IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 17, 2020, accepted January 12, 2021, date of publication February 2, 2021, date of current version February 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3056509

A Proof-of-Concept of Ultra-Edge Smart IoT Sensor: A Continuous and Lightweight Arrhythmia Monitoring Approach

SADMAN SAKIB[1], MOSTAFA M. FOUDA[2,3], (Senior Member, IEEE), ZUBAIR MD. FADLULLAH[1,4], (Senior Member, IEEE), NIDAL NASSER[5], (Senior Member, IEEE), AND WALEED ALASMARY[6], (Senior Member, IEEE)

[1]Department of Computer Science, Lakehead University, Thunder Bay, ON P7B 5E1, Canada
[2]Department of Electrical and Computer Engineering, College of Science and Engineering, Idaho State University, Pocatello, ID 83209, USA
[3]Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, Cairo 11672, Egypt
[4]Thunder Bay Regional Health Research Institute (TBRHRI), Thunder Bay, ON P7B 7A5, Canada
[5]College of Engineering, Alfaisal University, Riyadh 11533, Saudi Arabia
[6]Department of Computer Engineering, Umm Al-Qura University, Makkah 24381, Saudi Arabia

Corresponding author: Mostafa M. Fouda (mfouda@ieee.org)

The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: 19-COM-1-01-0009. In addition, parts of this paper were made possible by National Priorities Research Program (NPRP) from the Qatar National Research Fund (a member of Qatar Foundation) under Grant NPRP13S-0205-200270.

ABSTRACT Due to the proliferation of the Internet of Things (IoT), the IoT devices are becoming utilized at the edge network at a much higher rate. Conventionally, the IoT devices lack the computation resources required for carrying out ultra-edge analytics. In this paper, we go beyond the typical edge analytics paradigm, which is mostly limited to user-smartphones, and investigate how to embed intelligence into the ultra-edge IoT sensors. To conceptualize the smart IoT sensors with enhanced intelligence, we select the arrhythmia detection task employing Electrocardiogram (ECG) trace as one of the mobile health (mHealth) cases. The existing approaches are not feasible for ultra-edge IoT sensors due to the extensive noise-filtering and manual feature extraction phase. Hence, in this paper, to facilitate the analytics, we propose a Deep Learning-based Lightweight Arrhythmia Classification (DL-LAC) method, which employs only single-lead ECG trace and does not require noise-filtering and manual feature extraction steps. As the proposed technique, we design a one-dimensional Convolutional Neural Network (CNN) architecture. Complying with the ANSI/AAMI EC57:1998 standard, four heartbeat types are taken into consideration as class labels. The efficiency and the generalization ability of the proposed model are evaluated, employing four different datasets from PhysioNet. The experimental results demonstrate that the proposed DL method outperforms traditional methods such as the Delay Differential Equation (DDE)-based optimization, K-Nearest Neighbor (KNN), and Random Forest (RF). The proposed DL-LAC illustrates encouraging performance in terms of time and memory requirement when the trained model is transferred to virtualized microcontrollers connected to IoT sensors.

INDEX TERMS Internet of Things (IoT), arrhythmia, electrocardiogram (ECG), deep learning (DL), convolutional neural network (CNN), smart health, smart sensor.

I. INTRODUCTION

The escalation of Artificial Intelligence (AI), Internet of Things (IoT) sensors, and numerous wearable devices have radically enhanced mobile health (mHealth). However, due to the hurdle of incorporating intelligence into these resource-constrained IoT devices, the IoT sensors continue to be routine monitors. The conventional technique is to employ the IoT sensors and wearables to sense user's day to day health data such as Electrocardiogram (ECG), Electroencephalogram (EEG), temperature, respiration patterns, diabetes level, sleep patterns, weight change, and so forth. These health data accumulated by the regular IoT devices are dispatched to a remote cloud for medical analytics,

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

as portrayed in Fig. 1. Although this IoT and cloud-based medical analytics serve the purpose of health monitoring, it still raises a few major concerns that cloud-based architecture cannot avoid easily. This paradigm of ECG data analytics results in bandwidth consumption, delay due to transmitting the enormous amount of health data, and privacy concerns associated with the user's health data.

