], Naive Bayes and Quadratic Discriminant Analysis (QDA).

B. EVALUATION METHODOLOGY

The testing dataset is used to evaluate the performance of each classifier. As an example, the latent space representation of the best performing statistical learning algorithm, ExtraTrees, is presented in Fig. 8. In Fig. 8 (a) the principal components of the training data is shown. Fig. 8 (b) depicts the noiseless segmented latent space after training. In Fig. 8 (c) and Fig. 8 (d), noise is introduced as a class, to remove data points that would have a reconstruction loss larger than the loss required for the point to belong to a certain class, within the PCA latent space. In Fig. 8 (c), this noise class is introduced with a threshold by combining all the latent space regions with a class probability that is less than 0.95. In Fig. 8 (d)



(a) PCA for ±20% Faults. (b) PCA for ±50% Faults. (c) PCA for ±90% Faults.

FIGURE 6. Principal component analysis.

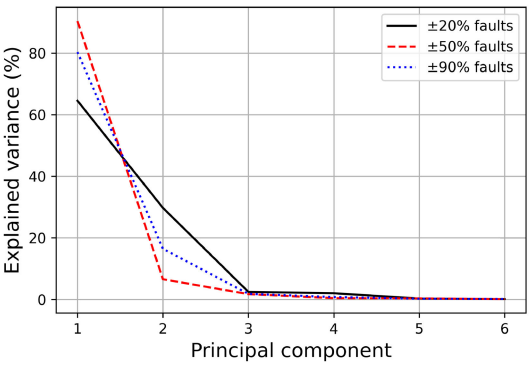


FIGURE 7. Variance explained by each principal component.

the noise class is introduced by adding uniform random data points into the latent space and labelling these data points as noise. When using the threshold noise class, the results for all the classifiers are presented with a box-and-whisker plot in Figs. 9 and 10. The classifiers have a mean testing accuracy that is around 10% higher on the steady-state data when compared to the start-up data. The KNN, SVM, Decision Tree, Random Forest, Adaboost, and ExtraTrees classifiers have higher average accuracies than the Naïve Bayes and QDA classifiers. However, all of these classifiers are outperformed by the best deep learning models.

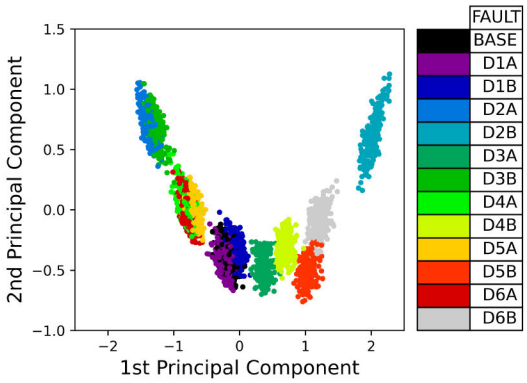
V. DEEP LEARNING

Deep learning extends machine learning by not requiring any pre-processing or feature engineering. This would enable a fully automated strategy and would be the first of its kind for classification in OBD using the time-series signal directly, subsequently allowing the use of start-up of the oscillation signal for the first time. The application of deep learning to time-series classification (TSC) [17] was driven by the successes of deep learning [27] in general and lately, detection, diagnostics and prognostics in physical asset management [16]. Deep Convolutional Neural Networks (CNNs) have achieved superior performance in supervised machine learning [28]. They have non-linear transformations that involve convolutions and max pooling which reduces computation time and complexity without losing the essence of the data. CNNs' performance depends on the size and quality of the training data. Nonetheless, state-of-the-art performance for TSC with the use of end-to-end CNN architecture has been demonstrated in [17] with the Fully Convolutional Network (FCN) and Residual Network (ResNet). The recent success of Inception-based networks [22] prompted their adaptation to TSC in [29]. The incomprehensible internal logic of the hidden layers in a deep neural network leads to uninterpretable black-box models. This can be mitigated with the use of a Class Activation Map (CAM) [30]. The second to last layer for all three convolutional networks is replaced with a Global Average Pooling, which drastically reduces the number of parameters and allows for using a CAM for these TSC networks [17], [31]. The CAM highlights the data points in the time-series data that contribute most to the classification, known as the discriminative regions.

