Machine Learning-Based Approach for Hardware Faults Prediction

Kasem Khalil

IEEE Transactions on Circuits and Systems I: Regular Papers

Machine Learning-Based Approach for Hardware Faults Prediction

Kasem Khalil, *Member, IEEE*, Omar Eldash, *Member, IEEE*, Ashok Kumar, *Senior Member, IEEE*, and Magdy Bayoumi, *Life Fellow, IEEE*

***Abstract*— Hardware failures are undesired but a common problem in circuits. Such failures are inherently due to the aging of circuitry or variation in circumstances. In critical systems, customers demand the system never to fail. Several self-healing and fault tolerance techniques have been proposed in the literature for recovering a circuitry from a fault. Such techniques come to the rescue when a fault has already occurred but they are typically uninformed about the possibility of an impending failure (i.e., fault prediction), which can be used as a pre-stage to fault tolerance and self-healing. This paper presents an approach to early fault prediction of circuits. The proposed method uses Fast Fourier Transform (FFT) to get the fault frequency signature, Principal Component Analysis (PCA) to get the most important data with reduced dimension, and Convolutional Neural Network (CNN) to learn and classify the fault. The proposed method is validated for working in different circuits by testing it using two circuits: comparator and amplifier. The comparator and amplifier are implemented using 45 nm technology on HSPICE to extract the failures dataset in terms of voltage, current, temperature, noise, and delay. This extracted data is used for training the proposed approach using Tensorflow. To the best of our knowledge, this is the first work of fault prediction at the transistor level for hardware system. The proposed approach considers aging, short-circuit, and open-circuit faults, and it provides a fault prediction accuracy of 98.93% and 98.91% for comparator and amplifier circuits, respectively. The proposed method is tested for two different circuits for its validation, and it consumes 1.08 W for Altera Arria 10 GX FPGA device.**

***Index Terms*— Fault prediction, machine learning, CNN, fast Fourier transform, principal component analysis, neural network, fault tolerance, self-healing, biomedical system.**

I. Introduction

PROGRESSIVE scaling of a device in a hardware system and Very Large Scale Integrated (VLSI) has enabled the creation of more complex circuits and systems. However, decreasing device size increases vulnerability to faults because of phenomena such as design defects, high energy particles, and aging. Consequently, the reliability of hardware is reduced, and such reduced reliability causes significant concern for devices used for critical needs such as healthcare and security. The traditional way to handle fault tolerance or self-healing, in the literature, has been based on redundancy or replication. Advanced VLSI introduces more complex processors and system, implemented on advanced hardware architecture. The hardware system performance may decrease or break due to hardware failures in some components. Failures occur when running some real-life tasks due to the aging of the hardware system or the surrounding environment changes. At the circuit level, a scaling of transistor technology also causes process variation, which causes degraded performance yield [1]. Process variations have a significant impact on digital circuits. They have a significant impact on mixed-signal and analog circuits, because of the design sensitivity to device mismatch and integration of sub-blocks that can vary in parameters such as noise, characteristics, and operating frequency.

Several research studies are being conducted on how to detect or predict any fault in the system [2]. Fault prediction techniques can significantly help the self-healing of a hardware system. A self-healing approach uses fault detection to know which part of the circuit or system has a fault to compensate (compensation is typically done through redundancy or replication). Self-healing is a part of intelligent hardware system research which aims to make the hardware system smart [3]. The intelligent hardware system is an important framework that is used for hardware optimization based on the changes in the surroundings, running operations, and competing goals.

In biomedical devices, an Electrocardiogram (ECG) is used to record the electrical activity of the heart. It presents the heart rate and rhythm information. It shows a diagnosis in case if there is heart enlargement due to high blood pressure, which is called hypertension or myocardial infarction, and is evidence of previous heart attack [4]. In the biomedical hardware component, a comparator is used to compare the input signal with a threshold reference voltage. The comparator is also used in Analog to Digital Converters (ADCs), which is used for ECG data recording [5]. An amplifier is also used in biomedical instrumentation systems. Input signals are sensitive to noise, and its voltage value is low. These signals are amplified, by the amplifier, for signal processing and ADCs [6], [7]. The main and common component of the comparator, amplifier, and all electronic circuits is a transistor. Therefore, transistor faults/failures are vital to consider in the design as well as fault predictions.

Manuscript received March 29, 2020; accepted July 14, 2020. This article was recommended by Associate Editor J.-Y. Kim. *(Corresponding author: Kasem Khalil.)*

Kasem Khalil, Omar Eldash, and Ashok Kumar are with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA 70504 USA (e-mail: kasem.khalil1@louisiana.edu).

Magdy Bayoumi is with the Department of Electrical and Computer Engineering, University of Louisiana at Lafayette, Lafayette, LA 70504 USA.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2020.3010743

Transistor faults are a result of aging or circumstance changes such as voltage, current, noise, delay, and temperature. Therefore, the transistor diagnosis must study these parameters for detecting or predicting faults/failures [8], [9]. There are several techniques to detect an existing fault. A self-healing or fault tolerance technique is then applied to fix the fault [10]–[12]. The drawback of fault detection is that some tasks or data will be lost due to the current fault (since the fault is fixed after it has already occurred), and it affects the system performance. The goal to solve this problem is a fault prediction that provides early transistor diagnosis. Thus, the machine learning role comes to provide an early prediction. Machine learning has different structures, such as a Recurrent Neural Networks (RNN) [13] and Convolutional Neural Network (CNN) [14]. The machine learning method depends on faults' parameters for learning. The benefit of early fault prediction is that it can help self-healing/ fault tolerance methods to fix the fault before it occurs, thus protecting the system performance. The data from fault prediction can be used by self-healing method to recover a fault. As the self-healing method gets the information of an imminent fault time and location, it can recover it without incurring a down-time for the system [15]. This paper focuses on the stage of fault prediction through generating a dataset of faults and fault classification.

In this paper, we propose an approach to transistor fault prediction using FFT, PCA, and CNN. This technique provides a high-accuracy machine learning method for predicting hardware faults. Early fault prediction helps to diagnose the fault. Therefore, the fault can be fixed early to avoid any missed data or operations. The contribution of this work can be summarized in the following points:

- A hardware fault prediction method is proposed. This method provides fault prediction in the transistor level, and it is based on FFT, PCA, and CNN.
- The proposed fault prediction is tested in two different circuits: comparator and amplifier that are used for ADC applications. The proposed method is tested in different circuits to show that its scalability to work for different circuits with the same performance.
- The proposed method provides a prediction with an accuracy of 98.93% which is higher than the traditional methods. The hardware implementation cost of the proposed method is presented.

The remainder of this paper is organized as follows. Section II presents an overview of faults. Section III presents the proposed method of fault prediction with applications on comparator and amplifier. In section IV, the implementation of the proposed method, and the experimental results are presented, followed by the conclusion in Section V.

II. Faults Overview

A fault is an abnormal physical condition in a hardware system that causes an error. An error is a demonstration of a fault in the hardware system. The output deviates from the expected value because the logical state of a component differs from the intended state. Furthermore, failure is the inability of the system to perform its functionality or behavior. A failure might happen due to chain error propagation to the system level. However, the fault in the hardware system is not a significant result in an error or failures as it might become inactive. Failure has occurred as a type of communication failures because of broken wire, loosening connectors, circuit board level shorts and opens, failing communication transceivers, communication timing issues, and electromagnetic interference. Transistor aging has become as troublesome phenomena in complex processors. The aging results in performance degradation and failure [16]. The main aging mechanisms are Electromigration (EM), Stress Migration (SM), Negative Bias Temperature Instability (NBTI), Time-Dependent Dielectric Breakdown (TDDB), and Hot Carrier Injection (HCI).

EM results due to the excessive stress of current density [17]. This phenomena lead to a sudden delay increase, short, or open faults. The EM issue is located in the interconnect, and it can be defined as the physical displacement of the ions of metal in the interconnection wires. This kind of displacement is resulted due to a large flow of electronics (which is called a large current density mechanism) that interacts with the metal ions [18]. Voids and hillocks are resulted due to this movement, and this phenomenon causes short circuits or open connections. As the EM is accelerated close to the metal grain boundaries, contact holes and vias are susceptible to this impact. The EM expression [17] is derived in terms of Mean Time To Failure (MTTF) as:

$$MTTF_{EM} \sim AJ^{-n} exp^{\frac{E_{aEM}}{KT}} \tag{1}$$

where A is the cross-section area of the wire, J is the current density in the wire, E_{aEM} is EM's material-dependent activation energy constant (0.9 for the copper interconnects), n is interconnect metal constant (1.1 for the copper interconnects), K is the Boltzmann constant, and T is the absolute temperature in Kelvin. Therefore, with a larger A and a smaller J, it results in a longer lifetime MTTF.

The SM occurs due to excessive structural stress [19]. This phenomenon is similar to EM where it leads to a sudden delay increase, short, or open faults. In this mechanism, the metal atoms migrate in the interconnects because of mechanical stress which is similar to electromigration. The stress migration is resulted by thermo-mechanical stresses that are produced by different rates of thermal expansion of different materials. The calculation of EM depends on MTTF due to stress migration [19], and it can be written by

$$MTTF_{SM} \sim \mid T_0 - T \mid^{-m} exp^{\frac{E_{aSE}}{KT}} \tag{2}$$

where T_0 is the metal stress-free temperature which is the deposition temperature of the metal, $E_{a_{SM}}$ is the material-dependent activation energy constant (0.9 for the copper interconnects), m is material constant (2.5 for the copper interconnects).