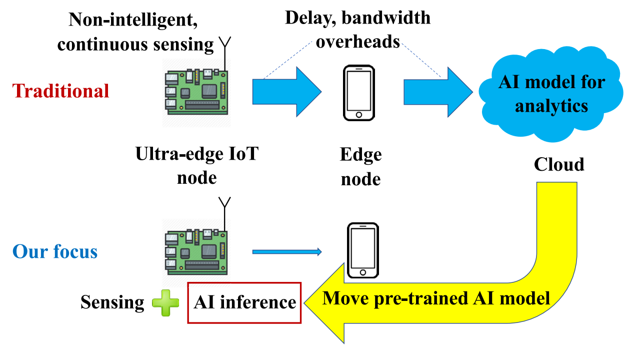


FIGURE 1. How migrate the pre-trained AI model towards the resource-constrained sensor.

Our goal in this paper is to analyze how to exploit the logic-in-sensor concept, recently introduced by the coauthors' earlier research work [1]. The logic-in-sensor architecture, which is based on Magnetic Tunnel Junction (MTJ)-based spintronic technology, can revolutionize the mHealth industry by enhancing the Quality of Service (QoS) such as communication delay, network bandwidth consumption and privacy of user's health data. Considering the user-smartphone as an edge device that is capable of some analytics, the proposed ultra-edge architecture shown in Fig.1 aims to bring the intelligence or the analytics from the cloud to the edge device using the logic-in-sensor concept. Following the hardware enhancement and AI-based intrinsic noise processing, as demonstrated in [1], in this paper, we intend to obtain a lightweight solution to relocate the cloud-based medical analytics to the ultra-edge smart IoT nodes, and hence, overcoming the issues as mentioned earlier.

We have chosen an essential use-case of cardiac arrhythmia, one of the major causes of Cardiovascular Diseases (CVDs) [2]. Cardiovascular diseases are the leading cause of death worldwide, which results in approximately 31% of all global deaths; however, the risk can be eliminated if detected and diagnosed with timely treatment [3]. Arrhythmias cause the heart not to pump blood in the body adequately, and the patients usually experience symptoms of faster or slower heart pulsations. Conventional clinically graded 12-lead ECG or consumer-grade wearables can be employed to monitor the heart activity of a person. The electrical activity of the heart is known as the ECG waveform, which is a crucial diagnostic tool used to monitor the conditions of the heart and can be used to identify arrhythmias [4]. Automatic detection of irregular heartbeats from ECG signals is a significant task for the smart diagnosis of CVDs, and it is becoming a prominent area where AI can be employed extensively to automate the process.

Recent advances in AI and the availability of more health data, the utilization of the deep neural network has proven to be indispensable for automating the smart healthcare system [4]. ECG data analytics using Machine Learning (ML) or AI techniques and analyzing time series ECG with non-linear Delay Differential Equations (DDEs) are explored broadly by traditional cloud-based medical analytics. However, the adaptation of localized embed intelligence at the ultra-edge devices is still not extensively studied in the literature. For diminishing the communication delay and network bandwidth with the cloud and preserve user-data privacy by considering the localized analysis of the health data, a more effective and lightweight analytics technique on-sensor is critical. Therefore, in this paper, we considered several ML techniques to pave the way to move the arrhythmia analytics from the centralized cloud paradigm to ultra-edge smart IoT. Among different AI approaches, we propose a Deep Learning-based Lightweight Arrhythmia Classification (DL-LAC) algorithm employing the one-dimensional Convolutional Neural Network (CNN) that emerges as the most viable solution for ultra-edge ECG analytics.