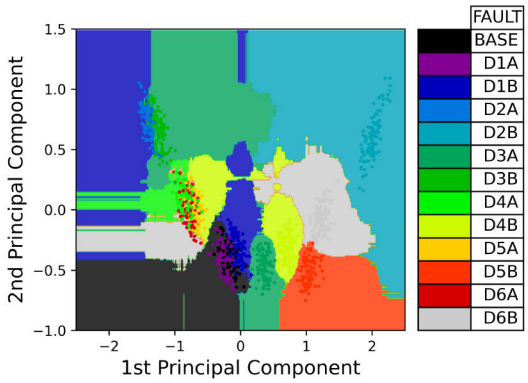
The deep learning approach does not require any feature extraction and classifies the time-series data directly using three convolutional networks, namely a Fully Convolutional Network (FCN) and Residual Network (ResNet) with the architecture outlined in [17], and an Inception-based network for time-series data, with the architecture outlined in [22]. The MLP [2] used in the harmonic analysis section is also used for comparison. The MLP network is adapted by increasing the input size to match that of the time-series data. Dropout is selected as a regularisation strategy, where some randomly selected neurons are deactivated in each training epoch [32].

A. TRAINING METHODOLOGY

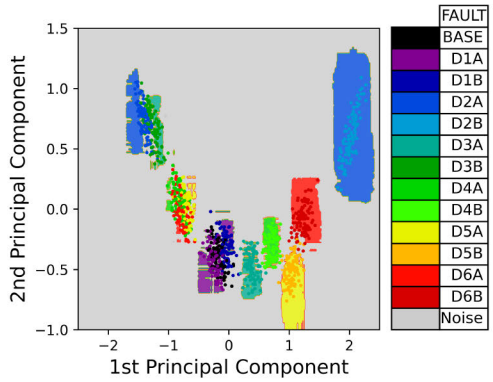
The data is mean-centered and then randomly split in training, validation and testing sets in a ratio of $\frac{1}{3} : \frac{1}{3} : \frac{1}{3}$. The training set is used exclusively for training a neural network, through back-propagation, that can predict the output class. After each epoch, the validation set is used to ensure that the model is not over-fitting on the training data. The validation data is generally used to assess the performance of a model, as seen in Figs. 11 and 12. One of the advantages specific to the CNNs over the MLP is that they are spatially invariant. This allows



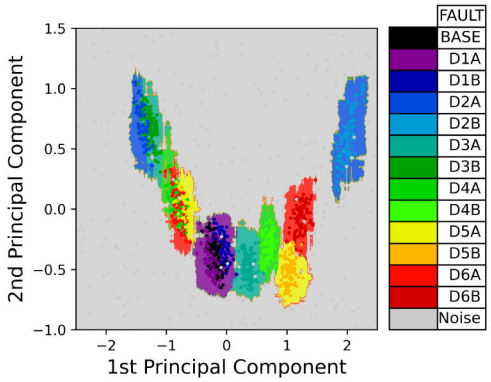
(a) First two principal components.



(b) Latent space segmentation with no noise.



(c) Latent space segmentation with 95% threshold for noise class.



(d) Latent space segmentation with stochastic noise class.

FIGURE 8. Latent space of the first two principal components when using the ExtraTrees classifier.