NBTI affects a PMOS transistor in terms of threshold voltage degradation due to a stressed transistor with negatively biased gate voltage [20]. NBTI can be defined as a threshold voltage shift due to a negative bias is applied to the gate of MOS at high temperature. The threshold voltage shift V_{th} depends on temperature, stress time, and voltage. The BTI

voltage shift ΔV_{th} can be written as follows.

$$\Delta V_{th} = A\ exp^{(\beta V_{GS})} exp^{(-\frac{E_a}{KT})} \alpha^n t^n \tag{3}$$

where A, β, and n are constants, V_{GS} is the gate voltage, α is the duty cycle which is the ratio of the stress time to total time, and t is operating time. E_a is the BTI activation energy constant.

TDDB refers to insulating film breakdown due to continuous stresses to a gate oxide-film causes. It causes a sudden delay increase (slow delay degradation up to a certain point) or a failure [21]. There are different Breakdown (BD) modes: Hard-BD (HBD) and Soft-BD (SBD). The HBD is the most harmful mode, and it causes a complete loss of the dielectric oxide properties with gate currents in the rage of mA. The SBD is defined as a partial loss of the oxide dielectric properties, and it causes an increase in the noise and magnitude of the gate current. The time to breakdown (t_{DB}) is expressed as probability distribution [22]:

$$F(t_{BD}) = 1\ -\ exp^{-(\frac{t_{DB}}{t_{63}})^{\beta}} \tag{4}$$

where t_{63} is the time to breakdown at 63%, and it proportional to the size of the transistor and inversely proportional to V_{GS}. β is called a process-dependent constant.

HCI affects a NMOS transistor by increasing the threshold voltage under the stress of source-drain voltage. It causes gradual delay degradation the same as NBTI [23]. The hot carriers are defined as particles that have high kinetic energy which is accelerated by a high electric field. The energetic electrons may be injected into the forbidden regions of the transistor (gate oxide layer). These electrons can get trapped or cause an up-normal interface. This kind of defect leads to an increased threshold voltage of V_{th}. Hot carrier effect is modeled using a power-law dependency on the stress time. The damage is proportional to increasing gate to source voltage (V_{GS}), drain to source voltage (V_{DS}) [24].

$$\Delta V_{th} \sim \frac{1}{\sqrt{L}} exp^{(\alpha_1 V_{GS})} exp^{(\alpha_2 V_{DS})} T^{n_{HC}} \tag{5}$$

where L is the transistor length, T is temperature, n_{HC} is constant ($n_{HC} = 0.5$), α_1 and α_2 re technology-dependent constant of voltage scaling.

The classification of faults is divided into three categories: permanent, transient, and intermittent fault [25]. The permanent fault is irretrievable physical damage in the system, and it is a continuous fault and stable with time. The permanent fault results from different sources such as; stuck-at zero or stuck-at one which fixes the logic values, short circuit due to a connection between two lines. Open-line is another fault that results from splitting a line into parts. Delay fault is due to the propagation delay in the hardware system. Furthermore, bridge source is due to two wires in a network that are connected accidentally and wire performs as a connection.

The transient fault is a fault that comes from external disturbance, and it may stay for a short period. According to transient time faults, there are three types: bit-flip, pulse, and delay. Bit-flip occurs when a value of bit changes to the opposite value, for example, switching value from '1' to '0' or vice versa. Pulse comes from the transition of a pulse, which is called single event transition. Delay comes from the mismatch in hardware, which makes propagation delay.

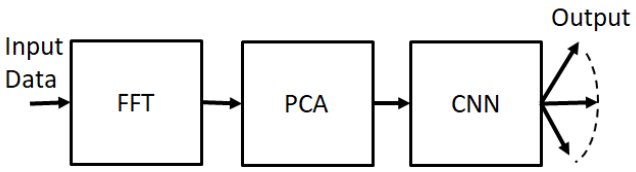


Fig. 1. The block diagram of the proposed method.

The intermittent fault is the third type, which results from unstable or marginal device operation, and it is difficult to detect them compared to the permanent one. This type of fault is a kind of transient fault that repeats with some frequencies.

III. The Proposed Method

Hardware faults cause performance degradation in a hardware system. This paper presents a novel approach for fault prediction in terms of aging, short-circuit, and open-circuit fault. The proposed method depends on FFT, PCA, and CNN, as shown in Fig. 1. The FFT is used to demonstrate a signal in the frequency domain, which gives a more suitable indication of faults. The fault impact on a signal is dominated by some major frequency components, and the dominant frequencies are significant for monitoring. The spectrum for low to high frequencies indicates an earlier warning of faults resulted from frequency components with smaller amplitudes. Different changes, also, in the frequency components and their bands are associated with various faults [26]. The next stage is based on PCA which is used to get the most important parameters where it removes unimportant data. The CNN gets the PCA's output as an input to learn from it the kind of faults and provides the fault classification at the final output result. The contribution of the proposed method can be explained as follows. Traditionally, fault detection/prediction can be provided using an Artificial Neural Network (ANN) or Support Vector Machine (SVM). The fault signals do not contain a lot of information to get an accurate result using ANN or SVM. Therefore, the first point is to extend the fault signal for getting sufficient data in learning. From this point, the idea of using FFT to the fault signals comes to extend the data. The data in the frequency domain provides a signature and characteristic of fault. After FFT, the data has some unimportant information as before FFT. Therefore, the PCA processing is used to remove this unimportant information to reduce the training time of CNN. The data after all these processes are still large and sufficient (the data has large sufficient features) for learning a network using CNN. The final data are complex; thus, CNN (CNN is used for complex data) is used for feature extraction and classification. The proposed method provides more accurate data to predict faults compared with the traditional methods such as ANN and SVM. This is due to that ANN and SVM are not a good choice for large sample classification while the proposed method can deal with large data using CNN. The description of each stage is discussed as follows.

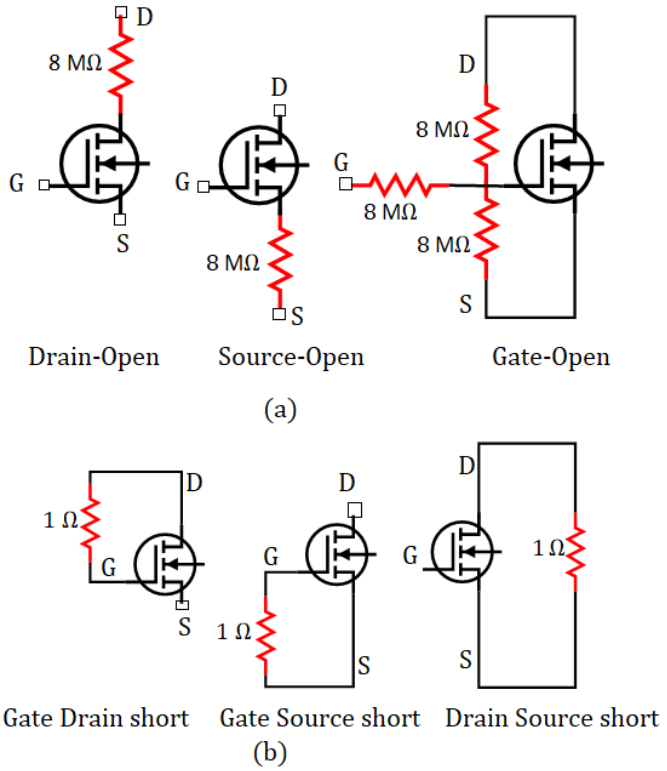


Fig. 2. Schematic diagram of the fault model (a) Open-circuit fault (b) Short-circuit fault.

A. Dataset and Fault Model

The proposed approach is used to predict a fault of a comparator and amplifier. These circuits are implemented on HSPICE using 45nm technology. The implementation is simulated to emulate aging, short-circuit, and open-circuit faults by modifying the diagnosis fault parameters. We extracted the data to build our dataset (voltage, current, noise, delay, temperature, EM, SM, NBTI, TDDB, HCI, input measurement errors, etc.) for the transistor level design, and this dataset is used in the proposed approach. On the one hand, the transistor open-circuit fault can be one of these forms: drain open, source-open, and gate-open. On the other hand, the transistor short-circuit has different shapes: gate-drain-short, drain-source-short, and gate-source-short. The model of the transistor short and open circuit is based on low and high resistance, respectively. For short-circuit fault, low resistance of 1 Ω is considered while high resistance of 8 $M\Omega$ is considered to open-circuit. The short-circuit and open-circuit faults are shown in Fig. 2. Using the experimental analysis using HSPICE, we changed the resistance for short-circuit until the voltage becomes very small and the voltage drop on this resistance has a very small variation when it is lower or equal 1 Ω. Therefore, the short-circuit resistance is chosen. For the open-circuit situation, a high resistance is used to present it. We changed the resistance value, and it is found that the voltage drop is very high when the resistance is 8 $M\Omega$. Also, the voltage variation is very small after increasing the resistance to more than 8 $M\Omega$. According to these results, the low and high resistance are selected. The aging faults may cause delay, noise, threshold voltage variation, or open-circuit, or short-circuit fault. Each transistor in comparator and amplifier circuits may get any one of these faults. The comparator and amplifier implementation are analyzed using Monte-Carlo analysis to present the effect of the transistor threshold-voltage variation to the circuit behavior. This simulation is done 100 times with the 6% variation in the threshold voltage. The final data has 14,683 samples, and it includes 150 samples for the nonfaulty state.

B. Feature Extraction and Classification

The fault feature selection and extraction is used to get a presentation of the fault and classification status. The feature selection and extraction, in the proposed approach, run through FFT, PCA, and CNN. FFT is used to get the frequency transformation of the input data. The PCA is used to reduce the data dimension by converting correlated data to uncorrelated data. This reduction will be helpful to CNN where the computation will be less. The CNN stage finalizes the feature extraction to be applied to the classification stage. Details of each stage will be discussed in the following subsections.

