

Received October 22, 2021, accepted November 7, 2021, date of publication November 15, 2021, date of current version November 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3128134

# An Efficient Heartbeats Classifier Based on Optimizing Convolutional Neural Network Model

EMAN MAGHAWRY<sup>1</sup>, TAREK F. GHARIB<sup>1</sup>, RASHA ISMAIL<sup>1</sup>,  
AND MOHAMMED J. ZAKI<sup>2</sup>, (Fellow, IEEE)

<sup>1</sup>Faculty of Computer and Information Sciences, Ain Shams University, Cairo 11566, Egypt

<sup>2</sup>Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Corresponding author: Eman Maghawry (eman\_amin@cis.asu.edu.eg)

**ABSTRACT** Recently, deep learning models have emerged as promising methods for the diagnosis of different diseases. Cardiac disease is among the leading life-threatening diseases on a global scale. The aim of this paper is to propose an optimized Convolutional Neural Network (CNN) model for the classification of electrocardiogram (ECG) heartbeat data. The proposed ECG classification approach is designed with an optimal CNN configuration to classify cardiac arrhythmias quickly and effectively. Finding an optimal configuration for the CNN hyperparameters is time-consuming and needs extensive experimentation. To overcome this challenge, we present an optimization step for the proposed CNN model using a customized genetic algorithm. It provides an automatic suggestion for the best hyperparameter settings of the proposed CNN. The challenge in utilizing the genetic algorithm is that its operators need to be customized to handle our problem domain. Our approach accepts raw ECG signals without any preprocessing steps, which has benefit in saving the computation time. Our approach also provides a resampling step to ensure generalization, to better handle imbalanced ECG classes. Experiments show promising results of our proposed approach against other approaches whose CNN hyperparameters setting depended on numerous trials, requiring extensive ECG feature extraction steps, and do not consider imbalanced classes. The performance of our proposed approach is better than other existing methods both in terms of higher classification accuracy (98.45%), and lower computational complexity.

**INDEX TERMS** Cardiac disease, convolutional neural network, hyperparameter selection, genetic algorithms.

## I. INTRODUCTION

Deep learning has proven successful in solving various problems in the medical applications area. Cardiac disease is one of the most common causes of death globally [1]. Using deep learning as a diagnostic tool for heart disease can reduce the risk of various heart-related complications, such as heart failure and stroke. ECG is a test tool that represents via a waveform signal the rhythm of cardiac electrical activity using skin-attached sensors [2]. Analysis of the ECG data is beneficial because it can be used to identify different heart diseases such as arrhythmia.

Arrhythmia is the abnormality of the heart rate [3]. It is categorized as life-threatening or non-life threatening. Patients with arrhythmia may feel that their heartbeats are too slow,

too fast, or have irregular beats. However, arrhythmia diagnostics require professional cardiologists to accurately classify irregular ECGs, which is a time-consuming process due to the large differences in ECG beats. Therefore, automatic heart disease diagnosis and classification techniques have been used in recent decades to assist cardiologists in making better decisions to improve the accuracy of disease recognition [4].

Early methods mainly used machine learning methods for the automatic classification of heartbeats but required important steps such as feature selection, feature extraction, and classification for the ECG time series data. The authors of [5]–[7] employed several methods of feature extraction and machine learning classification techniques [8], [9]. However, it is difficult to capture and extract all relevant features from a time series such as the ECG.

The associate editor coordinating the review of this manuscript and approving it for publication was Hiu Yung Wong<sup>1</sup>.

Recently, deep learning techniques have played an important role in solving various classification problems and have achieved outstanding performance for accuracy of time series classification. Deep learning is a kind of machine learning technique that consists of multiple regression layers within a hierarchical architecture, with information processing taking place across these successive layers [10]. Deep learning techniques recognize more complex features from the data due to the number of hidden layers. Consequently, the results of the classification of deep learning models are generally superior compared to other machine learning techniques [11] for complex data.

The goal of this paper is to develop an approach that improves the accuracy and performance of classification of ECG signals based on deep learning.

### A. RELATED WORK

A convolutional neural network (CNN) is a popular deep learning model that has been successfully used in many fields such as computer vision and image processing [12], [13]. The work in [14] focuses on physiological data for signals in the form of 1D signal records that sample data points over time. The CNN can capture the position of invariant patterns when analyzing physiological signals [15]–[17]. In addition, the CNN is characterized by a lower sensitivity to noise, which increases its ability to obtain useful information in case of noise in the signals [18]. Such abilities are obtained from the deep hierarchical structure, as features can be represented and learned in a more abstract manner within the layers of the network.

CNNs have been used in for classifying heartbeat arrhythmia and detecting other heart diseases [18]–[22]. The work in [16] introduced a technique to automatically detect atrial fibrillation based on deep convolutional neural networks. They can learn the features of data by designing a computational model using a combination of multiple processing layers [23]. In [18], an encoder and decoder approaches are used to construct a deep network structure of twenty-seven layers. The input signals are decreased to low dimensional vectors within the encoder, and the signals are regenerated via the decoder. Their approach introduces representations of both high- and low-level signals in the hidden layers of the model. In [19] the segmented signals of ECG are processed by a CNN model with 11 layers. Its structure is comprised of four convolutional layers, three fully connected layers and four max pooling layers. The authors of [24], [25] proposed cardiac disease diagnosis system consisting of long short-term memory and CNNs. [10], [26], [27] use a 1D-CNN classifier that can classify cardiac arrhythmia automatically. Their model can process heartbeats directly without additional feature extraction steps, and they demonstrate their competitiveness in classification accuracy compared to traditional methods.

Some CNN-based research has applied preprocessing to ECG signals, such as wavelet transforms [21], [31], [32] or denoising of ECG signals [33]. In [34], they proposed a method of normalizing length records to produce records

of equal length to be submitted to their CNN model. They developed a layer of 31 one-dimension residual CNNs to diagnose five types of heartbeats [4]. Stationary wavelet transform (SWT) and short-term Fourier transform (STFT) were utilized to preprocess ECG segments to generate a 2-dimensional matrix input for a deep CNN in [16]. In [21] they denoised the baseline wander from the ECG signal before submitting the signals to their deep learning model.