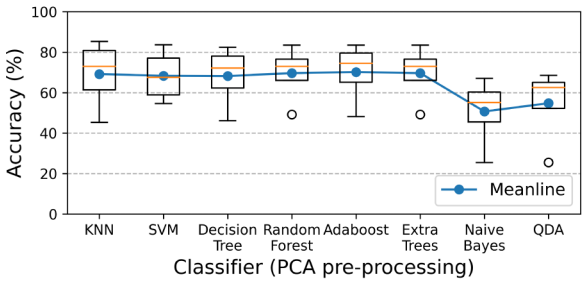


FIGURE 9. Box-and-whisker diagram showing the accuracy of all statistical learning algorithms using start-up data.

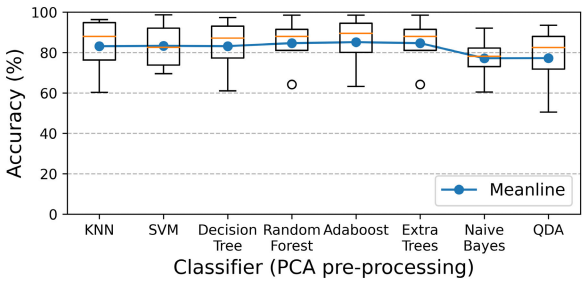


FIGURE 10. Box-and-whisker diagram showing the accuracy of all statistical learning algorithms using steady-state data.

for the detection of discriminative features in a manner that is independent of the position of these features [17].

B. EVALUATION METHODOLOGY

In Fig. 11, four models are compared in terms of validation accuracy using the ±50% dataset, while in Fig. 12, they are compared using the ±20% dataset. In both figures the validation accuracy is shown with a rolling average of 20 epochs and with a shaded region indicating two standard deviations of this rolling average (2σ). The Inception network in both cases initially converges to a higher validation accuracy than the FCN. Still, the FCN converges to a higher validation accuracy by the end of the training epochs. The Residual network achieves a maximum validation accuracy higher than all the other models, but this does not correspond to similar accuracy on the testing data. This could be because the network is over-fitting on the validation data due to validation bias or high variance in the testing data [33] due to the multi-modal nature of the deep learning models. These models are constrained significantly by the training and validations times

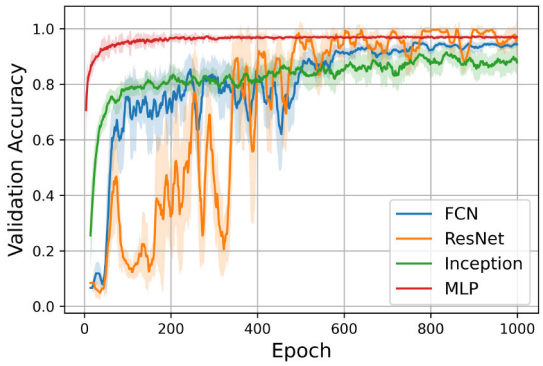


FIGURE 11. Deep learning validation accuracy for ±50% faults.

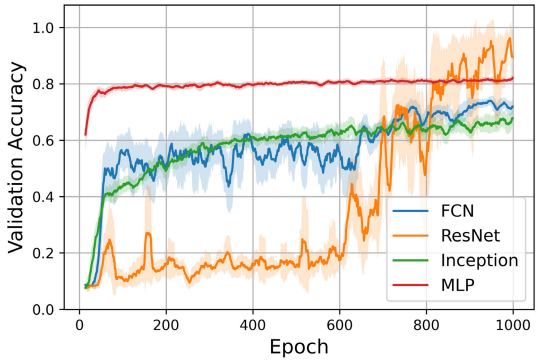


FIGURE 12. Deep learning validation accuracy for ±20% faults.

to produce superior results. When considering both training time and performance, the MLP presents a good trade-off.

C. NETWORK INTERPRETABILITY

The Inception network is interpreted with the use of a CAM on the global average pooling layer. The discriminative regions of the time-series are highlighted in a darker shade than the non-discriminative regions. The model uses various regions in the time-series to classify the data. Two common fault cases from the same dataset, namely start-up data for the ±50% set for faults D4B and D2A, are shown in Figs. 13 and 14. Fig. 13 shows how the model uses the start-up region of the signal as the discriminative region. Fig. 14 shows how the model uses the peaks of the oscillating signal as the discriminative region. The discriminative regions have an increased percentage contribution under extensive training.