The proposed CNN-based model is trained at a central node and then can be transferred to the logic-in-sensor simulation for inference. The proposed model can be used to classify heartbeats employing raw single-lead, and it does not require any noise-filtering of the ECG signal, which makes the system lightweight and easy to integrate with the ultra-edge node. In this vein, the proposed deep learning-based CNN employs the recommendation of Association for the Advancement of Medical Instrumentation (AAMI) for the arrhythmia classification task. We have considered four classes of heartbeats, namely N, S, V, and F, in this paper, which represents normal, supraventricular ectopic, ventricular ectopic, and fusion beats, respectively [5]. To evaluate the model's generalization ability, we experimented using four clinically graded ECG datasets and considered different experimental settings to test the model's performance using accuracy, precision, and f-score as performance metrics. Lastly, due to the high fabrication cost of a single logic-in-sensor (approaching $15k for the entire circuit and a further $10k for further customization), we illustrate the viability of the proposed method's feasibility as a lightweight solution in an emulated ECG sensor with a Raspberry Pi and a few other IoT devices.

The remainder of the paper is constructed as follows. Sec. II surveys the relevant research work. The problem of traditional cloud-based analytics and the necessity of lightweight analytics at the smart logic-in-sensor is discussed in Sec. III. The data preparation is outlined in Sec. IV. Our proposed input representation and deep learning model are manifested in Sec. V. The performance of our proposal is assessed in Sec. VI and contrasted with those of K-Nearest-Neighbour (KNN), Support Vector

Machine (SVM) and Random Forest (RF). Finally, Sec VII concludes the paper.

II. RELATED WORK

Due to the availability of IoT devices that can deliver health data, researchers are have been working on ECG classification [6]. As an indispensable strategy for diagnosing heart diseases, ECG monitoring is comprehensively studied and analyzed. It is vital to detect cardiovascular diseases timely, and for that purpose, continuous observation of ECG for a prolonged period is essential. However, the conventional method of long-time ECG monitoring is invasive and expensive, and it hinders the daily activity of the patients. To overcome this issue and introduce some level of automation in the ECG monitoring system, cloud-based ECG analytics can be employed where the ECG signal is usually transmitted using wireless transmission techniques such as Bluetooth, Zigbee, or Wi-Fi [7]–[9]. Therefore, most of these traditional automated ECG monitoring systems analyze the data at the cloud and then send feedback back to the user or care-providers. One of the proposed cloud-based analytics where the ECG data are collected using a wearable monitoring node and are transmitted straight to the IoT cloud using Wi-Fi [10]. An IoT-based patient monitoring system is proposed where data is then processed using a Raspberry Pi, and useful information is delivered to the IoT cloud for cloud-based analytics [11].

In this proposed system [12], AdaBoost and Gradient Boosting algorithm were applied to classify ECG using single-lead ECG. An automatic and fast ECG arrhythmia classifier based on a brain-inspired ML approach known as Echo State Networks (ESN) was implemented in for faster ECG analytics [13]. In another work, an accurate arrhythmia classification method for ECG was proposed based on extreme weighted gradient boosting (XGBoost) using a broad range of feature set [14]. In [15], to tackle the patients' privacy concerns, Baza *et al.* have proposed a mimic learning-based machine learning approach for automatic, secure, and efficient analysis of Cardiovascular activities. A clustering-based feature extraction algorithm followed by employing a number of well-known ML classifiers for accurate recognition and classification of arrhythmias is proposed in [16]. Researchers have also employed mathematical methods to decompose ECG, such as a nonlinear DDE was utilized to classify ECG by differentiating features for various heart diseases [17].

Apart from traditional ML techniques, researchers have also employed neural networks and deep learning-based approaches for the classification of ECG heartbeats. In one of the research works, the convolutional neural network of 34-layer was adopted to classify with high accuracy that transcends the cardiologist performance [18]. Principal Component Analysis (PCA) based feature extraction followed by a Multi-Layer Perceptron (MLP) was utilized in another research [19]. Deep-learning-based, Long Short-Term Memory (LSTM) algorithm was proposed in [20], having considerable low computational costs. Recurrent Neural Networks (RNN) was used for binary classification (normal and abnormal) of heartbeat in this research [21]. A Deep Genetic Ensemble of Classifiers (DGEC) was proposed by combining deep learning algorithms with an ensemble learning and genetic optimization of parameters for the classification of various types of arrhythmias [22]. In our recent work [23], these issues were raised and an attempt was made to embed AI at the IoT sensor level to perform ECG prediction at the ultra-edge network. However, the work concluded the need for a systematic investigation and computational analysis to conceptualize a fusion of logic and sensing to render a continuous and lightweight arrhythmia monitoring system.