Limitations of most previous research works are that they use pre-processing steps such as noise removal or feature extraction for ECG signals instead of inputting raw data directly into the deep learning model, which increases the computation time. In addition, most research uses pre-defined and manual parameter value tuning for their proposed CNN models. These hyperparameters of the CNN model must be specified prior to the training process. The tuning of hyperparameter values is a complementary part of the construction of any deep learning model. These hyperparameters can include the batch size, the number of layers, the cells in each layer, the type of activation function and so on. The choice of these values can have an important impact on the training process. It can affect the final accuracy of the model by causing the resulting model to under fit or overfit. Therefore, these parameter values should be optimally determined. To find this optimal configuration is a challenge and a time-consuming process, due to the selection of parameter values from a combinatorial set of choices.

Our approach to selection of the optimal parameters is based on a genetic algorithm (GA), which is a heuristic algorithmic approach inspired by evolutionary theory. There are several studies that have applied genetic algorithms to CNNs, such as [28]–[30]. In [28], they used a CNN for diagnosing breast cancer; the weights of the CNN were optimized using a genetic algorithm (GA). They evaluated their GA-CNN model against three different optimizers; their results indicate that their GA-based model performed similarly to the Adam optimizer. Also, in [29] they used a CNN for diagnosing Alzheimer's disease; they used genetic algorithms to optimize the network architecture of the CNN. Their version of the genetic algorithm is specific to the given input dataset, and it can select the best-performing CNN architecture, including the optimization algorithm and activation function. In [30] the authors introduced an approach for building optimal deep neural networks (e.g., CNNs, LSTMs, RNNs) using genetic algorithms suited for specific classification tasks. Their GA approach selects the required number of layers and nodes in each layer, and it automatically finds the best suited deep neural network architecture. The evaluation of the network is based on the accuracy of classification. To our knowledge, we are not aware of existing work that applies GA to the task of ECG signals classification. The difference between our GA approach against other approaches is that we developed a specific method for individual encoding of the GA which was suited to our problem domain. The scheme of an individual encoding in the GA population is presented to adapt to the time-series ECG data classification task. Instead of manually

adjusting the CNN hyperparameters settings, we designed a customized genetic algorithm to select the best CNN hyperparameters from multiple configurations to improve the classification accuracy and avoid time consuming.

## B. OUR CONTRIBUTIONS

Considering the shortcomings of existing CNN-based methods for the automatic classification of heartbeats, we aim to develop an approach that improves accuracy and computational performance in the classification of ECG signals. Our proposed CNN model has a smaller number of layers to avoid complexity. In addition, our proposed approach accepts raw input ECG signals without any preprocessing steps since the CNN model can extract features during the learning process. Also, the best configuration for the proposed CNN hyperparameters is selected using a customized genetic algorithm. We chose CNNs for our ECG classification approach because of its strengths in automatic and adaptive learning of informative features. Especially for the time-series ECG data, CNNs are able to automatically extract predictive temporal patterns, without the need to preprocess or hand-tune the data or model. Also, it has a major advantage in terms of high noise resistance compared to other methods. Furthermore, we customize the genetic algorithm due to its advantage in finding good solutions in a large space of solutions, especially due to its fitness function, which uses a non-knowledge-based optimization process for evaluation. In our case, the genetic algorithm selects the best hyperparameter settings from multiple configurations to improve the accuracy of the ECG signals classification task.

In summary, improvements in the classification performance of ECGs are introduced through the following contributions:

- We design a CNN model with low structural complexity using only two convolutional blocks.
- Our approach does not need any pre-processing steps for the ECGs Signals.
- We develop a custom genetic algorithm to effectively select the best hyperparameters for our CNN model. The scheme of individual encoding in the population of our GA is introduced to adapt to our problem domain.
- We resample the dataset to address the imbalanced class problem. A simplified resampling method was used to avoid computational costs. It based on up sampling the minority classes and down sampling the majority class.

## II. MATERIAL AND METHODS

### A. PROPOSED CNN ARCHITECTURE

Our proposed CNN model automatically classifies ECG signals to detect cardiac arrhythmias without applying any preprocessing or feature extraction steps. A 1D-CNN is convenient for time-series data such as ECG signals from which samples are taken at regular intervals [4]. Our CNN structure is mainly a combination of two convolutional blocks (six layers), fully connected layers (two layers), and an output softmax classification layer, as shown in Fig. 1.

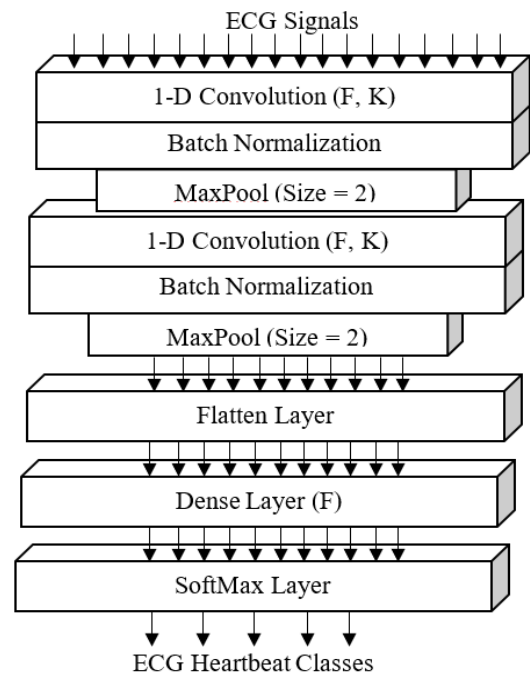


FIGURE 1. The proposed CNN structure.

We configured our CNN model with these settings to simplify its structure as much as possible to avoid any additional computational cost, and reduce the structural complexity. In each convolutional block, there are  $F$  filters with size  $K$  for the kernels. The level of the pooling layer, also called subsampling or down sampling, exploits the feature space of the previous layers to generate a new feature space by fetching the maximum values within a specified region from previous feature space. The main aim of this layer is to make the dimensionality of the feature maps half the actual size from the previous layer to decrease the number of computations and avoid the problem of data overfitting. The batch normalization layer allows better stability and performance of the CNN structure by avoiding vanishing gradients [35]. It is important to note that CNNs are ideally suited for automatically extracting informative and predictive features from the 1D time-series ECG data. The important input subsequences are captured by the  $F$  filters (or masks) that act as useful signals for the downstream prediction task and are trained via backpropagation. Furthermore, max-pooling extracts information across the filters, and then the second convolution layer extracts higher-level features from the 1D data. What is particularly impressive is that these features are learned without human supervision and can deliver comparable or often better predictive performance compared to hand-tuned features, as in our case.

The CNN has a flattened layer which transforms the input features of the previous layer to the appropriate output size to be passed to a dense-connected layer of ( $F$ ) units.