1) Fast Fourier Transform: Each hardware fault represents itself by a unique frequency signature [26]. Therefore, FFT is used to represent faults in a frequency domain to be easier for identifying a fault. Furthermore, the FFT is used to do data compression and feature extraction by preprocessing on the original signals. The FFT is a version of the Discrete Fourier Transform (DFT), but the FFT is faster [27]. The FFT utilizes some advanced algorithms to do the same thing as the DFT but in much less time. For instance, a DFT computation of N points in a fundamental way, using the definition, takes $O(N^2)$ arithmetic operations, while the FFT computation of the same result is in only $O(N\ logN)$ operations. Therefore, the advantages of using FFT can be described as follows. It provides the main features of a fault characteristic signals, it is easier to distinguish the faults by using the spectrum where each fault has a unique frequency signature, and FFT computation of N points is only $O(N\ logN)$ operations. A window function is used to get finite sequences for processing. The FFT is used for fault prediction at the system level in an embryonic hardware system in which faulty cells are repaired [15]. In the proposed technique, the FFT processes output signals and the first b frequencies are used for the feature data for the next step to PCA, where $b \ll$ the number of samples which is obtained by iteration. The PCA is used to improve the diagnostic accuracy and computational efficiency of hardware faults.

Assume $x_{i,n}$ is a discrete output signal (e.g, voltage, current, temperature, …) with $i = 1, 2, 3, \ldots\ldots\ldots, m$ and $n = 0, 1, 2, 3, \ldots\ldots\ldots, N-1$. After the FFT transformation, the output will be $X_{i,k}$ with $i = 1, 2, 3, \ldots\ldots\ldots, m$ and $k = 0, 1, 2, 3, \ldots\ldots\ldots, b-1$ where b is the retained harmonics size and m is the training samples size. The FFT is given by:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \quad k = 0, 1, \ldots., N-1 \tag{6}$$

$$W_N = e^{\frac{-j2\pi}{N}} \tag{7}$$

$$X(k) = \sum_{n\ even} x(n)\ W_N^{kn} + \sum_{n\ odd} x(n)\ W_N^{kn} \tag{8}$$

$$X(k) = \sum_{m=0}^{\frac{N}{2}-1} x(2m)\ W_N^{2km} + \sum_{m=0}^{\frac{N}{2}-1} x(2m)\ W_N^{2km} \tag{9}$$

With $W_N^2 = W_{N/2}$ substitution, the equation can be expressed as

$$= \sum_{m=0}^{\frac{N}{2}-1} h_1(m)\ W_{N/2}^{km} + \sum_{m=0}^{\frac{N}{2}-1} h_2(m)\ W_{N/2}^{km} \tag{10}$$

$$X(k) = H_1(k) + W_N^k\ H_2(k), \quad k = 0, 1, \ldots.., N-1 \tag{11}$$

where H_1(k) and H_2(k) are the $N/2$ point DFTs of the sequences h_1(m) and h_2(m), respectively. H_1(k) and H_2(k) are periodic, with period $N/2$, therefore $H_1(\mathrm{k}+\mathrm{N}/2) = H_1(\mathrm{k})$ and $H_2(\mathrm{k}+\mathrm{N}/2) = H_2(\mathrm{k})$. In addition, the factor $W_N^{k+N/2} = -W_N^k$. Thus, the equation may be expressed as

$$X(k) = H_1(k) + W_N^k\ H_2(k), \quad k = 0, 1, \ldots.., \frac{N}{2} \tag{12}$$

$$X(k+\frac{N}{2}) = H_1(k) - W_N^k\ H_2(k), \quad k = 0, 1, \ldots.., \frac{N}{2} \tag{13}$$

where N is the number of sampling points in output discrete signal. By these equations, the FFT transform of the input signal will be calculated to present the signature of the fault in the frequency domain.

2) Principal Component Analysis: PCA is used for dimension-reduction. It can reduce a large set of variables to a small set that still contains the most important information in the large set, to reduce the computation time for the next stage. Feature reduction using PCA process reduces the signal dimension and extracts the important, relevant features into feature vectors. The PCA assists fault prediction at the system level in an embryonic system [15]. The PCA is based on a mathematical procedure to transform a number of correlated variables into a (smaller) number of uncorrelated variables. The PCA depends on using an orthogonal transformation to convert variables set into values set of linearly uncorrelated variables [28], [29]. PCA mathematical analysis in this paper is presented. For a set of sample vectors x = $\{x^1, x^2, x^3, \ldots\ldots\ldots\ldots, x^n\}$ and orthogonal normalized basis A_i where $i = 1, 2, \ldots\ldots, +\infty$. For the orthogonal basis:

$$A_i A_k = \begin{cases} 1, & \text{if } i = k \\ 0, & \text{if } i \neq k \end{cases} \tag{14}$$

Each sample vector can be given as an infinite superposition of basis vectors which a basis has the same dimension. The sample vector is expressed as,

$$x^n = \sum_{i=1}^{\infty} \alpha_i^n A_i \tag{15}$$

The PCA depends on representing the original sample approximately by finite basis vector in order to reduce the error to the smallest possible. Thus, the estimated sample vector to the first d basis vector will consider the first d points, and it is given by,

$$\tilde{x}^n = \sum_{i=1}^{d} \alpha_i^n A_i \tag{16}$$

The subtraction between equation. 15 and equation. 16 is given by,

$$x - \tilde{x} = \sum_{i=1}^{\infty} \alpha_i A_i - \sum_{i=1}^{d} \alpha_i A_i \tag{17}$$

$$= \sum_{i=1}^{d} \alpha_i A_i + \sum_{i=d+1}^{\infty} \alpha_i A_i - \sum_{i=1}^{d} \alpha_i A_i \tag{18}$$

$$= \sum_{i=d+1}^{\infty} \alpha_i A_i \tag{19}$$

The error of PCA is calculated by using mean expectation (E), and it is obtained by,

$$error = E[(x - \tilde{x})(x - \tilde{x})^T] \tag{20}$$

$$= E[(\sum_{i=d+1}^{\infty} \alpha_i A_i)(\sum_{i=d+1}^{\infty} \alpha_i A_i)^T] \tag{21}$$

$$= E[\sum_{i=d+1}^{\infty} \alpha_i^2] \tag{22}$$

The error can be expressed using calculation in the following equations,

$$A_i^T x = \sum_{m=1}^{\infty} A_i^T \alpha_m A_m = \alpha_i \tag{23}$$

$$x^T A_i = \sum_{m=1}^{\infty} A_m^T \alpha_m A_i = \alpha_i \tag{24}$$

$$error = E \sum_{i=d+1}^{\infty} A_i^T x x^T A_i \tag{25}$$

$$error = \sum_{i=d+1}^{\infty} A_i^T E[x x^T] A_i \tag{26}$$

$$error = \sum_{i=d+1}^{\infty} A_i^T X A_i \tag{27}$$

Using the error value, the basis coefficients is adjusted by the error value to become as small as possible. The error is calculated using equation. 22 or equation. 27, where $X = E[xx^T]$. The minimum error value is obtained under constrained condition which is $A_i^T A_i = 1$. Therefore, the obtained equation is

$$X A_i = \lambda_i A_i \tag{28}$$

The minimum error value can be achieved when the basis vector is the eigenvectors of $E(xx^T)$. These eigenvectors can be calculated using a scatter matrix S,

$$S = \sum_{i=1}^{m} [(x_i - X_j)(x_i - X_j)^T] \tag{29}$$

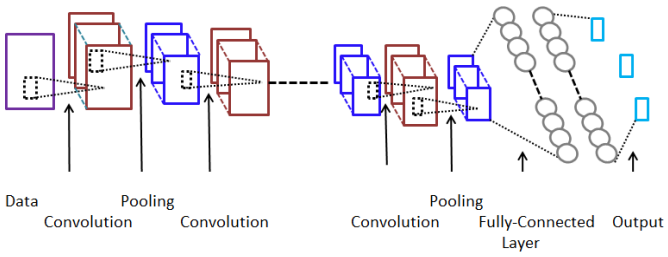


Fig. 3. Convolutional neural network architecture.

The eigenvectors are used to represent the components. The first mode or component of the sample vectors is referred by the eigenvector which corresponds to the largest eigenvalue. The second component refers to the eigenvector, which corresponds to the second largest eigenvalue, and the sequence of the other components is defined in the same definition. Consequently, the sample vectors go towards a lower dimension, which presents the benefit of using the PCA technique to the next stages of learning.

3) Convolutional Neural Network (CNN): CNN is used after PCA block to get PCA's output as CNN's input. The CNN architecture has three layers: convolutional, pooling, and fully connected layer, as shown in Fig. 3. The purpose of the convolutional layer is to learn feature representations of the input. It has multiple convolutional kernels to compute different feature maps. The convolution's output applies to a nonlinear activation function, and the used activation function in the proposed architecture is called Rectifier Linear Unit (RELU) [30]. The RELU function is used to add nonlinearity and provides robustness against noise in the input for the classification model, and the convolutional output size and RELU function are given equation 30 to 31, respectively.

$$F_{size} = \frac{N - F}{S} + 1 \tag{30}$$

where F_{size} is the resulted size of convolution, N is the input size, F is the filter size, and S is the stride size.