III. PROBLEM FORMULATION

As manifested in the previous section, the healthcare sector still needs accelerating improvement in establishing smart healthcare with embedded intelligent sensors. As our research focus in this paper is lightweight arrhythmia monitoring, we will discuss the drawbacks of the existing ECG/arrhythmia monitoring system and the hurdles associated with transferring the existing analytics to ultra-edge IoT. Traditionally, researchers have employed diverse heartbeat classification techniques that generally require a number of pre-processing steps such as noise filtering, manual feature extraction, and so forth. The steps needed by the conventional heartbeat classification employing ML methods are exhibited in Fig. 2. Diverse methods such as DWT, DDEs [24], and ML techniques are commonly utilized in the conventional feature extraction and classification tasks. Though these ECG analytics techniques overcome many drawbacks of the manual ECG monitoring, it still lacks the potential to be integrated with logic-in-sensors due to the extensive computational steps. These conventional ECG monitoring approaches mostly rely on multi-lead ECG signal and requires multiple preparatory steps (i.e., noise filtering), which is a significant issue for combining these models with the ultra-edge IoT logic-in-sensors [23].

Apart from ML techniques, traditionally DDE-based optimization techniques have also been proposed for the ECG monitoring task. However, the non-linear DDE for the time-series ECG analysis technique cannot adequately infer the system models in varying heart conditions. In this approach, exhaustive search or heuristics must be developed to select the most competent model for any given classification task, which is a considerable challenge for lightweight ECG analytics. Conventionally a non-linear DDE can be expressed as follows:

$$\begin{aligned} f(a_i, x_{\tau_j}) = {} & a_1 x_{\tau_1} + a_2 x_{\tau_2} + a_3 x_{\tau_3} + \ldots + a_{i-1} x_{\tau_n} \\ & + a_i x_{\tau_1} x_{\tau_1} + a_{i+1} x_{\tau_1} x_{\tau_2} + a_{i+2} x_{\tau_1} x_{\tau_3} \ldots \\ & + a_{j-1} x_{\tau_n}^2 + a_j x_{\tau_1}^3 + a_{j+1} x_{\tau_1} 2 x_{\tau_2} + \ldots \\ & \vdots \\ & \ldots + a_1 x_{\tau_n}^m, \end{aligned} \tag{1}$$

Here, x_{τ_j} can be expressed as: $x_{\tau_j} = x(t - \tau_j)$, in Eq. 1, n, t, m, and τ_j represents the number of delays, time, the degree of non-linearity, and time delays, respectively. The selection of optimal time-delays and monomials is imperative for building an effective DDE-based classification system. For example, to select the optimal model for classification using the DDE-based model, the authors applied the genetic algorithm in [25]. Therefore, these approaches are not appropriate for integrating with the logic-in-sensors for ultra-edge IoT analytics in polynomial time.

Apart from expensive computational requirements, some of the other issues with the traditional ECG monitoring system are that it requires internet connectivity to communicate with the cloud servers for ECG analytics. Hence, it consumes considerable network bandwidth if the number of users is high. Furthermore, due to continuous data transmission, cloud-based analytics can also raise significant privacy concerns for the user's private data. Therefore, this approach can be a hindrance to secure ECG analytics for arrhythmia detection. To address this challenge, we focus on developing an automated, efficient, and lightweight system with localized intelligence that can be deployed and integrated with the logic-in-sensors for ultra-edge IoT analytics. To develop a lightweight ECG/arrhythmia monitoring system, we envision an AI-aided technique for classifying heartbeats employing a raw single-lead ECG signal and compared the proposed model with traditional ML techniques adopting the architecture depicted in Fig 2.