In addition, the last layer employs softmax [36] which determines the class by normalizing the results of the previous

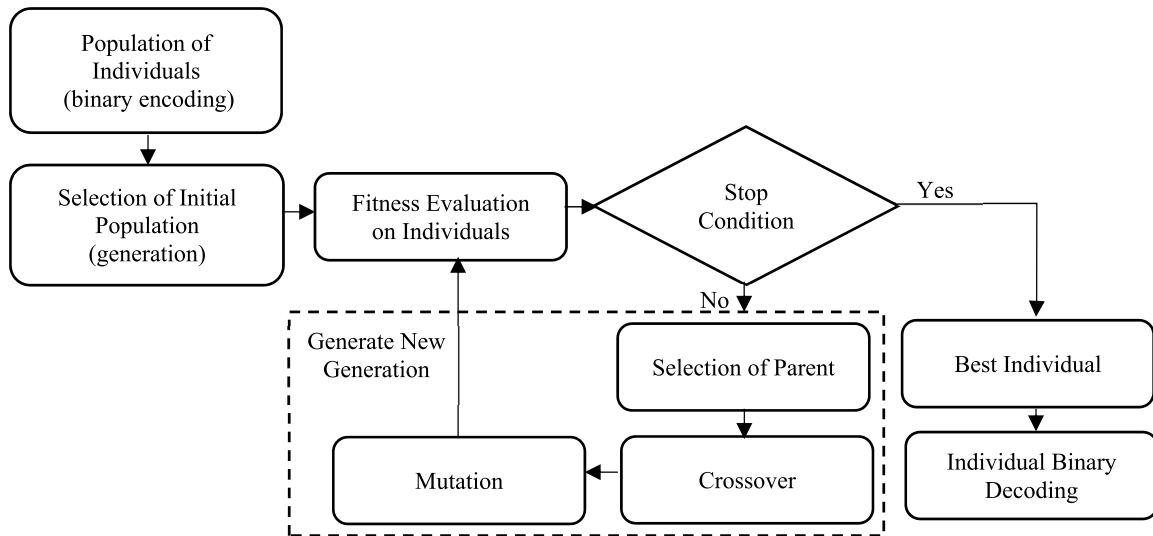


FIGURE 2. The architecture of the genetic algorithm.

layers and computing a probability distribution for the various classes [24].

The customized CNN structure involves multiple settings of the following hyperparameters [34]

- Number of convolutional blocks (CNN Depth) ( $N = 2$ ).
- Filters ( $F_i$ ) and kernels sizes ( $K_i$ ) in each convolutional block ( $i = 1, \dots, N$ ).
- Number of epochs: iterations over the training dataset.
- Batch size: samples segmented into several smaller batches.
- Max-pooling size: gradual down sampling of the feature space.
- Activation function and optimizer for CNN: Activations include rectified linear unit (*relu*), sigmoid, exponential linear unit (*elu*), hyperbolic tangent (*tanh*), and so on. Optimizers include root mean square propagation, stochastic gradient descent (*sgd*), adaptive gradient algorithm, adaptive moment estimation (*Adam*), etc.

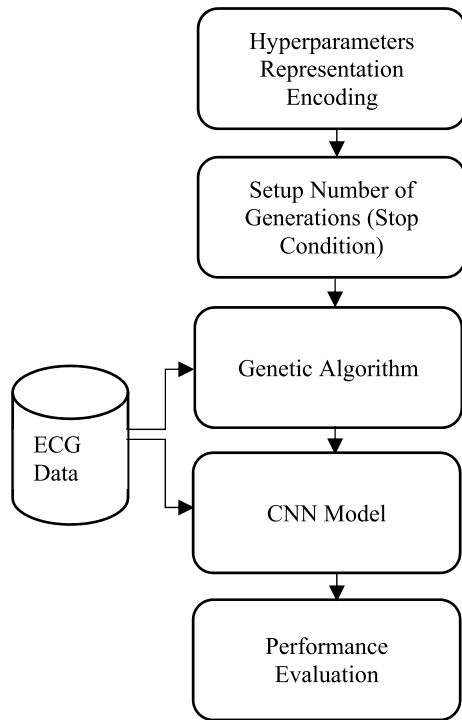
## B. OPTIMIZATION FOR CNN HYPERPARAMETERS

Hyperparameter optimization can be defined as how to decrease the number of trials while finding the best hyperparameters to obtain the optimum CNN settings. The relationship between model performance and parameter settings should be specified for hyperparameter optimization.

One of the issues with neural models like the CNN proposed above is the need to tune the hyperparameters of the network, spanning the different elements outlined above, e.g., the number and size of filters, the learning rate, the activation function, and the optimizer. It is difficult to tune these by hand, and typically one can perform a grid search or other systematic search to sample the best parameters, which can be time consuming. Instead, we propose to use a genetic algorithm to accomplish this task.

The fundamental ingredient of the genetic algorithm is a population of individuals (with chromosomes composed of genes). Each individual represents a solution to the given problem and has a fitness value that measures the quality of the solution. The algorithm as shown in Fig. 2 operates on a randomized population of individuals. Binary encoding used for individuals of the population. Then, random selection of the initial population, known as a generation, is employed. Thereafter, the algorithm performs a fitness evaluation on this generation. Selection, crossover and mutation operators based on the fitness value are applied to produce a new generation. New individuals are added for each generation according to the genetic operators of the population. Operators of the genetic algorithm such as crossover, mutation and selection are used to make the population's characteristics more diverse. The crossover operator randomly distributes genes of both parents to generate a new one. The mutation operator modifies the genes of some individual strings. The main aim of crossover and mutation operators is to explore new regions in the search space. The selection operator chooses the most suitable individuals within the population, considering that their offsprings have a better likelihood of survival in the next generation due to their high fitness value. Individuals with the worst fitness values are deleted, and more offspring members are produced from individuals with higher fitness values. The previous process is repeated until certain stopping criteria are met.

The genetic algorithm needs to define a suitable representation of a solution for a given problem, which is considered one of its challenges. The appropriate selection of values of the mutation and crossover operators are critical and sometimes complex to find the best solution, since they have no prior knowledge of the function structure that needs to be optimized.



**FIGURE 3.** Design of the genetic algorithm for optimal CNN configuration.

Fig. 3 presents the design of a genetic algorithm based on our problem domain to calculate the optimal CNN hyperparameters giving consequently the highest classification accuracy. These selected hyperparameters are then used to build and train the optimal CNN model. We developed a certain method of individual encoding. Each individual in the population is a candidate solution that introduces CNN hyperparameters for our proposed architecture.