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{31}$$

The pooling layer comes after the convolutional layer, and its goal is to achieve shift-invariance by reducing the resolution of the feature maps to reduce the dimension of the output feature maps. The pooling layer output's size is obtained according to the filter size and moving step of the kernels (stride) as given by the equation 30. The pooing layer is usually used between two convolutional layers. Each resulting feature map is connected to its corresponding feature map of the previous convolutional layer. The average pooling type is used in our design [31]. In our design, the number of convolutional layers and the number of pooling layers is five layers. The parameters of CNN are changed many times, and the final parameters which give the desired performance are set as follows. A filter size of 5×5 is used in the convolution operation, and the pooling layer uses a 3×3 filter size using stride, which equals one.

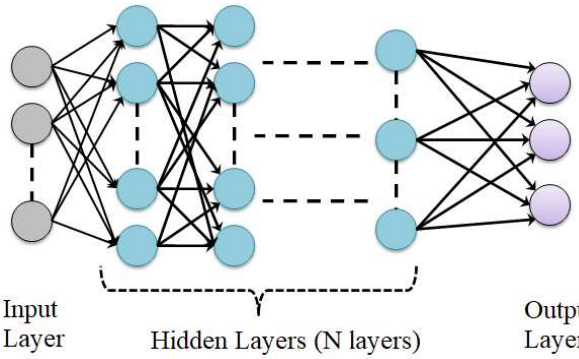


Fig. 4. Fully-connected layer architecture.

After multiple convolutional and pooling layers, there is a fully connected layer to perform high-level classifications [32]. All neurons in a certain layer are connected to each neuron of the next layer to generate global semantic information [33], [34], as shown in Fig. 4. A multi-level perceptron to assist with fault prediction at the system level in an embryonic system is discussed in [15]. In our model, Softmax regression is used for classification operation. It calculates the probabilities of each class versus all other classes, and the function is obtained by,

$$f(x_j) = \frac{e^{x_j}}{\sum_1^i e^{x_i}} \quad for \quad j = 1, 2, 3, \ldots, k \tag{32}$$

where x is the input signal and K is the number of output classes.

C. Training and Testing

The training steps of the proposed method are shown in Fig. 5. The total number of samples is 14683 samples are used for training, validation, and testing. In the training mode, the 60% of the samples are used to training, and they are 8809 samples. The 20% of the samples (2937 samples) are used for validation, and 20% (2937 samples) are used for testing. The data sample applies to the FFT preprocessing to get the data signature in the frequency domain. The output result of FFT is transferred to parameters setting of PCA to get feature compression by using a vector basis. The feature selection and compression extraction in this step gets the uncorrelated data with small dimensions. The CNN stage gets the PCA stage's output as its input to complete feature extraction and the learning steps. The next stage is the classification stage is used to determine the fault status. Finally, we compare the predicted fault status with the actual fault status. If the predicted fault is the same as the actual fault, the training will stop. Otherwise, feedback of the result returns to the parameter setting of PCA to update coefficients for getting more accurate learning. This iteration continues until the predicted fault status is the same as the actual one. In the testing phase, new data is applied to FFT transformation, the FFT's result is sent to the PCA processing. The roles of CNN and classification comes after PCA processing to complete the process, and they provide the final classification decision of the fault. The structure of the flow is shown in Fig. 5. The model of training and testing is shown in Fig. 6. The process is divided into

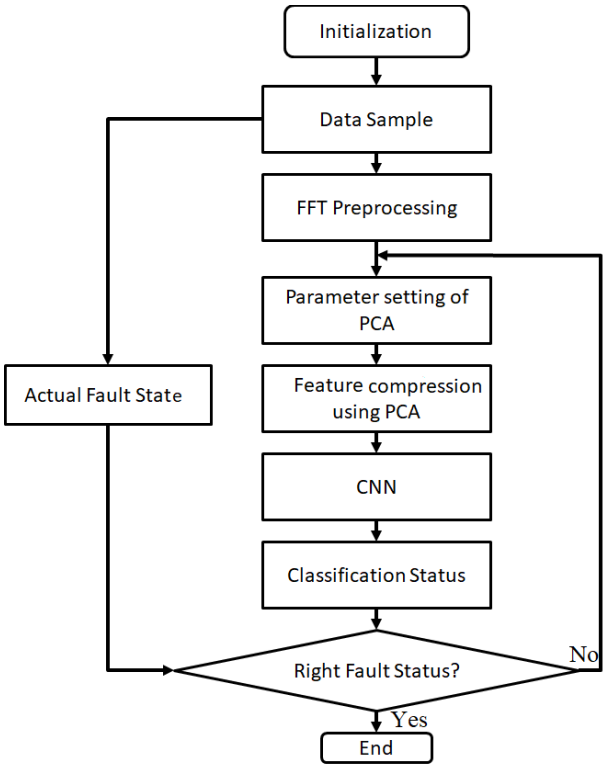


Fig. 5. Training flow chart of the proposed approach.

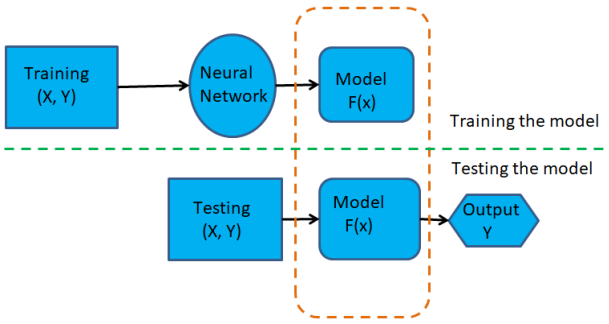


Fig. 6. A model of training and testing of data using CNN.

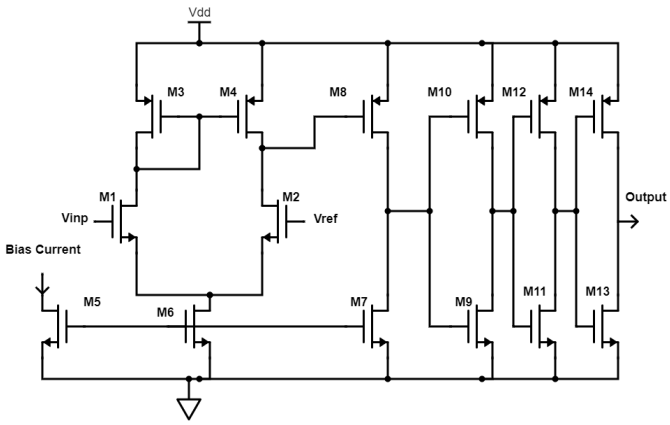


Fig. 7. Schematic diagram of comparator.

phases: training and testing. In the training phase, an Input x is applied to training the neural network to build up a model $F(x)$ for the input data, and this model is used for testing future inputs. In the testing phase, the model is tested with new input x to verify its operation, and the model provides the corresponding output of $Y = F(x)$ for the input x. Output data of the PCA block is applied to the neural network for training. In a training case, the neural network will learn to build a pattern of the fault shape and this training will be presented in a model. A model includes a hypothesis of the output depending on the applied data. Thus, this model is used to classify the output during a testing stage. The testing is the next step after training to test the new inputs. In the testing case, the inputs are applied to FFT and PCA process, and then the result is used for testing the network. The PCA data is applied to the learning network using the model, which is built during a training stage to get the classification.

D. Evaluation

The evaluation of the proposed failure prediction method is provided using metrics in [15], [35]. The evaluation of the proposed method is studied on Tensorflow and Altera Arria 10 GX FPGA 10AX115N2F45E1SG device. Metrics of evaluation is described as follows.

1) True-Positives (TP): True-positive refers to the total number of prediction failures correctly within a specific duration. For example, if the corrected number of prediction is 85 from 100 within 1 minute, then the true-positive will be 85.

2) False-Positives (FP): False-positive is the number of failures which have not occurred but mistakenly predicted within a specific duration.

3) False-Negatives (FN): False-negative is the total number of unpredictable failures which has been occurred within a specific duration. For example, the corrected number of prediction is 85 from 100 within 1 minute, it means the number of unpredictable failures is 15. Thus, the false-negative equals 15.

4) Sensitivity: Sensitivity refers to the ratio between the corrected number of identified failures and the total sum of true-positive and false-negative. Sensitivity can be expressed by:

$$Sensitivity = \frac{TP}{TP + FN} \tag{33}$$

5) Precision: Precision is the ratio between the corrected number of identified failures and the sum of the corrected and uncorrected predicted failures. Thus, precision can be expressed in terms of true-positives and false-positives as:

$$Precision = \frac{TP}{TP + FP} \tag{34}$$

6) Tension: It is the relation between sensitivity and precision, which should be balanced. Increasing precision results in a decreasing sensitivity, so, there is a trade-off between them. The sensitivity improves with low false-negatives which results in increasing false-positives, and it reduces the precision. For example, in preliminary fault screening of a hardware