TABLE 3. YL and TE for the steady-state data.

	Harmonics		Statistical learning		Deep learning	
Fault set	YL	TE	YL	TE	YL	TE
$\alpha = 0.2$	**22.56%**	3.96%	53.6%	4.18%	55.20%	**3.11%**
$\alpha = 0.5$	**1.51%**	**1.50%**	33.59%	2.31%	7.63%	2.04%
$\alpha = 0.9$	**0.75%**	0.82%	7.26%	0.40%	6.45%	**0.0%**

TABLE 4. Yield loss and test escapes for the start-up data.

	Statistical learning		Deep learning	
Fault set	YL	TE	YL	TE
$\alpha = 0.2$	**34.48%**	1.94%	56.55%	**1.66%**
$\alpha = 0.5$	**2.24%**	0.14%	5.97%	**0.0%**
$\alpha = 0.9$	0.78%	**0.0%**	**0.0%**	0.7%

VI. SUMMARY OF RESULTS

To enable comparison to prior work, the fault diagnosis presented previously may be aggregated to binary fault detection, common in conventional OBT. In this context, the false detection of any fault in a working circuit may be considered yield loss (YL). In contrast, the false classification of any faulty circuit as a working circuit may be considered a test escape (TE). The results as a percentage error for all three parametric fault testing points, namely $\alpha = 0.2, \alpha = 0.5, \alpha = 0.9$ (corresponding with ±20%, ±50% and ±90% respectively), is shown for the steady-state data in Table 3 and the start-up data in Table 4. The best classifier fault group with the lowest percentage error for TE and YL is shown in bold for both result sets.

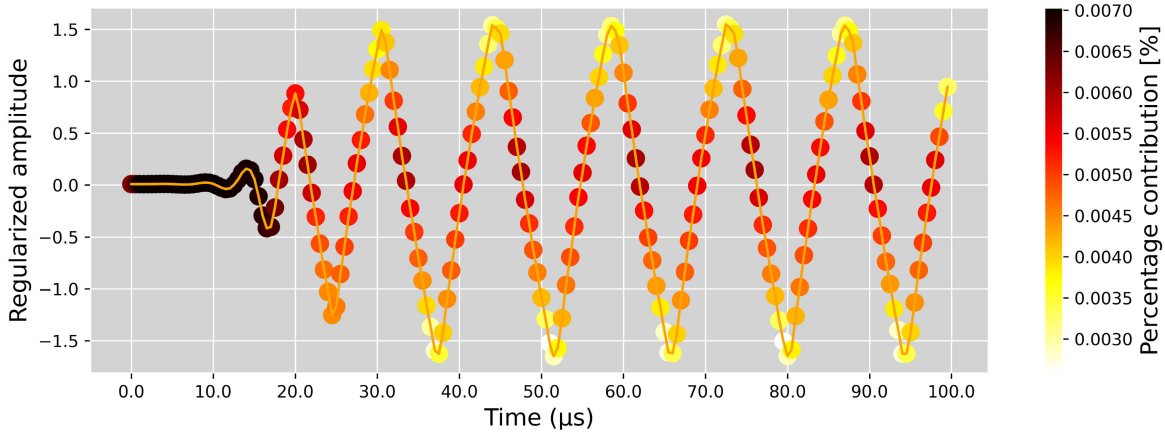


FIGURE 13. Class activation map showing start-up discriminative region.