The fitness function is based on computing the classification accuracy during the training of a separate validation set. The stopping criteria is based on the number of generations. Number of iterations before termination depends on the complexity and problem type. In our case, it was sufficient, in terms of calculating classification accuracy, to set the number of generations to 50 in order to avoid computational complexity.

For our genetic algorithm, the scheme of encoding of an individual in the population is presented in Fig. 4. An individual is encoded in the genetic algorithm as a binary string, where a collection of bits in each individual string represents some characteristics of the solution. The population size should be reasonable, since if it is large, the search space is increased, and concomitantly the computational load. Therefore, the initial population consists of 30 individuals. There are five hyperparameters: filter size, kernel size, learning rate, activation function and optimizer; each hyperparameter value is represented by 4 binary genes. For example, the types of activation functions are categorized into numbers while these numbers are encoded into their corresponding binary numbers. Thus, each individual is represented by a sequence of 20 binary genes as shown in Fig. 4.

We used uniform crossover, where each bit in the new offspring was randomly chosen among the two parents by picking parent gene values at each selected position. After choosing the two parents by the selection method, we generate a random binary vector of individual length. To create the offspring, in case the value is 1, then the gene from the first parent is chosen. Otherwise, the gene from the second parent is selected [37]. All resulting offsprings from crossover are then passed to the mutation process. We used probabilistic bitwise mutation where a bit is turned from 0 to 1 or vice versa at a random position. The probability of mutation normally increases with the population size to encourage the diversity of individuals to obtain global optimal solutions. However, the use of crossover and selection does not guarantee optimality. Therefore, mutation is used to alleviate this concern by providing novel offsprings that are different from parents, which is required to skip the local search space. Tournament selection is used, where individuals of the population are randomly selected to perform several tournaments among themselves. The individual winner in each tournament is chosen for the crossover operator. The crossover operator uses a uniform probability to select two individuals, then chooses the individual with the highest fitness value.

### III. EXPERIMENTAL ENVIRONMENT

#### A. DATASET

We used the MIT-BIH arrhythmia dataset [38] from the Physionet website [39]. It contains 48 samples of ECG signals obtained from 47 subjects and was used in the performance evaluation of the optimized CNN. The sampling frequency of ECG signals is 360 Hz. The length of the data for each ECG sample is about 30 minutes. The proposed approach can classify ECG heartbeats in five classes, as recommended by the Association for Advancement of Medical Instrumentation (AAMI) [40]. Table 1 summarizes the heartbeat classes.

#### B. THE RESAMPLING METHOD

We can observe a big difference in the number of ECG heartbeats within different classes. This class imbalance can lead to the suppression of the minority class in favor of the majority class during the training process. This can lead to a false increase in the overall accuracy because the majority class dominates the learning process. Previous research didn't consider the class imbalance problem when classifying ECG heartbeats. We address this point by applying resampling to the dataset for better generalization. We use a simplified resampling method to avoid computational costs.

We up sampled the minority classes to randomly duplicate their observations to reinforce the signal. We also down sampled the majority class by randomly deleting subjects from the majority class to stop its signal from dominating the learning process. Table 1 shows that the number of segments within all classes is the same size after balancing, which is beneficial for verifying the outcome of the learning process.



Binary Encoded Representation (20 bits)

Filters size	Kernel Size	Learning Rate	Activation Function	Optimizer
--------------	-------------	---------------	---------------------	-----------

FIGURE 4. Encoding scheme of an individual.

TABLE 1. Heartbeat classes.

Heartbeat Class	Number of ECG Heartbeats	Original Class Fraction	Class Fraction after Balancing
Normal (N)	72471	82.2%	20%
Supra ventricular ectopic (S)	2223	2.5%	20%
Ventricular ectopic (V)	5788	6.6%	20%
Fusion (F)	641	0.7%	20%
Unknown (Q)	6431	7.3%	20%

TABLE 2. Confusion matrix.

Confusion Matrix							
True Classification	Predicted classification						
		N	S	V	F	Q	Total
	N	TC <sub>00</sub>	TC <sub>10</sub>	TC <sub>20</sub>	TC <sub>30</sub>	TC <sub>40</sub>	ΣTC <sub>i0</sub>
	S	TC <sub>01</sub>	TC <sub>11</sub>	TC <sub>21</sub>	TC <sub>31</sub>	TC <sub>41</sub>	ΣTC <sub>i1</sub>
	V	TC <sub>02</sub>	TC <sub>12</sub>	TC <sub>22</sub>	TC <sub>32</sub>	TC <sub>42</sub>	ΣTC <sub>i2</sub>
	F	TC <sub>03</sub>	TC <sub>13</sub>	TC <sub>23</sub>	TC <sub>33</sub>	TC <sub>43</sub>	ΣTC <sub>i3</sub>
	Q	TC <sub>04</sub>	TC <sub>14</sub>	TC <sub>24</sub>	TC <sub>34</sub>	TC <sub>44</sub>	ΣTC <sub>i4</sub>
	Total	ΣTC <sub>0i</sub>	ΣTC <sub>1i</sub>	ΣTC <sub>2i</sub>	ΣTC <sub>3i</sub>	ΣTC <sub>4i</sub>	ΣΣTC <sub>ji</sub>

For learning, the dataset was split into 80% training and 20% test datasets. The training set is further divided keeping aside a 20% validation set to calculate the loss and accuracy to evaluate the performance of the proposed approach. The validation set is used for the hyperparameter selection via the genetic algorithm.

### C. EXPERIMENTAL SETUP

The proposed approach was developed on the Colab platform [41]; experiments were performed on a computer equipped with a 2.20GHz Intel Xeon CPU and 12GB NVIDIA Tesla K80 GPU.

### D. PERFORMANCE METRICS

Our approach's performance was evaluated using the following confusion matrix as presented in Table 2.