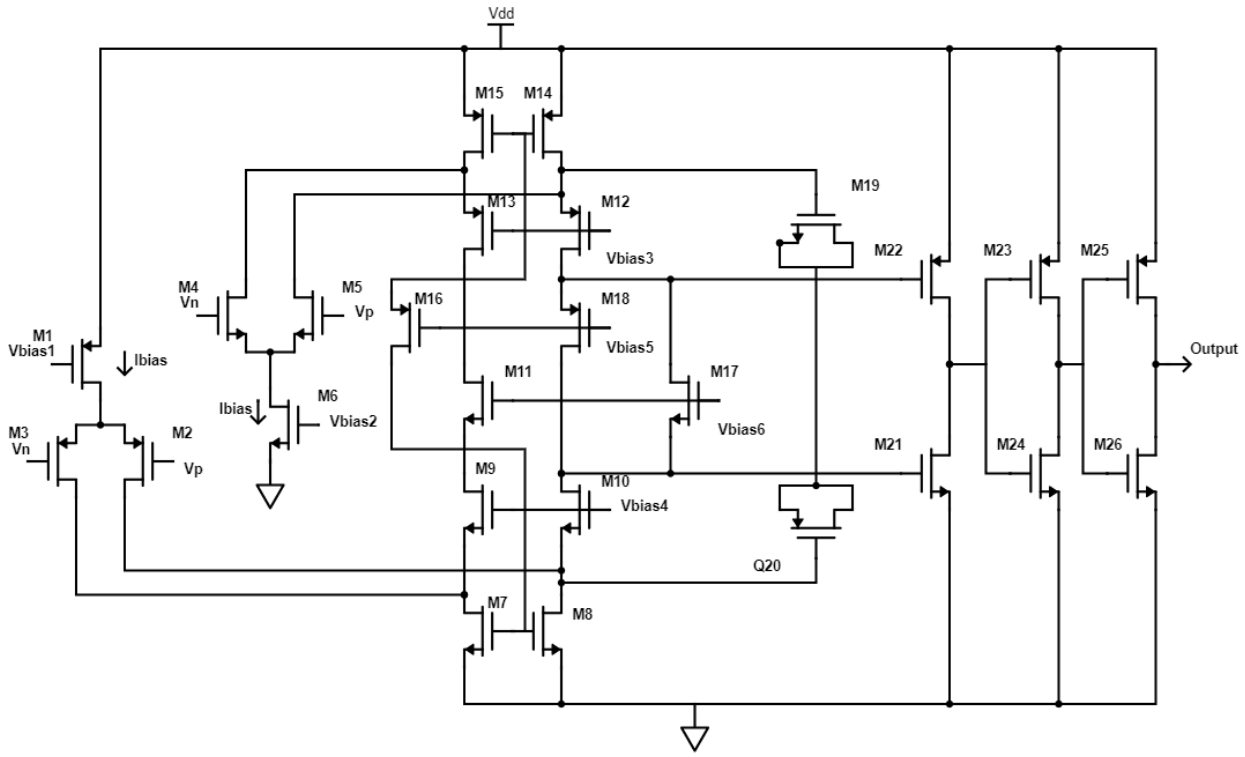


Fig. 8. Schematic diagram of amplifier.

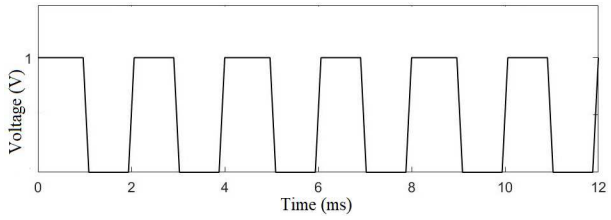


Fig. 9. The comparator input signal.

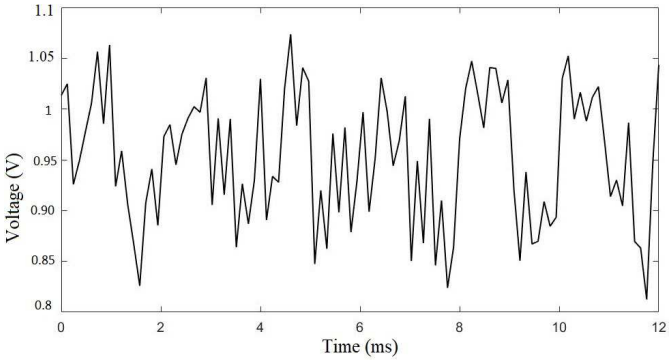


Fig. 10. The comparator output signal of the open-circuit fault state.

system for follow-up maintenance, it would probably need a sensitivity near to "1" to find the hardware section which has the fault, and we can accept a low precision if the follow-up maintenance is not significant. The tension is given by:

$$Tension = \frac{2 * Sensitivity * Precision}{Sensitivity + Precision} \tag{35}$$

7) Specificity: It measures the proportion of actual negatives that are correctly identified.

$$Specificity = \frac{TN}{TN + FP} \tag{36}$$

8) Accuracy: The accuracy of a test is its ability to differentiate classes correctly.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{37}$$

IV. Implementation and Experimental Results

The proposed approach is implemented to predict faults in the comparator and amplifier circuits. The comparator schematic diagram is shown in Fig. 7, and the schematic diagram of the amplifier is shown Fig. 8 [7]. Two basic current sources are used for the operational amplifier. The comparator and amplifier comprise a number of transistors [7], [36], [37], and we focus on transistor fault prediction. The comparator and amplifier circuits are implemented on HSPICE using 45 nm technology with voltage source of 1 V to simulate it in normal mode and faulty mode. The AC analysis for the comparator and amplifier is described as follows. For the comparator the input signal is a square wave as shown in Fig. 9. The reference voltage is used to be 0.5 V, so, if the input is higher than 0.5 V, the output is high. The output is low if the input value is lower than 0.5 V. The resistance load of this circuit is 1 KΩ and a temperature of 300 K. The output signal is the same as the input signal in the normal case. The resulted output for open-circuit and short-circuit faults are shown in Fig. 10 and Fig. 11, respectively.

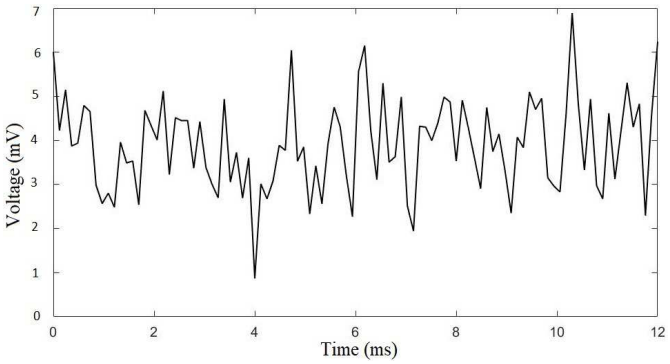


Fig. 11. The comparator output signal of the short-circuit fault state.

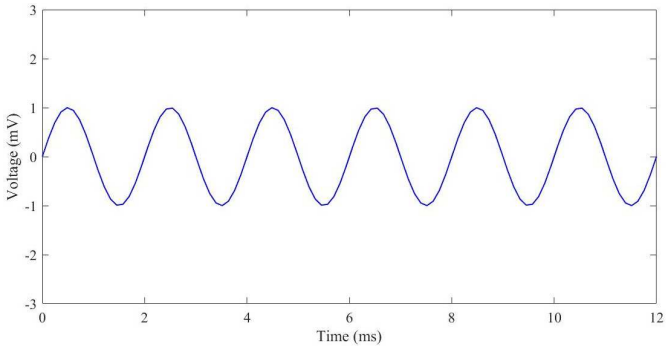


Fig. 12. The amplifier input signal of the normal state.

For the amplifier circuit, a sine wave is used as input as shown in Fig. 12. The load resistance and temperature are 1 KΩ and 300 K, respectively. The output in the normal state is amplified signal as shown in Fig. 13. The output is distorted due to open-circuit and short-circuit faults as shown in Fig. 14 and Fig. 15, respectively. These results indicate the effect of faults on the performance of the circuits. Fault prediction parameters (voltage, current, temperature, delay, noise, EM, etc as mentioned in section III-A) are modified in the simulation, and we extracted the behavior of the transistor to consider aging, short-circuit, and open-circuit faults. We have tried the simulation 100 times to get a more accurate dataset. This dataset is used for learning the proposed approach. In practice, analog signals are read, and this can be periodical. These values are used in FFT operation for the frequency domain purpose. The result is performed by PCA to select the essential data to apply it for CNN learning and classification. This is beneficial in several applications such as biomedical machines, aerospace devices, military machines, etc. The proposed approach is implemented on Tensorflow to show the prediction learning performance. The extracted data is applied to the FFT transformation stage to present data in the frequency domain. In this stage, A sampling frequency f of 60 kHz is used with a measuring time of 0.3 s in each simulation, so the sampling number N is 1800. The sampled output signals are converted by FFT into 0–49 harmonics. The first value which 0 represents the DC component of the output signals. The proposed method is tested 100 times, and the first 42 harmonics are found to be sufficient to meet the demand for the accuracy of fault diagnosis. For voltage parameter, the FFT

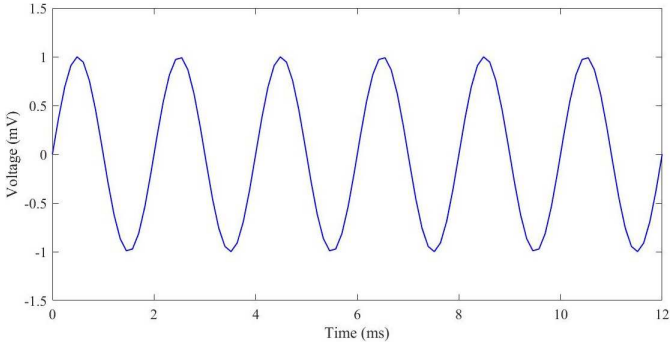


Fig. 13. The Amlpifier output signal of the normal state.

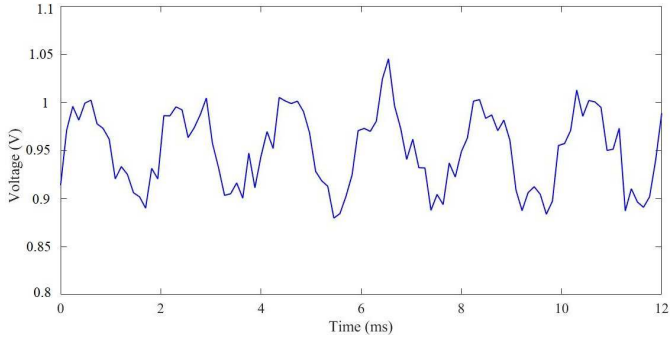


Fig. 14. The Amlpifier output signal of the open-circuit fault state.