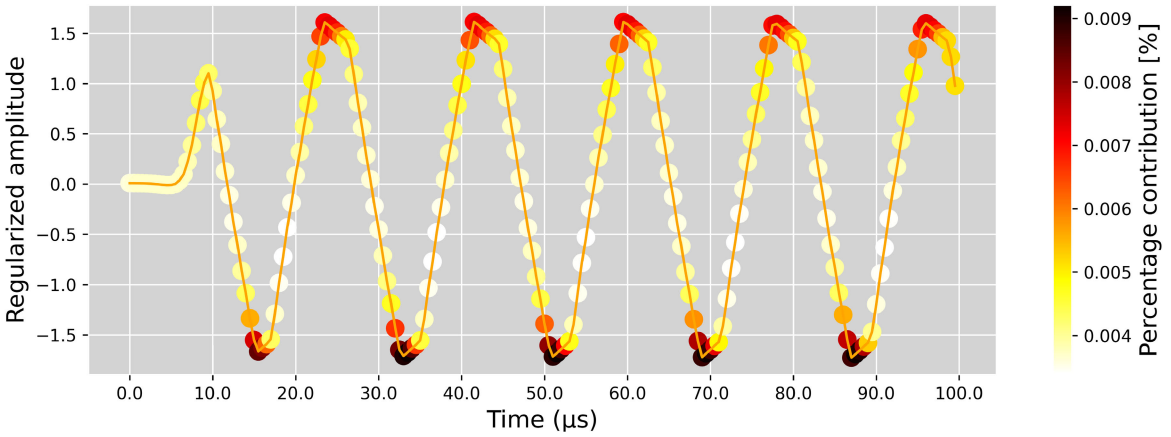


FIGURE 14. Class activation map showing steady-state discriminative region.

TABLE 5. Overall results.

	Harmonics		Statistical learning		Deep learning	
Fault set	Start-up	Steady	Start-up	Steady	Start-up	Steady
$\alpha = 0.2$	-	65.4%	71.2%	72.5%	44.3%	**76.8%**
$\alpha = 0.5$	-	82.9%	83.1%	87.6%	69.7%	**98.2%**
$\alpha = 0.9$	-	90.1%	90.3%	95.8%	86.8%	**98.7%**

The classification accuracy for each fault group on both the start-up and steady-state data is shown in Table 5, with the highest accuracy for steady-state and start-up shown in bold.

Tables 3-5 show that the statistical and deep learning approaches can be used to detect faults with accuracy comparable to the harmonic analysis approach when considering the YL and TE metrics. The deep learning and statistical learning approaches outperform the harmonic analysis approach when performing diagnosis, as seen in Table 5. The single parametric defects seen in [2] would correspond to the $\alpha = 0.2$ using harmonic analysis in this work, as seen with the accuracy of 65.4%. This accuracy is lower than the accuracy seen in the prior work, which diagnoses all the faults correctly. However, these faults do not consider component tolerances of non-faulty components when performing diagnosis, further highlighting the impact of our approach to modeling the circuit. When considering the non-faulty component tolerances, the deep learning approach outperforms the harmonic analysis approach by an average of 11.77% for the three parametric fault sizes (α) considered in this work.

VII. CONCLUSION

This study demonstrated the use of time-series classification techniques for OBD, which subsequently enabled the classification of OBD signals within the start-up region. Class Activation Maps were used to interpret the deep learning networks showing how the discriminative regions vary between start-up or steady-state features. The statistical learning and deep learning networks outperformed the harmonic analysis approaches when using simulation data for diagnosis. This advances the state-of-the-art by introducing new methods for fault diagnosis and using the start-up region for fault diagnosis.

This work paves the way for several avenues of future inquiry. From a circuit perspective, the effect of the process, supply voltage and temperature variation may be considered (as is done to some extent in [4]) and variation in the active circuit elements. The probabilistic result outputs of all the classifiers may be amended to functional or performance testing, rather than structural testing, where key performance parameters are the classification output rather than the diagnosis of a specific fault [34]. Finally, the techniques may be evaluated at RF frequencies or combined with built-in self-testing [8], generating a time-series from an oscillator envelope rather than the Nyquist-rate sampled output.