For a class  $n$ , we define the following parameters based on the confusion matrix:

- True Positive ( $TP_n$ ) =  $TC_{nn}$  (1)

- True Negative ( $TN_n$ ) =  $(\sum_n TC_{nn}) - TC_{nn}$  (2)

- False Positive ( $FP_n$ ) =  $(\sum_i TC_{ni}) - TC_{nn}$  (3)

- False Negative ( $FN_n$ ) =  $(\sum_j TC_{jn}) - TC_{nn}$  (4)

The metrics of precision and recall of each class are defined below, as well as the Macro F1-score and accuracy:

- Precision: the percentage of total samples that are predicted as positive.

$$\text{Precision}_n (\%) = \frac{TP_n}{TP_n + FP_n} \times 100 \quad (5)$$

- Recall: the percentage of positives correctly classified.

$$\text{Recall}_n (\%) = \frac{TP_n}{TP_n + FN_n} \times 100 \quad (6)$$

- F1-score: the balance between the precision and recall metrics.

$$\text{F1-Score} = 2 \times \frac{\text{Recall}_n \times \text{Precision}_n}{\text{Recall}_n + \text{Precision}_n} \quad (7)$$

- Macro F1-score: used to evaluate the performance with multiple classes

$$\text{Macro F1} = \frac{1}{N} \sum_{i=0}^N \text{F1-score} \quad (8)$$

- Accuracy: the percentage of correctly predicted samples across all inputs.

$$\text{Accuracy}(\%) = \frac{\sum TC_{nn}}{\sum \sum TC_{ji}} \quad (9)$$

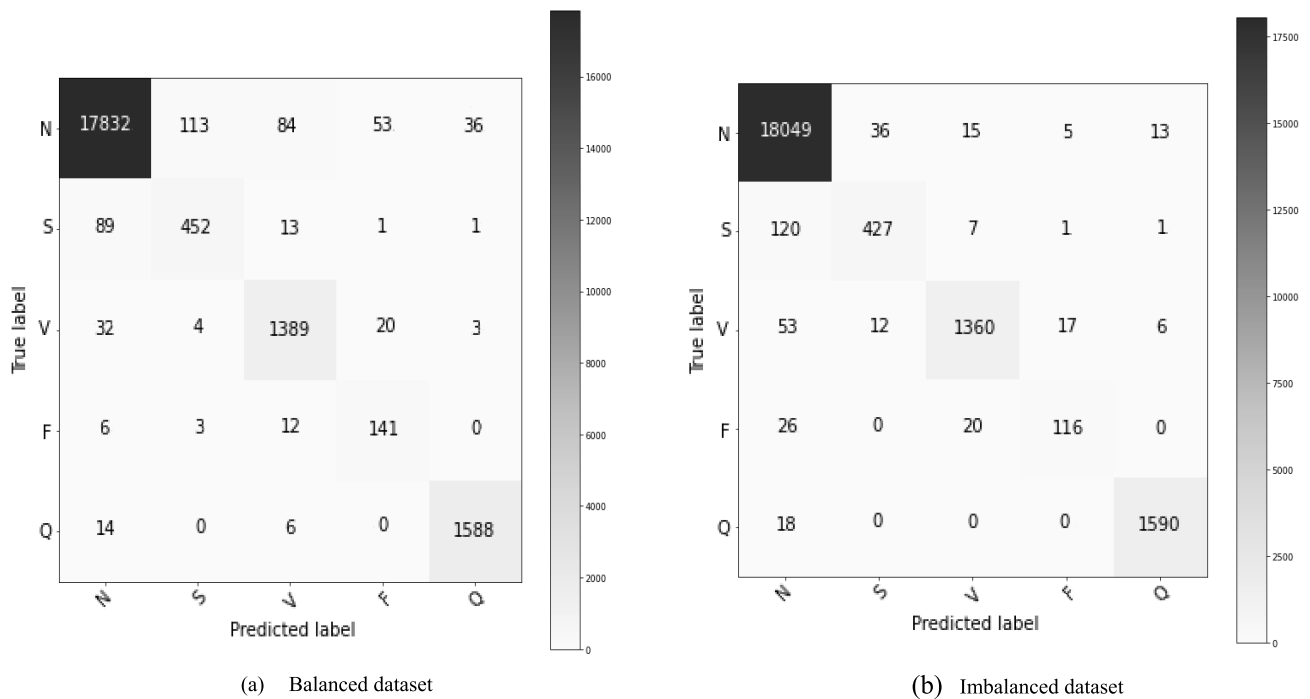
## IV. RESULTS AND DISCUSSION

The fitness function of the genetic algorithm was based on calculating the accuracy of the proposed approach with the values of different individuals that reflect different hyperparameters values. The best individual suggested by the genetic algorithm after running 50 generations was decoded to get the parameter values. The number of individuals is 30. We used probabilistic bitwise mutation where a bit is turned from 0 to 1 or vice versa at a random position with a mutation probability of 0.1 [37].

The optimal values suggested by the most efficient genetic algorithm are presented in Table 3. The genetic algorithm was applied on the dataset before and after applying resampling. Fig. 5 shows a visual representation of the confusion matrix with and without resampling, to evaluate the performance

**TABLE 3.** The optimal hyper-parameters for the CNN.

Hyper-parameter	Imbalanced dataset	Balanced dataset
	Best value	Best value
Number of Filters	70	128
Kernel Size	5	10
Learning Rate	0.0001	0.0001
Batch Size	32	32
Activation	Relu	Relu
Optimizer	Adam	Adam

**FIGURE 5.** Normalized average confusion matrix.

of the proposed approach for the testing set. The matrix highlights large number of correct responses in dark squares and small number of incorrect responses in light-coloured squares for each heartbeat class. Table 4 shows the absolute values of the F1-score, recall and precision for our proposed approach, with and without resampling.

Table 4 demonstrates that our approach has the ability to accurately classify the majority class (N) with high accuracy, on both balanced and imbalanced datasets. It can be observed that the recall for the class N reached 100% in the imbalanced dataset because it is the majority among other classes. However, the other minor classes had a more significant increase in recall results with the balanced dataset. Although the number of correctly classified N classes reduced with a balanced dataset, the number of abnormal classes incorrectly classified as normal heartbeats reduced even further.

However, for all these measured metrics, the recall of our approach with balanced dataset is better than for the imbalanced dataset.

Table 5 presents the comparison between our approach with state-of-the-art approaches that also use CNNs for ECG classification. The comparison was carried out in terms of the number of layers utilized in their CNN model and the number of classes. All comparative approaches employ the same original dataset (imbalanced). The performance metrics are used to evaluate our approach against other proposed approaches. Many approaches do not calculate the F1-score, which is regarded as an important indicator of imbalanced data. Classification time required for the ECG signal heartbeat is also reported for our proposed approach and other approaches.