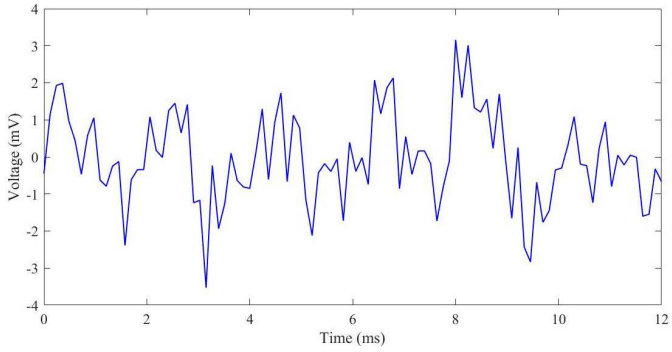


Fig. 15. The Amlpifier output signal of the short-circuit fault state.

of the voltage signal in normal mode without any fault is shown in Fig. 16. In case faulty mode, the FFT of the voltage signal in open-circuit and short-circuit faults, respectively, are presented in Fig. 17, and Fig. 18. For another parameter, the FFT of the current signal in normal mode, open-circuit fault, and short-circuit fault are shown in Fig. 19, Fig. 20, and Fig. 21, respectively. The same procedures are done on the rest of the parameters.

The FFT indicates a difference in the frequency domain of a parameter signal in normal and faulty mode. Therefore, the benefit of FFT is to get a unique signature of each fault, which helps in the learning. The next step after FFT is PCA, in this stage, the role of PCA is to get the most important parameters by transforming the correlated data to uncorrelated data. The vector basis is used in this transformation. A Cumulative Percentage of Variance (CPV) is used to measure the principal components using variation value selected by the first n latent variables. The results show the first Principal

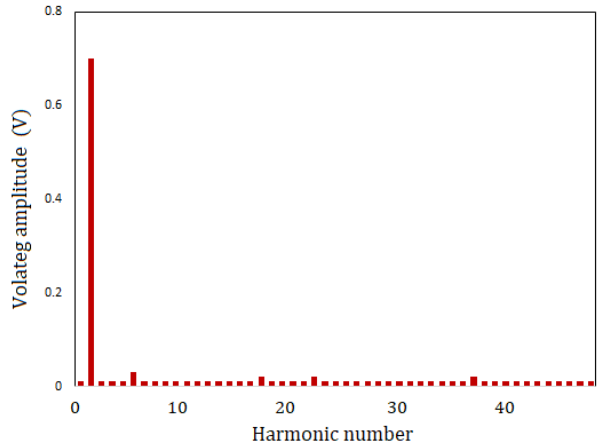


Fig. 16. Harmonics voltage amplitude after FFT without fault.

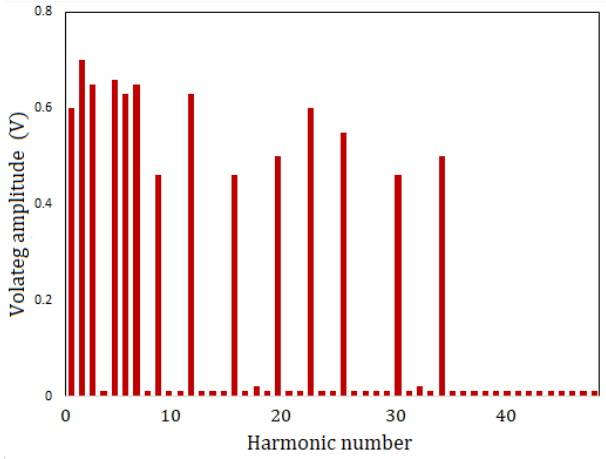


Fig. 17. Harmonics voltage amplitude after FFT of open-circuit fault.

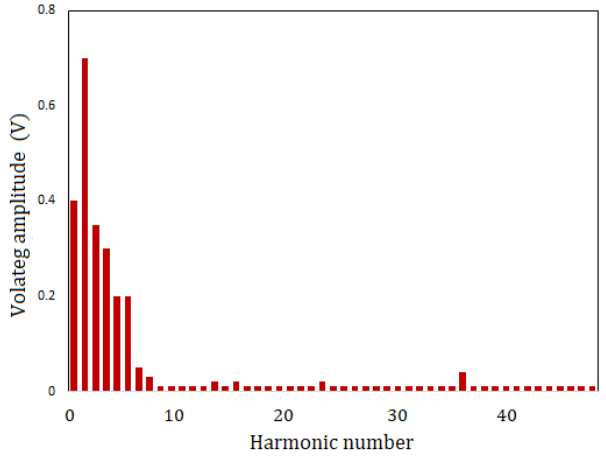


Fig. 18. Harmonics voltage amplitude after FFT of short-circuit fault.

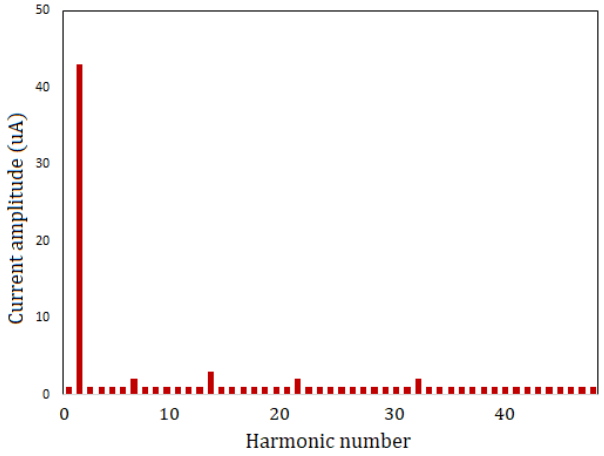


Fig. 19. Harmonics current amplitude after FFT without fault.

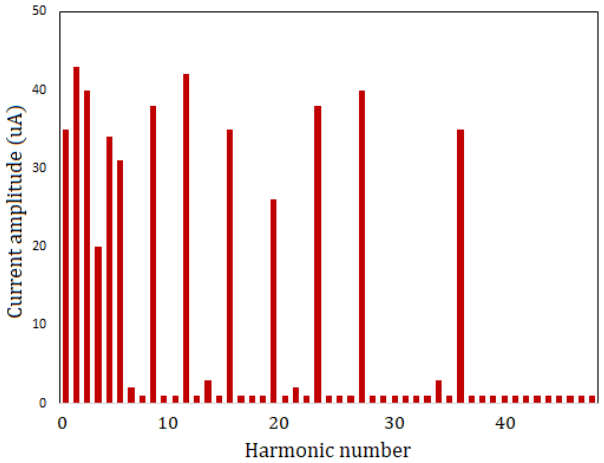


Fig. 20. Harmonics current amplitude after FFT of open-circuit fault.

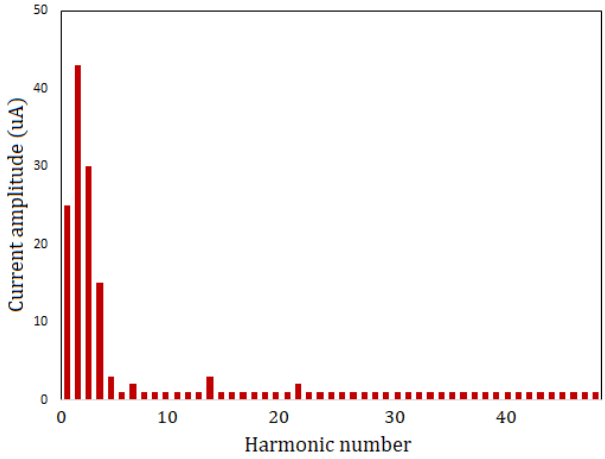


Fig. 21. Harmonics current amplitude after FFT of short-circuit fault.

Component (PC) of PCA contains 84% of the total energy as shown in Fig. 22. The first and second PCs contain 96% of the total energy, and the result will be constant after the 5^{th} PC. Therefore, we can use the first or second component for data presentation.

The CNN role comes after PCA for learning. The CNN is implemented by five convolutional layers with filter size 5×5, five convolutional layers, and five pooling layers with filter size using one step stride. The last stage is the classification stage, which is based on the fully-connected layer, and it classifies the fault at the final output. The fully-connected layer is implemented by three hidden layers, and each layer has 1,000 neurons, and the last output layer has three neurons. The

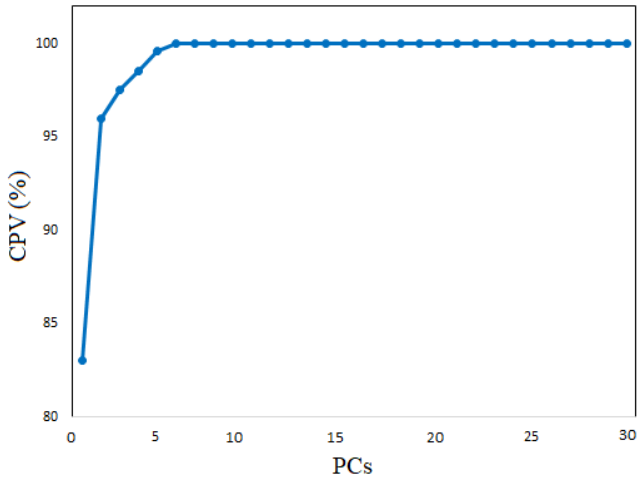


Fig. 22. The simulation result of the principal component.