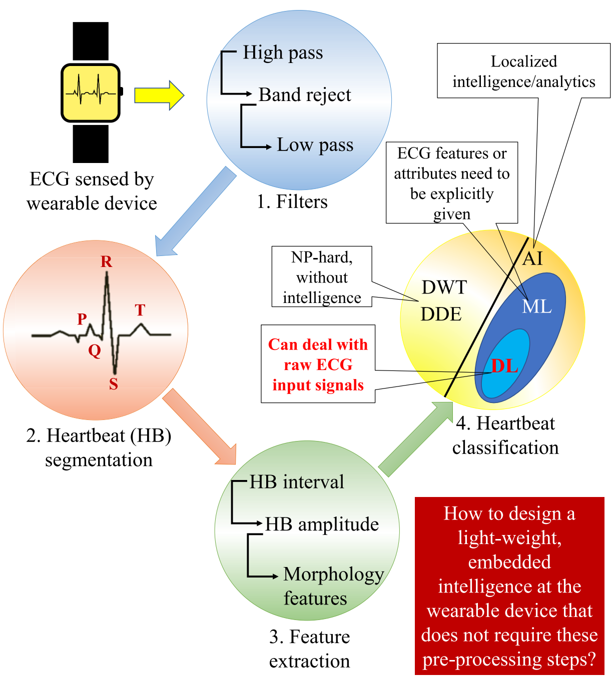


FIGURE 2. Steps of conventional ECG heartbeat classification.

IV. DATA PREPARATION

We have conducted ECG signal analysis to detect arrhythmia by utilizing the MIT-BIH Supraventricular Arrhythmia Database (DS1) [26], MIT-BIH Arrhythmia Database (DS2) [27], St Petersburg INCART 12-lead Arrhythmia Database (DS3), and Sudden Cardiac Death Holter Database (DS4) [28] from PhysioNet [29]. The datasets contain recordings of many traditional and life-threatening arrhythmias along with cases of normal heartbeat rhythm. Various researchers have employed these datasets for diverse ECG based research [30], [31].

The datasets comprise a text header file, a binary file, and a binary annotation file with.txt,.dat, and.atr extensions, respectively.

1) Header file (.hea): This file contains a brief text file that explains the signals' contents, such as the name of the record's file, number of examples, type and format of the ECG signal, and so forth.
2) Binary file (.dat): The binary files include digitized representations of the ECG signals of each record.
3) Annotation files (.atr): The annotation files contain heartbeat labels that define the type of ECG signals at a particular time in the ECG record.

We generated four separate heartbeat categories following the Association for the Advancement of Medical Instrumentation (AAMI) EC57 standard from the annotation files in each of the datasets. The summary of mappings between the heartbeat annotations for each class is demonstrated in Table 1. We have employed the DS1 (MIT-BIH Supraventricular Arrhythmia Database) for the hyper-parameter tuning and the training phase. In the running/inference stage, we test the model using the other three datasets (i.e., DS2, DS3, and DS4). We exploited multiple datasets to evaluate the generalization ability of the proposed model. Although each of the datasets contains multiple ECG lead's data, we have employed the lead II in our experiment as our model only requires single-lead-ECG tracing. The distribution of four heartbeat labels is manifested in the Table 2.

V. PROPOSED METHODOLOGY

A. PROPOSED CNN MODEL STRUCTURE

In this section, we illustrate the proposed lightweight heartbeat classification technique for arrhythmia detection that can be deployed and integrated with AI-aided logic-in-sensor. A lightweight model for classification is an essential part of integrating the AI-aided model at the ultra-edge IoT sensors for faster analysis. Hence, we primarily focused on designing the deep learning-based model that only requires a single lead raw ECG signal so that the model can be sufficiently lightweight. Sensors with embedded intelligence can be utilized for long-term, accurate monitoring of a person's cardiac activity, which is demonstrated in one of the coauthors' previous works [1]. Keeping the concept of logic-in-sensor in focus, we developed a deep-learning-based lightweight