REFERENCES

[1] D. Binu and B. S. Kariyappa, "A survey on fault diagnosis of analog circuits: Taxonomy and state of the art," *AEU-Int. J. Electron. Commun.*, vol. 73, pp. 68–83, Mar. 2017.

[2] M. A. Stošović, M. Milić, M. Zwolinski, and V. Litovski, "Oscillation-based analog diagnosis using artificial neural networks based inference mechanism," *Comput. Elect. Eng.*, vol. 39, no. 2, pp. 190–201, 2013.

[3] K. Arabi and B. Kaminska, "Testing analog and mixed-signal integrated circuits using oscillation-test method," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 16, no. 7, pp. 745–753, Jul. 1997.

[4] T. Stander, P. Petrashin, L. Toledo, W. Lancioni, C. Vazquez, and F. C. Dualibe, "Influence of oscillator topology on fault sensitivity in oscillation based testing (OBT) of OTAs," in *Proc. Argentine Conf. Electron. (CAE)*, Mar. 2019, pp. 1–5.

[5] P. A. Petrashin, L. E. Toledo, W. J. Lancioni, C. Vazquez, T. Stander, and F. C. Dualibe, "Influence of passive oscillator component variation on OBT sensitivity in OTAs," in *Proc. IEEE 19th Latin-Amer. Test Symp. (LATS)*, Mar. 2018, pp. 1–4.

[6] P. Petrashin, L. Toledo, W. Lancioni, P. Osuch, and T. Stander, "Oscillation-based test applied to a wideband CCII," *VLSI Design*, vol. 2017, pp. 1–6, May 2017.

[7] M. Ballot and T. Stander, "A RF amplifier with oscillation-based BIST based on differential power detection," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–4.

[8] H. P. Nel, T. Stander, and F. C. Dualibe, "Built-in oscillation-based self-testing of a 2.4 GHz LNA in 0.35μm CMOS," in *Proc. 25th IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Dec. 2018, pp. 837–840.