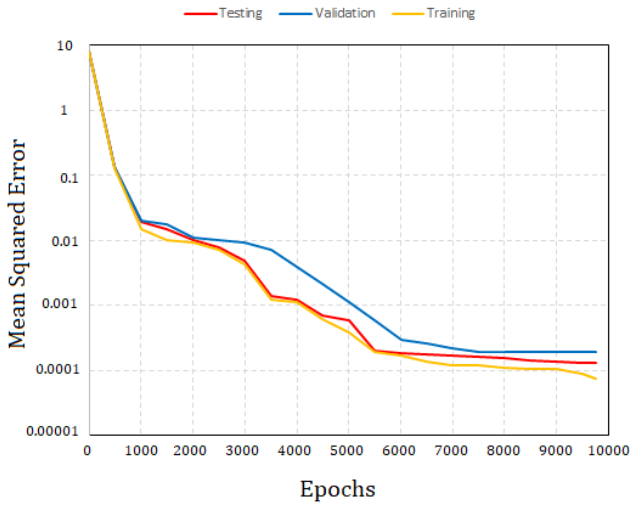


Fig. 23. Mean squared error of the learnning method.

TABLE I
EVALUATION PARAMETER RESULTS

Metrics	Comparator circuit	Amplifier circuit
Sensitivity	91.4%	91.25%
Specificity	94.56%	94.51%
Precision	95.3%	95.27%
Tension	93.3%	93.21%
Accuracy	98.93%	98.91%

TABLE II
COMPARISON BETWEEN THE PROPOSED METHOD AND THE STATE-OF-THE-ART TECHNIQUES

Metric	The proposed method	ANN	SVM
Accuracy	98.93%	88.74%	87.35%
Sensitivity	91.4%	82.3%	81.67%
Specificity	94.56%	84.01%	83.85%
Precision	95.3%	84.91%	84.17%
Power W	1.08	0.84	0.78

simulation results of the total behavior in terms of accuracy, specificity, etc. which are explained in section III-D, are shown in Table. I. The result shows the proposed approach can predict a fault with high-accuracy. A comparison between the proposed method and the state-of-the-art techniques for the comparator circuit is shown in Table. II. The method of using FFT and CNN gives an accuracy of 97.16%, while the proposed method has higher accuracy. The proposed method has the capability to greatly improve diagnostic accuracy, and reduce the running time. Therefore, the proposed method is competitive with the atate-of-the-art technique. We studied the mean square error, which is the average squared difference between the final output of CNN and the target value, and the simulation result is shown in Fig. 23. Furthermore, the regression value is studied which, refers to the correlation between the final output and the actual target value. The result of both mean square error and regression are shown in Table. III. The digital signal processing utilizes Fourier transform, which can be used in the fault prediction process. Therefore, this FFT is not an additional step. On the one hand, the fault prediction using FFT and CNN, provides an accuracy of 98.97% while the training time is 11.9 minutes, and the number of training parameters is 510,317. On the other hand, the proposed method using "FFT + PCA + CNN" provides an accuracy of 98.93% with a training time of 3.2 minutes, and the number of parameters is 25,385. These results show the proposed method provides almost the same accuracy while the number of parameters and the training time are less. The fault prediction method is used as a pre-stage of a self-healing method. The idea is based on recovering future faults. Therefore, the need for fault prediction within minimum time is significant to allow the self-healing method recovers the fault early. If the fault prediction technique spends a longer time, this may affect the time before healing to be shorter. Therefore, we focused to provide a fault prediction method with high speed. Thus, the proposed method is efficient, suitable, and reliable for real-time applications.

TABLE III
LEARNING PERFORMANCE

	Mean Square Error		Regression Value	
Metric	Comparator	Amplifier	Comparator	Amplifier
Training	$7.46\text{x}10^{-5}$	$7.48\text{x}10^{-5}$	$9.8\text{x}10^{-1}$	$9.81\text{x}10^{-1}$
Validation	$1.9\text{x}10^{-4}$	$1.91\text{x}10^{-4}$	$9.8\text{x}10^{-1}$	$9.81\text{x}10^{-1}$
Testing	$1.28\text{x}10^{-4}$	$1.29\text{x}10^{-4}$	$9.8\text{x}10^{-1}$	$9.81\text{x}10^{-1}$

The proposed approach has been implemented on hardware using VHDL and Xilinx Vivado on Altera Arria 10 GX FPGA 10AX115N2F45E1SG device. The simulation results of the hardware implementation in terms of registers, LUTs, DSPs, Buffers, block RAM, Flip Flop (FF), and power are shown in Table. IV for the proposed method, ANN, and SVM. These are the consumed hardware resources for the proposed, ANN, and SVM methods. The hardware resources consumption for FFT and PCA are shown in Table. V and Table. VI, respectively. These results present the used resources, which are the consumed resources, and utilization (Util.) which is the ratio of used resources to the total available resources. TThe proposed method has a delay of 350 ms. The power consumption of the proposed method is 1.08 W, which is comparable with the ANN, SVM methods 0.84 W and 0.78 W, respectively. The

TABLE IV
COMPARISON OF RESOURCES UTILIZATION ON HARDWARE IMPLEMENTATION BETWEEN THE PROPOSED METHOD AND OTHER TRADITIONAL WORK

	The proposed method		ANN		SVM	
Logic Utilizing	Used	Util.	Used	Util.	Used	Util.
Number of slice registers	9834	9%	7607	7%	7388	6.8%
Number of Slice LUTs	6498	11%	4686	8.02%	4639	7.92%
Number of BUFs	8	5.92%	6	4.4%	6	4.4%
Number of DSPs	34	8.83%	24	6.23%	22	5.71%
Number of Block RAM	16	10.81%	11	6.87%	10	6.76%
Memory LUTs	387	2.24%	341	1.98%	331	1.92%
FF	11342	12.14%	9291	9.95%	9189	9.84%
Power (W)	1.08	—	0.84	—	0.78	—

TABLE V
HARDWARE UTILIZATION OF FFT

Logic Utilizing	**Used**	**Available**	**Utilization**
Number of slice registers	685	5472	12.5%
Number of Slice LUTs	1312	10960	12.2%
Number of BUFs	10	135	7.4%
Number of DSPs	16	425	3.76%
Number of Block RAM	4	39	10.25%
Memory LUTs	387	17269	2.24%
FF	1108	10960	10.10%

TABLE VI
HARDWARE UTILIZATION OF PCA

Logic Utilizing	**Used**	**Available**	**Utilization**
Number of slice registers	3218	107266	3%
Number of Slice LUTs	1462	28520	5%
Number of BUFs	2	115	1.7%
Number of DSPs	18	185	9.7%
Number of Block RAM	7	105	6.66%
Memory LUTs	125	9271	1.3%
FF	425	21257	2%

operating frequency is 120 MHZ. The proposed approach will be very beneficial for fault tolerance where the fault prediction allows fault tolerance or self-healing method to fix this fault early without affecting the system performance. It is desirable to apply a self-healing in aerospace hardware devices. The cost of fixing faults using external interference is high. Therefore, the proposed approach will be beneficial to this field.

The result shows the proposed approach has high-accuracy to predict fault, which can be used to fix this fault by self-healing or isolating the defective components and keeping the system works using the available components. The proposed method in a system, a self-healing method is not triggered until getting a signal indicative of a fault. If the system has a future fault, the proposed method predicts a fault within 27 clock cycles, and it provides the type of fault and coordinates to the self-healing method. Once the self-healing method gets this information, it performs a self-healing mechanism to recover this fault [38], [39]. Based on our experimental observations, the proposed method of transistor-level fault prediction can be applied to more complex circuits accurately. The cost of repair may vary depending on a unit that may need replacement. For very complex systems, however, system-level fault detection and healing may be more economical such as in [15]. Continued research in this area will shed more light on the usage, accuracy, and tradeoff of fault prediction and healing at different levels of design abstractions. The proposed method utilizes the existing FFT of a system, if present, to avoid adding an additional FFT block. To save power consumption, the proposed method can be applied periodically instead of running all the time. This period can be selected to be less than or equal to the prediction time of the proposed method to not lose the prediction.

V. CONCLUSION

This paper presented an approach of early transistor fault prediction using FFT, PCA, and CNN. The proposed approach utilizes the fault signature in the frequency domain by FFT. The FFT result is applied to PCA to get the most important values with less dimension. The CNN stage is used subsequently to complete the final feature presentation and fault classification. The proposed approach is tested on the comparator and amplifier circuits which are implemented using 45 nm technology to study the transistor fault in terms of aging, short-circuit, and open-circuit faults. The proposed approach is implemented using Tensorflow, and the result shows the proposed approach could predict a fault by the accuracy of 98.93%. The proposed method contributes to providing a high accuracy to a diagnostic fault within reasonable time. The proposed method is compared with the state-of-the-art methods, and the result shows the proposed method has a more accurate result with a lower error. Finally, the proposed approach is implemented in hardware VHDL on Altera Arria 10 GX FPGA 10AX115N2F45E1SG device, and it consumes 1.08 W.