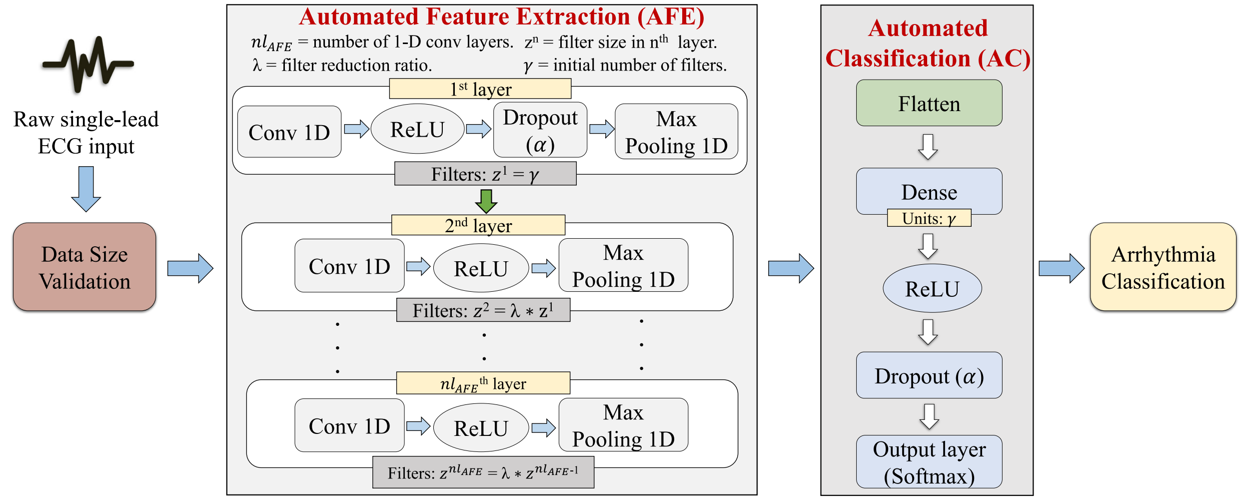


FIGURE 3. Proposed training architecture leveraging CNN structure for the considered use-case. Once the model is trained at the cloud, it is transferred to the smart IoT sensor's AI module.

TABLE 1. Mapping DS1, DS2, DS3, and DS4 datasets to the AAMI heartbeat classes [32].

Heartbeat Class	Heartbeat Annotation
N (Normal)	N (Normal)
	L (Left bundle branch block beat)
	R (Right bundle branch block beat)
	e (Atrial escape beat)
	j (Nodal (junctional) escape beat)
S (Supraventricular ectopic beat)	A (Atrial premature beat)
	a (Aberrated atrial premature beat)
	J (Nodal (junctional) premature beat)
	S (Supraventricular premature beat)
	V (Premature ventricular contraction)
V (Ventricular ectopic beat)	E (Ventricular escape beat)
F (Fusion beat)	F (Fusion of ventricular & normal beats)

TABLE 2. Frequency of heartbeats of each class in DS1, DS2, DS3, and DS4.

Heartbeat Class	DS1	DS2	DS3	DS4
N	1,62,323	90,621	1,53,672	7,45,671
S	12,197	2,781	1,960	1,893
V	9,941	7,236	20,012	23,616
F	23	803	219	309

model that can be integrated with these AI-aided sensors for analysis of ECG at the ultra-edge device. The acquired results of the ECG analytics can then be sent from the IoT nodes to the care-providers.

We propose an automated deep learning-based one dimensional (1-D) CNN that does not necessitate any noise-filtering and manual feature extraction. The CNN model detects unique patterns automatically from the raw single-lead ECG signal. The ECG signals are sampled at a frequency of f_s before passing to CNN as input. The lightweight ECG analysis for arrhythmia detection task takes an ECG signal as input $X = [x_1, x_2, x_3, \ldots x_n]$, and outputs a sequence of labels $Y = [y_1, y_2, y_3, \ldots y_n]$. Here each y_i represents one of four different heartbeat classes and in terms of arrhythmia classification $y_i \in \{F, N, V, S\}$. Table 1 exhibit of the summary of each of the classes. We consider a minimum length of ECG signal noted as δ to be passed as input to the model. Every output label corresponds to a portion of the input ECG signal, and collectively the output labels cover the full sequence of the ECG signal record of a subject.