[9] V. Litovski, M. Andrejeviá, and M. Zwolinski, "Analogue electronic circuit diagnosis based on ANNs," *Microelectron. Rel.*, vol. 46, no. 8, pp. 1382–1391, Aug. 2006.
[10] M. A. Stosovic, D. Milovanovic, and V. Litovski, "Hierarchical approach to diagnosis of mixed-mode circuits using artificial neural networks," *Neural Netw. World*, vol. 21, no. 2, p. 153, 2011.
[11] V. Litovski, "Analog filter diagnosis using the oscillation based method," *J. Electr. Eng.*, vol. 63, no. 6, pp. 349–356, Dec. 2012.
[12] Y. Wang, H. Wang, D. Meng, and B. Zhou, "Oscillation-based diagnosis by using harmonics analysis on analog filters," in *Proc. Tech. Papers Int. Symp. VLSI Design, Autom. Test*, Apr. 2014, pp. 1–4.
[13] J. Peralta, M. Costamagna, G. Peretti, E. Romero, and C. Marques, "Estimating the quality of oscillation-based test for detecting parametric faults," in *Proc. 10th Latin Amer. Test Workshop*, Mar. 2009, pp. 1–6.
[14] A. Singhee and R. A. Rutenbar, "Why quasi-Monte Carlo is better than Monte Carlo or Latin hypercube sampling for statistical circuit analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 11, pp. 1763–1776, Nov. 2010.
[15] A. Michard, F. Cacho, D. Celeste, and X. Federspiel, "Global and local process variation simulations in design for reliability approach," in *Proc. IEEE 25th Int. Symp. On-Line Test. Robust Syst. Design (IOLTS)*, Jul. 2019, pp. 72–75.
[16] W. Booyse, D. N. Wilke, and S. Heyns, "Deep digital twins for detection, diagnostics and prognostics," *Mech. Syst. Signal Process.*, vol. 140, Oct. 2020, Art. no. 106612.
[17] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Deep learning for time series classification: A review," *Data Mining Knowl. Discovery*, vol. 33, no. 4, pp. 917–963, Jul. 2019.
[18] T. Han, Y.-F. Li, and M. Qian, "A hybrid generalization network for intelligent fault diagnosis of rotating machinery under unseen working conditions," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 3520011.
[19] N. Wongprommoon, A. Tiamsuphat, and P. Prommee, "Low-complexity Chebyshev high-pass filter based on OTA-C," in *Proc. 43rd Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2020, pp. 373–376.
[20] D. Y. Denisenko, N. N. Prokopenko, and N. V. Butyrlagin, "All-pass second-order active RC-filter with pole Q-factor's independent adjustment on differential difference amplifiers," in *Proc. IEEE East-West Design Test Symp. (EWDTS)*, Sep. 2019, pp. 1–4.
[21] *Micro-Cap 12 Download*, Spectrum Software, Sunnyvale, CA, USA, 2021.
[22] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-V4, inception-resnet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, 2017, pp. 4278–4284.
[23] C. M. Bishop, "Machine learning and pattern recognition," in *Information Science Statistics*. Berlin, Germany: Springer, 2006.
[24] D. N. Wilke, P. S. Heyns, and S. Schmidt, "The role of untangled latent spaces in unsupervised learning applied to condition-based maintenance," in *Proc. Int. Workshop Modeling Simulation Complex Syst. Sustain. Energy Efficiency*. New York, NY, USA: Springer, 2021, pp. 38–49.
[25] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY, USA: Springer, 2001.
[26] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomised trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.
[27] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25, Dec. 2012, pp. 1097–1105.
[29] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. M'uller, and F. Petitjean, "Inceptiontime: Finding alexnet for time series classification," *Data Mining Knowl. Discovery*, vol. 34, pp. 1936–1962, Sep. 2019.
[30] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.
[31] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1578–1585.
[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2004.
[33] A. Y. Ng, "Preventing 'overfitting' of cross-validation data," in *Proc. ICML*, vol. 97, 1997, pp. 245–253.
[34] A. Goyal, M. Swaminathan, A. Chatterjee, D. C. Howard, and J. D. Cressler, "A new self-healing methodology for RF amplifier circuits based on oscillation principles," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 10, pp. 1835–1848, Oct. 2012.

JACOB B. CLOETE received the B.Eng. degree in mechanical engineering from the University of Pretoria, in 2020, where he is currently pursuing the B.Eng. degree (Hons.).

His research interests include physics-based (computational fluid dynamics) and data-driven modeling.

TINUS STANDER (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees in electronic engineering from the University of Stellenbosch, Stellenbosch, South Africa, in 2005 and 2009, respectively.

From 2010 to 2012, he was an RF and Microwave Engineer with Denel Dynamics (a division of Denel SOC Ltd.), Irene, South Africa, before joining the Carl and Emily Fuchs Institute for Microelectronics, Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South Africa, in 2013. He currently serves as an Associate Professor and the Institute's Principal Investigator for microwave and mm-wave microelectronics, with personal research interest includes the application of distributed passives on-chip and built-in self-testing. He is also registered as a Professional Engineer with the Engineering Council of South Africa and works as a Scientific Advisor at Multifractal Semiconductors (Pty.) Ltd., Pretoria.

DANIEL N. WILKE received the B.Eng., M.Eng., and Ph.D. degrees in mechanical engineering from the University of Pretoria, South Africa, in 2002, 2005, and 2010, respectively.

Since 2007, he has been teaching and researching at the Department of Mechanical Engineering, University of Pretoria, as a Lecturer, a Senior Lecturer, an Associate Professor, and a Professor. He has coauthored the book *Practical Mathematical Optimization* (Springer) with Prof. Jan Snyman, in 2018. His research interests include physics-based (discrete element and finite element modeling) and data-driven generative modeling from a strongly biased optimization viewpoint supported by efficient GPU computing.