It can be concluded from Table 5 that our approach with balanced dataset has higher classification accuracy for five heartbeat classes compared to the other approaches proposed in [27], [42]–[44] that depend on predefined or manual hyperparameters tuning of their CNN models. Table 5 also reports the recall achieved by our approach against the other approaches. The recall presents the percentage of positives

**TABLE 4.** Performance metrics for our approach.

Class	Imbalanced dataset			Balanced dataset		
	Precision	Recall	F1-score	Precision	Recall	F1-score
N	99.0	100	99.0	99.0	98.0	99.0
S	90.0	77.0	83.0	79.0	81.0	80.0
V	97.0	94.0	95.0	92.0	96.0	94.0
F	83.0	72.0	77.0	66.0	87.0	75.0
Q	99.0	99.0	99.0	98.0	99.0	98.0
Accuracy	98.0	98.0	98.0	98.0	98.0	98.0
Macro-F1	94.0	88.0	91.0	87.0	92.0	89.0

**TABLE 5.** Comparison of our proposed approach to existing approaches.

Literatures	#Layers	# Heartbeat Classes	Database	Precision	Recall	F1 score	Classification Time(sec)	Accuracy (%)
Zubair et al. [27]	8	5	MIT-BIH	-	-	-	168	92.7
Acharya et al. [42]	9	5	MIT-BIH	91.54	96.71	-	-	94.03
Sellami et al. [43]	9	5	MIT-BIH	88.51	90.90	-	64.18	88.34
Xiang et al. [44]	2-level 1D CNN	5	MIT-BIH	-	-	-	-	97.8
Xu et al. [33]	9	5	MIT-BIH	98.0	98.0	98.0	264	99.41
Taissir et al. [45]	12	5	MIT-BIH	-	-	-	184	98.41
Our approach	9	5	MIT-BIH	98.0	98.0	98.0	150	98.45

correctly classified. It can be observed from the results that our approach with balanced dataset has higher recall for five heartbeat classes compared to the other approaches [42], [43]. The work in [44] extracts RR interval features and proposes a 2-level 1D CNN, with multi-layer perceptrons (MLP) for classifying ECG signals. However, their approach is not entirely accurate because of the high variability of morphological features of beat-to-beat, which vary depending on circumstance and time. The approach in [45] has classification accuracy close to our approach, but they use more layers (12 instead of 9 in ours). This causes additional computational cost and increases the structural complexity of their model. It is also important to note that they use an ECG segmentation step before ECG classification, which increases the complexity and is time consuming. Also, it can be noted from Table 5 that the classification time required for our proposed approach is less than other methods. This is because our approach does not require any pre-processing steps for the ECG signals and due to the smaller number of layers in our CNN model. There are other studies such as [35] that have classification accuracy close to our proposed approach, but they use more layers, and their model classifies only three classes of heartbeat while our approach can classify five heartbeat classes.

Furthermore, it can be observed from the previous results that [33] reaches 99.4% accuracy which is slightly better than

our proposed approach. However, in their approach a raw ECG signal is broken down by Daubechies wavelet. They further segment the ECG signal into heartbeats by exploiting the information of the R-peaks position annotated by the dataset. Their CNN is composed of 9 layers, comprised of 4 convolutional layers, 2 fully connected layers, single softmax layer and 2 subsampling layers. On the other hand, the advantage of our proposed approach over [33] is that our model avoids additional computational cost and time for extensive preprocessing steps of ECG signals as inputs to the CNN. The classification time required for the ECG signal heartbeats using our proposed approach is between 144 and 156 seconds, unlike the work in [33], which needs more time – between 260 and 268 seconds. Fig. 6 reports the mean of the classification time and accuracy of 10 training repeats. It also reports the standard deviation for overall accuracy. The mean accuracy was 98.1 and 99.4, the standard deviation was 0.17 and 0.02 of our approach and [33], respectively.

In summary, our proposed approach has the following advantages. First, there is no need to do any pre-processing steps for the ECG signals. Computational complexity has been reduced due to the smaller number of convolutional blocks in our CNN model. In addition, the hyperparameters of the CNN model have been optimized by a customized genetic algorithm which avoids extensive trials. Also, our approach has taken into consideration the imbalanced nature



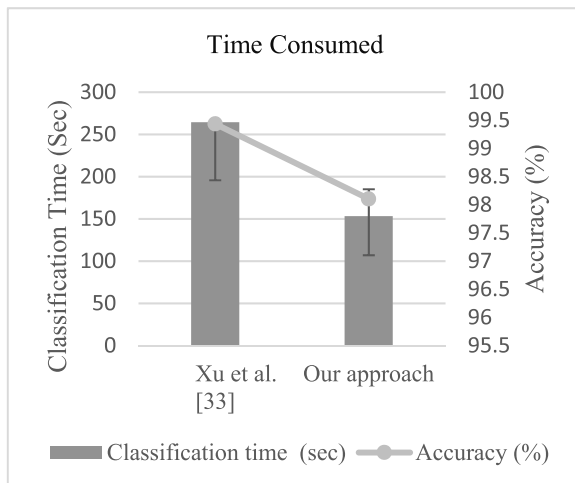


FIGURE 6. Time analysis.

of the ECG dataset. Our proposed method with the genetic algorithm can be combined with other models, for example, in their feature extraction step to explore more features to handle many classification classes. Finally, our proposed CNN architecture had higher classification performance than most of the approaches mentioned.

## V. CONCLUSION

Finding an optimal configuration for a deep learning model for a specific problem domain is a challenge. The objective of this paper is to propose an efficient approach using an optimized CNN which can classify cardiac heartbeats in five heartbeats classes (N, S, V, F, and Q). The best configuration of hyperparameter values for the CNN model was optimized by utilizing a customized genetic algorithm. The ECG signals do not require any pre-processing steps for our approach. Resampling method is applied to the dataset to overcome its imbalanced nature. The proposed approach achieved an overall classification accuracy of 98.45% for five classes of heartbeats. Compared with other studies, the results of our approach are promising because of its lower structural complexity and computational time, its high accuracy in the classification of ECG heartbeats, and number of detected heartbeats classes. In future work, we will seek to improve the classification accuracy of diagnosing different heart diseases by combining our proposed method with other models. Also, we can extend our customized genetic algorithm with other models by setting their hyperparameters to fit within the individual encoding scheme.