As the deep learning-based solution, a 1-D CNN is designed and used because of its exceptional performance in automatically detecting patterns in the ECG signal. The proposed CNN model can be defined briefly as the combination of the convolution layers, max-pooling layer, and fully-connected layers. Fig. 3 represents the architecture of the proposed CNN model. Here the model receives raw ECG signal as input and generates heartbeat classes as output. CNN consists of two segments; the first segment comprises nl_{AFE} number of 1-D convolution layers performing Automated Feature Extraction (AFE) from the raw single-lead ECG signal and an Automated Classification (AC) module that process the extracted features using nl_{AC} number of fully connected layer followed by the output layer for classification. The 1-D convolution operation can be expressed as in Eq. 2.

$$x_k^l = \sum_{i \in nl_{AFE}} (x_i^{l-1} * w_i^l + b_k^l) \quad (2)$$

Here, x_k^l and b_k^l can be defined as the input and bias for the k^{th} node of l^{th} layer, respectively. The kernel is defined as w_i^l and the input of the i^{th} node of the $(l-1)^{th}$ layer is denoted as x_i^{l-1}. To select the optimal activation function for the proposed model, we performed hyper-parameter tuning. The Rectified Linear Unit (ReLU) [33] is selected as activation function, Ω, defined in Eq. 3.

$$\Omega(z) = max(0, z) \tag{3}$$

In the first convolution layer, we also apply dropout with a rate of α as the regularization technique, which will serve the network in avoiding overfitting. Hence, the model can gain enhanced generalization ability by randomly disregarding some selected neurons in the hidden layers. After the regularization layer, we employ the subsampling technique to compress the size of the ECG data and reduce computation time. We have employed the max-pooling layer to obtain the maximum value in a particular region. Eq. 4 determines the output of the j_{th} unit of the subsampling layer l. where x_j^l represents the output of the j_{th} unit of layer l and $x_{j_{output}}^{l-1}$ represents the j_{th} output group of layer l-1. The kernel size of the max-pooling layer is set to a constant km_{init} for each of the layers.

$$x_j^l = subsample(x_{j_{output}}^{l-1}) \tag{4}$$

The n^{th} layer of the AFE module produces a feature matrix from the ECG data. The extracted features by the initial module are relinquished to the subsequent stage for further analysis. In the next stage, the AC module consists of a single flatten layer, followed by a fully connected layer and an output layer. The flatten layer is responsible for transforming the features into a vector that can be forwarded into a fully connected [34]. ReLU and softmax activation functions are selected to be used in the fully-connected layer and output layer, respectively.

B. DEEP LEARNING-BASED LIGHTWEIGHT ARRHYTHMIA CLASSIFICATION (DL-LAC) ALGORITHM

In this subsection, we present the steps of the training and inference phases of our proposed DL-LAC algorithm.

The training phase of the proposed DL-LAC algorithm includes Algorithms 1, 2, and 3. The training stage of DL-LAC commences from Algorithm 1 with the inputs D, k, ξ, B, Ω, and δ. The details of each of the inputs are provided in the algorithm's input section. The training phase of the algorithm returns the trained model (M_t), which is further harnessed in the inference phase. The algorithm initiates with initializing the required parameters in the steps 1 to 3. Then, in step 4, the ECG signal and the corresponding heartbeat class labels are loaded from the dataset, which is later utilized in training. After that, in step 5, the ECG data is validated by checking with a pre-defined size threshold in the DSV algorithm described in Algorithm 2. Afterward, in the steps 6-11, the training ECG data and the heartbeat labels are employed to train the model (M_t) using k-fold stratified cross-validation. At the penultimate step, the trained model (M_t) is stored for further testing and validation. Finally, in step 13, the algorithm concludes by returning the trained model.

For the data size validation purpose, our proposed DSV algorithm is demonstrated in Algorithm 2. This algorithm's main objective is to validate the length or size of the

Algorithm 1 Training Phase of DL-LAC

Input: D (ECG data collection for training), k (number of fold in cross-validation), ξ (number of epoch), B (mini-batch size), Ω (activation function), δ (threshold for data size)

Output: M_t (Trained model)