REFERENCES

[1] U. E. Avci *et al.*, "Transistor threshold voltage variation optimization," U.S. Patent App. 16080914, Jan. 17, 2019.
[2] B. Yuce, N. F. Ghalaty, and P. R. Schaumont, "Microprocessor fault detection and response system," U.S. Patent 10452493, Oct. 22, 2019.
[3] K. Khalil, O. K. Eldash, and M. Bayoumi, "A novel approach towards less area overhead self-healing hardware systems," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2017, pp. 1585–1588.
[4] C. Saritha, V. Sukanya, and Y. N. Murthy, "ECG signal analysis using wavelet transforms," *Bulg. J. Phys.*, vol. 35, no. 1, pp. 68–77, Feb. 2008.
[5] T. Marisa *et al.*, "Pseudo asynchronous level crossing ADC for ECG signal acquisition," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 2, pp. 267–278, Apr. 2017.
[6] R. Sehgal, F. Van Der Goes, and K. Bult, "A 13-mW 64-dB SNDR 280-MS/s pipelined ADC using linearized integrating amplifiers," *IEEE J. Solid-State Circuits*, vol. 53, no. 7, pp. 1878–1888, Jul. 2018.
[7] F. Mohd-Yasin, M. Yap, and M. Reaz, "CMOS instrumentation amplifier with offset cancellation circuitry for biomedical application," *WSEAS Trans. Circuits Syst.*, vol. 6, no. 1, pp. 171–174, 2007.
[8] T. Wang, J. Qi, H. Xu, Y. Wang, L. Liu, and D. Gao, "Fault diagnosis method based on FFT-RPCA-SVM for cascaded-multilevel inverter," *ISA Trans.*, vol. 60, pp. 156–163, Jan. 2016.

[9] A. Kumar and A. P. Singh, "Transistor level fault diagnosis in digital circuits using artificial neural network," *Measurement*, vol. 82, pp. 384–390, Mar. 2016.

[10] W. Sleszynski, J. Nieznanski, and A. Cichowski, "Open-transistor fault diagnostics in voltage-source inverters by analyzing the load currents," *IEEE Trans. Ind. Electron.*, vol. 56, no. 11, pp. 4681–4688, Nov. 2009.

[11] K. Khalil, O. Eldash, A. Kumar, and M. Bayoumi, "Self-healing hardware systems: A review," *Microelectron. J.*, vol. 93, Nov. 2019, Art. no. 104620.

[12] N. Raj, J. Mathew, G. Jagadanand, and S. George, "Open-transistor fault detection and diagnosis based on current trajectory in a two-level voltage source inverter," *Procedia Technol.*, vol. 25, pp. 669–675, 2016.

[13] K. Khalil, O. Eldash, A. Kumar, and M. Bayoumi, "Economic LSTM approach for recurrent neural networks," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 11, pp. 1885–1889, Nov. 2019.

[14] S. Hoo-Chang *et al.*, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1285–1298, May 2016.

[15] K. Khalil, O. Eldash, A. Kumar, and M. Bayoumi, "Intelligent fault-prediction assisted self-healing for embryonic hardware," *IEEE Trans. Biomed. Circuits Syst.*, early access, May 19, 2020, doi: 10.1109/TBCAS.2020.2995784.

[16] M. Noda, S. Kajihara, Y. Sato, K. Miyase, X. Wen, and Y. Miura, "On estimation of NBTI-induced delay degradation," in *Proc. 15th IEEE Eur. Test Symp.*, May 2010, pp. 107–111.

[17] G. Gielen *et al.*, "Emerging yield and reliability challenges in nanometer CMOS technologies," in *Proc. Design, Autom. Test Eur.*, Mar. 2008, pp. 1322–1327.

[18] J. Lienig, "Introduction to electromigration-aware physical design," in *Proc. Int. Symp. Phys. Design (ISPD)*, 2006, pp. 39–46.

[19] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "The impact of technology scaling on lifetime reliability," in *Proc. Int. Conf. Dependable Syst. Netw.*, 2004, p. 177.

[20] M. Yi, Y. Gan, Z. Huang, and H. Liang, "Co-mitigating circuit PBTI and HCI aging considering NMOS transistor stacking effect," in *Proc. Int. Symp. Integr. Circuits (ISIC)*, Dec. 2016, pp. 1–5.

[21] T. Nigam, A. Kerber, and P. Peumans, "Accurate model for time-dependent dielectric breakdown of high-k metal gate stacks," in *Proc. IEEE Int. Rel. Phys. Symp.*, Apr. 2009, pp. 523–530.

[22] E. Maricau and G. Gielen, "Transistor aging-induced degradation of analog circuits: Impact analysis and design guidelines," in *Proc. ESSCIRC (ESSCIRC)*, Sep. 2011, pp. 243–246.

[23] P.-H. Chen *et al.*, "An 80 mV startup dual-mode boost converter by charge-pumped pulse generator and threshold voltage tuned oscillator with hot carrier injection," *IEEE J. Solid-State Circuits*, vol. 47, no. 11, pp. 2554–2562, Nov. 2012.

[24] E. Maricau, P. De Wit, and G. Gielen, "An analytical model for hot carrier degradation in nanoscale CMOS suitable for the simulation of degradation in analog IC applications," *Microelectron. Rel.*, vol. 48, nos. 8–9, pp. 1576–1580, Aug. 2008.

[25] R. P. Bastos and F. S. Torres, "Effects of transient faults in integrated circuits," in *On-Chip Current Sensors for Reliable, Secure, and Low-Power Integrated Circuits*. Cham, Switzerland: Springer, 2020, pp. 1–16, doi: 10.1007/978-3-030-29353-6_1.

[26] S. Chen, X. Li, and J. Xiong, "Series arc fault identification for photovoltaic system based on time-domain and time-frequency-domain analysis," *IEEE J. Photovolt.*, vol. 7, no. 4, pp. 1105–1114, 2017.

[27] P. Duhamel and M. Vetterli, "Fast Fourier transforms: A tutorial review and a state of the art," *Signal Process.*, vol. 19, no. 4, pp. 259–299, Apr. 1990.

[28] I. T. Jolliffe, *Principal Component Analysis*. Evanston, IL, USA: Routledge, 2002.

[29] Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee, and C. Wu, "Proper orthogonal decomposition and its applications—Part I: Theory," *J. Sound Vib.*, vol. 252, no. 3, pp. 527–544, 2002.

[30] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[31] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 3304–3308.

[32] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, vol. 8689. Cham, Switzerland: Springer, 2014, pp. 818–833, doi: 10.1007/978-3-319-10590-1_53.

[33] K. Khalil, O. Eldash, A. Kumar, and M. Bayoumi, "An efficient approach for neural network architecture," in *Proc. 25th IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Dec. 2018, pp. 745–748.

[34] Z. Zhang, "Artificial neural network," in *Multivariate Time Series Analysis in Climate and Environmental Research*. Springer, 2018, pp. 1–35, doi: 10.1007/978-3-319-67340-0.

[35] W. Zhu *et al.*, "Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations," in *Proc. NESUG: Health Care Life Sci.*, vol. 19. Baltimore, MD, USA, p. 67, 2010.

[36] K. Khalil, M. Abbas, and M. Abdelgawad, "A low propagation delay dispersion comparator for low cost level-crossing ADCs," in *Proc. 8th IEEE Design Test Symp.*, Dec. 2013, pp. 1–5.

[37] K. Kozmin, J. Johansson, and J. Kostamovaara, "A low propagation delay dispersion comparator for a level-crossing AD converter," *Analog Integr. Circuits Signal Process.*, vol. 62, no. 1, p. 51, 2010.

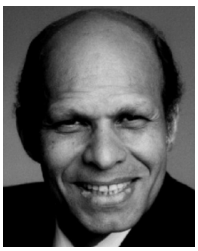
[38] K. Khalil, O. Eldash, and M. Bayoumi, "A cost-effective self-healing approach for reliable hardware systems," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5.

[39] K. Khalil, O. Eldash, A. Kumar, and M. Bayoumi, "Flexible self-healing router for reliable and high-performance Network-on-Chips architecture," in *Proc. 31st IEEE Int. System-on-Chip Conf. (SOCC)*, Sep. 2018, pp. 152–157.

Kasem Khalil (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering and electronics and communications engineering from Assiut University, Asyut, Egypt, in 2009 and 2014, respectively, and the second M.Sc. degree in computer engineering from the Center of Advanced Computer Studies (CACS), University of Louisiana at Lafayette, USA, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include VLSI, microelectronics, reconfigurable hardware, self-healing hardware systems, machine learning, hardware accelerators, network-on-chip, intelligent hardware systems, and the Internet of Things.

Omar Eldash (Member, IEEE) received the B.Sc. degree in electrical engineering from Fayoum University, Faiyum, Egypt, in 2009, and the M.Sc. degree in computer engineering from the Center of Advanced Computer Studies (CACS), University of Louisiana at Lafayette, USA, in 2016, where he is currently pursuing the Ph.D. degree. His research interests include system-on-chip, reconfigurable hardware, hardware accelerators, dynamic hardware, machine learning, and hardware accelerators.

Ashok Kumar (Senior Member, IEEE) received the B.Tech. degree from IIT (BHU), India, and the M.S. and Ph.D. degrees in computer engineering from the University of Southwestern Louisiana. He is currently an Associate Professor with the School of Computing and Informatics, University of Louisiana at Lafayette. His research interests include intelligent and energy-efficient software and hardware systems.



Magdy Bayoumi (Life Fellow, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Cairo University, Egypt, the M.Sc. degree in computer engineering from Washington University at St. Louis, St. Louis, and the Ph.D. degree in electrical engineering from the University of Windsor, ON. He is currently the Department Head of the Electrical and Computer Engineering Department, University of Louisiana at Lafayette, Lafayette, LA, USA. His research interests include VLSI design and architectures, digital signal processing, and wireless ad hoc and sensor networks. He was a recipient of the 2009 IEEE Circuits and Systems Meritorious Service Award and the IEEE Circuits and Systems Society 2003 Education Award. He was the Vice President for Conferences of the IEEE Circuits and Systems Society.