## REFERENCES

- [1] Y. Ping, C. Chen, L. Wu, Y. Wang, and M. Shu, "Automatic detection of atrial fibrillation based on CNN-LSTM and shortcut connection," *Healthcare*, vol. 8, no. 2, p. 139, May 2020.
- [2] S. Vijayarangan, B. Murugesan, R. Vignesh, S. Preejith, J. Joseph, and M. Sivaprakasam, "Interpreting deep neural networks for single-lead ECG arrhythmia classification," in *Proc. 42nd Annu. Int. Conf. Eng. Med. Biol. Soc. (EMBC)*, Montreal, QC, Canada, Jul. 2020, pp. 300–303.
- [3] G. Sannino and G. De Pietro, "A deep learning approach for ECG-based heartbeat classification for arrhythmia detection," *Future Gener. Comput. Syst.*, vol. 86, pp. 446–455, Sep. 2018.
- [4] Z. Li, D. Zhou, L. Wan, J. Li, and W. Mou, "Heartbeat classification using deep residual convolutional neural network from 2-lead electrocardiogram," *J. Electrocardiology*, vol. 58, pp. 105–112, Jan. 2020.
- [5] F. A. Elhaj, N. Salim, A. R. Harris, T. T. Swee, and T. Ahmed, "Arrhythmia recognition and classification using combined linear and nonlinear features of ECG signals," *Comput. Methods Programs Biomed.*, vol. 127, pp. 52–63, Apr. 2016.
- [6] C.-H. Lin, Y.-C. Du, and T. Chen, "Adaptive wavelet network for multiple cardiac arrhythmias recognition," *Expert Syst. Appl.*, vol. 34, no. 4, pp. 2601–2611, 2008.
- [7] I. Güler and E. D. Übeyli, "ECG beat classifier designed by combined neural network model," *Pattern Recognit.*, vol. 38, no. 2, pp. 199–208, 2005.
- [8] A. Batra and A. V. Jawa, "Classification of arrhythmia using conjunction of machine learning algorithms and ECG diagnostic criteria," *Int. J. Biol. Biomed.*, vol. 1, no. 1, pp. 1–7, 2016.
- [9] C. K. Roopa and B. S. Harish, "A survey on various machine learning approaches for ECG analysis," *Int. J. Comput. Appl.*, vol. 163, no. 9, pp. 25–33, Apr. 2017.
- [10] Ö. Yildirim, P. Plawiak, R.-S. Tan, and U. R. Acharya, "Arrhythmia detection using deep convolutional neural network with long duration ECG signals," *Comput. Biol. Med.*, vol. 102, no. 1, pp. 411–420, Nov. 2018.
- [11] S. Kusuma and J. D. Udayan, "Analysis on deep learning methods for ECG based cardiovascular disease prediction," *Scalable Comput., Pract. Exper.*, vol. 21, no. 1, pp. 127–136, Mar. 2020.
- [12] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, p. 9, Jul. 2016.
- [13] X. Yang, Q. Hu, and S. Li, "Electrocardiogram classification of lead convolutional neural network based on fuzzy algorithm," *J. Intell. Fuzzy Syst.*, vol. 38, no. 4, pp. 3539–3548, Apr. 2020.
- [14] B. Rim, N.-J. Sung, S. Min, and M. Hong, "Deep learning in physiological signal data: A survey," *Sensors*, vol. 20, no. 4, p. 969, Feb. 2020.
- [15] J. H. Tan, Y. Hagiwara, W. Pang I. Lim, and M. Adam, "Application of stacked convolutional and long short-term memory network for accurate identification of CAD ECG signals," *Comput. Biol. Med.*, vol. 94, pp. 19–26, Mar. 2018.
- [16] Y. Xia, N. Wulan, K. Wang, and H. Zhang, "Detecting atrial fibrillation by deep convolutional neural networks," *Comput. Biol. Med.*, vol. 93, pp. 84–92, Feb. 2018.
- [17] Q. Zhang, D. Zhou, and X. Zeng, "HeartID: A multiresolution convolutional neural network for ECG-based biometric human identification in smart health applications," *IEEE Access*, vol. 5, pp. 11805–11816, 2017.
- [18] O. Yildirim, T. R. San, and U. R. Acharya, "An efficient compression of ECG signals using deep convolutional autoencoders," *Cognit. Syst. Res.*, vol. 52, pp. 198–211, Dec. 2018.
- [19] U. R. Acharya, H. Fujita, S. L. Oh, U. Raghavendra, J. H. Tan, M. Adam, A. Gertych, and Y. Hagiwara, "Automated identification of shockable and non-shockable life-threatening ventricular arrhythmias using convolutional neural network," *Future Gener. Comput. Syst.*, vol. 79, pp. 952–959, Feb. 2018.
- [20] U. R. Acharya, H. Fujita, O. S. Lih, M. Adam, J. H. Tan, and C. K. Chua, "Automated detection of coronary artery disease using different durations of ecg segments with convolutional neural network," *Knowl.-Based Syst.*, vol. 132, pp. 62–71, Sep. 2017.
- [21] U. R. Acharya, H. Fujita, S. L. Oh, Y. Hagiwara, J. H. Tan, and M. Adam, "Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals," *Inf. Sci.*, vols. 415–416, pp. 190–198, Nov. 2017.
- [22] M. M. Al Rahhal, Y. Bazi, M. Al Zuair, E. Othman, and B. BenJdira, "Convolutional neural networks for electrocardiogram classification," *J. Med. Biol. Eng.*, vol. 38, no. 6, pp. 1014–1025, 2018.
- [23] H. Goh, N. Thome, M. Cord, and J.-H. Lim, "Learning deep hierarchical visual feature coding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2212–2225, Dec. 2014.
- [24] S. L. Oh, E. Y. K. Ng, R. S. Tan, and U. R. Acharya, "Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats," *Comput. Biol. Med.*, vol. 102, no. 1, pp. 278–287, Nov. 2018.

- [25] A. Picon, U. Irusta, A. Álvarez-Gila, E. Aramendi, F. Alonso-Atienza, C. Figuera, U. Ayala, E. Garrote, L. Wik, J. Kramer-Johansen, and T. Eftestøl, "Mixed convolutional and long short-term memory network for the detection of lethal ventricular arrhythmia," *PLoS ONE*, vol. 14, no. 5, May 2019, Art. no. e0216756.
- [26] S. Savalia and V. Emamian, "Cardiac arrhythmia classification by multi-layer perceptron and convolution neural networks," *Bioengineering*, vol. 5, no. 2, p. 35, May 2018.
- [27] M. Zubair, J. Kim, and C. Yoon, "An automated ECG beat classification system using convolutional neural networks," in *Proc. 6th Int. Conf. IT Converg. Secur. (ICITCS)*, Sep. 2016, pp. 1–5.
- [28] K. Davoudi and P. Thulasiraman, "Evolving convolutional neural network parameters through the genetic algorithm for the breast cancer classification problem," *Simulation*, vol. 97, pp. 511–527, Jun. 2021.
- [29] S. Lee, J. Kim, H. Kang, D.-Y. Kang, and J. Park, "Genetic algorithm based deep learning neural network structure and hyperparameter optimization," *Appl. Sci.*, vol. 11, no. 2, p. 744, Jan. 2021.
- [30] D. Shukla and U. Singh, "A generic framework for evolution of deep neural networks using genetic algorithms," in *Proc. Int. Conf. Innov. Comput. Commun. Adv. Intell. Syst. Comput.*, vol. 1165, D. Gupta, A. Khanna, S. Bhattacharyya, A. Hassanien, S. Anand, and A. Jaiswal, Eds. Singapore: Springer, 2021, pp. 117–127.
- [31] D. Li, J. Zhang, Q. Zhang, and X. Wei, "Classification of ECG signals based on 1D convolution neural network," in *Proc. IEEE 19th Int. Conf. E-Health Netw., Appl. Services (Healthcom)*, Oct. 2017, pp. 1–6.
- [32] U. B. Baloglu, M. Talo, O. Yildirim, R. S. Tan, and U. R. Acharya, "Classification of myocardial infarction with multi-lead ECG signals and deep CNN," *Pattern Recognit. Lett.*, vol. 122, pp. 23–30, May 2019.
- [33] X. Xu and H. Liu, "ECG heartbeat classification using convolutional neural networks," *IEEE Access*, vol. 8, pp. 8614–8619, 2020.
- [34] C.-H. Hsieh, Y.-S. Li, B.-J. Hwang, and C.-H. Hsiao, "Detection of atrial fibrillation using 1D convolutional neural network," *Sensors*, vol. 20, no. 7, p. 2136, Apr. 2020.
- [35] R. Avanzato and F. Beritelli, "Automatic ECG diagnosis using convolutional neural network," *Electronics*, vol. 9, no. 6, p. 951, Jun. 2020.
- [36] W. Liu, Y. Wen, M. Scut, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 507–516.
- [37] S. Loussaief and A. Abdelkrim, "Convolutional neural network hyperparameters optimization based on genetic algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 10, pp. 252–266, 2018.
- [38] R. Mark and G. Moody, *MIT-BIH Arrhythmia Database Directory*. Cambridge, MA, USA: Massachusetts Institute of Technology, 1988.
- [39] *Physio Database*. Accessed: Dec. 15, 2020. [Online]. Available: <http://www.physionet.org/>
- [40] *Testing and Reporting Performance Results of Cardiac Rhythm and St Segment Measurement Algorithms*, document ANSI/AAMI EC38, American National Instrumentation, 1998.
- [41] *Colab Platform*. Accessed: Jan. 4, 2021. [Online]. Available: <https://colab.research.google.com/notebooks/intro.ipynb#recent=true>
- [42] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, and A. Gertych, "A deep convolutional neural network model to classify heartbeats," *Comput. Biol. Med.*, vol. 89, pp. 389–396, Oct. 2017.
- [43] A. Sellami and H. Hwang, "A robust deep convolutional neural network with batch-weighted loss for heartbeat classification," *Expert Syst. Appl.*, vol. 122, pp. 75–84, May 2019.
- [44] Y. Xiang, J. Luo, T. Zhu, S. Wang, X. Xiang, and J. Meng, "ECG-based heartbeat classification using two-level convolutional neural network and RR interval difference," *IEICE Trans. Inf. Syst.*, vol. E101.D, no. 4, pp. 1189–1198, 2018.
- [45] T. F. Romdhane, H. Alhichri, R. Ouni, and M. Atri, "Electrocardiogram heartbeat classification based on a deep convolutional neural network and focal loss," *Comput. Biol. Med.*, vol. 123, Aug. 2020, Art. no. 103866.



**EMAN MAGHAWRY** received the M.Sc. degree in computer sciences from Ain Shams University, Cairo, Egypt, in 2016. She is currently an Assistant Lecturer with the Information Systems Department, Faculty of Computer and Information Sciences, Ain Shams University. Her current research interests include data mining, machine learning, and deep learning.



**TAREK F. GHARIB** received the Ph.D. degree in theoretical physics from Ain Shams University, Cairo, Egypt, in 1994. He is currently a Full Professor of information systems at Ain Shams University, where he is also the Head of the Information Systems Department, Faculty of Computer and Information Sciences. He has over 70 publications. His research interests include developing novel data mining and machine learning techniques, especially for applications in text mining, social networks, bioinformatics, and data analytics. He received the National Science Foundation Award, in 2001.



**RASHA ISMAIL** received the Ph.D. degree in data mining and data warehousing from Ain Shams University, Egypt, in 2009. She was the Director of the credit hours programs, from 2013 to 2019. She has contributed in implementing of the current credit-hour curriculum for the undergraduate programs of bioinformatics and software engineering at Ain Shams University, where she is currently a Professor and the Vice Dean of postgraduate studies and research affairs with the Faculty of Computer and Information Sciences. In addition, she is also the Director of the Ain Shams Portal. Her current research interests include data science, software engineering, information retrieval, big data analytics, data mining, and bioinformatics.



**MOHAMMED J. ZAKI** (Fellow, IEEE) received the Ph.D. degree in computer science from the University of Rochester, in 1998. He is a Professor of computer science with Rensselaer Polytechnic Institute (RPI). He is also the Associate Department Head and the Graduate Program Director of the CS Department, RPI. He has over 250 publications and six patents, including the *Data Mining and Machine Learning* textbook (2nd Edition) (Cambridge University Press, 2020). His research interests include developing novel data mining and machine learning techniques, especially for applications in text mining, social networks, bioinformatics, and personal health. He is an ACM Distinguished Scientist. He was a recipient of the National Science Foundation CAREER Award and the Department of Energy Early Career Principal Investigator Award. He is the Founding Co-Chair of the BIODDD series of workshops. He was the Program Co-Chair of SDM'08, SIGKDD'09, PAKDD'10, BIBM'11, CIKM'12, ICDM'12, IEEE BigData'15, and CIKM'18. He is currently serving on the Board of Directors for ACM SIGKDD. He is currently an Associate Editor of *Data Mining and Knowledge Discovery*.

...