1 $M_t \leftarrow \emptyset$
2 $X_\delta \leftarrow []$
3 $y_\delta \leftarrow []$
4 $X, y \leftarrow$ read ECG signal and annotated heartbeats from D
5 X_δ $y_\delta \leftarrow$ call DSV(X, y) from algo. 2
6 **for** *(fold no.* $j = 1$ *to* k*)* **do**
7 $\quad X_{train}, y_{train}, X_{val}, y_{val} \leftarrow$ set data and labels of j^{th} fold from X_δ, y_δ
8 $\quad F_{train} \leftarrow$ call AFE(X_{train}, Ω) from algo. 3
9 $\quad$ update the model parameters of M_t by passing F_{train} through the AC module as depicted in Fig 3
10 $\quad$ compute validation performance using X_{val}, y_{val}
11 **end for**
12 save the model parameters of M_t
13 return M_t

Algorithm 2 Data Size Validation (DSV)

Input: X (data), y (heartbeat labels), δ (threshold for data size)

Output: X_δ (updated data after size validation), y_δ (updated heartbeat labels after size validation)

$X_\delta \leftarrow []$
$y_\delta \leftarrow []$
for *(*$i = 1$ *to* $length(X)$*)* **do**
$\quad$ **if** *(*$length(X_i) < \delta$*)* **then**
$\quad\quad$ continue
$\quad$ **else**
$\quad\quad X_\delta \leftarrow$ add $X_i[1 : \delta]$
$\quad\quad y_\delta \leftarrow$ add y_i
$\quad$ **end if**
end for
return X_δ, y_δ

ECG signal by checking with a pre-defined threshold of δ. The algorithm takes X, y, and δ as input and produces an updated version of X and y, denoted as X_δ and y_δ, respectively. We utilize this algorithm in both the training and inference phase before the start of their workflow.

In the Algorithm 3, the required steps for the AFE module of the proposed model is manifested. We utilize this algorithm from step 8 of the Algorithm 1, to extract the unique features from the ECG signal. The extracted feature matrix from this algorithm is then employed in the later module for classification. The inputs to the algorithm are X and Ω, whereas the extracted unique features (F_{train}) are returned as

Algorithm 3 Automated Feature Extraction (AFE)

Input: X_t (training data), Ω (activation function)
Output: F_x (extracted features)
1 initialize γ (initial filter size), λ (filter size reduction factor), nl_{AFE} (number of conv. layers), α (dropout rate)
2 $z^1 \leftarrow \gamma$
3 **foreach** *layer* $i \in nl_{AFE}$ **do**
4 $\quad F_x \leftarrow$ pass X_t through the convolution layer with z^i and Ω
5 $\quad$ **if** $(i = 1)$ **then**
6 $\quad\quad F_x \leftarrow$ apply regularization of rate α (dropout)
7 $\quad$ **end if**
8 $\quad F_x \leftarrow$ update F_x by passing through sub-sampling layers (max-pooling)
9 $\quad z^{i+1} \leftarrow z^i * \lambda$
10 **end foreach**
11 return F_x

Algorithm 4 Inference Phase of DL-LAC

Input: $path_{test}$ (test data location)
Output: y_{pred} (predictions by the model)
1 $X_{test}, y_{test} \leftarrow$ load all test ECG data and corresponding class labels from $path_{test}$
2 $M_t \leftarrow$ load the pre-trained model
3 $X_{test}, y_{test} \leftarrow$ call DSV(X_{test}, y_{test}) from algo. 2
4 $y_{prob} \leftarrow$ predict the probabilities for each sample of X_{test} employing the model M_t
5 $y_{pred} \leftarrow$ argmax(y_{prob})
6 return y_{pred}

the output of the algorithm. Step 1 and 2 initialize the required parameters. In steps 3 to 10, the automated feature extraction module's main workflow is illustrated for nl_{AFE} number of convolution layers. In step 4, the input is passed through the 1-D convolution, the results of which will then be forwarded to the later layers. We employed dropout with a rate of α for the first convolution layer ($i = 1$), which is expressed in steps 5 to 7. Step 8 performs the sub-sampling operation using the max-pooling technique described in the previous subsection (Eq. 4). After that, we update the number of filters to be used by the reduction factor λ, in the next convolution layer in step 9. Finally, in step 11, the extracted feature matrix is returned for the next module to use.

In the inference phase, the proposed DL-LAC algorithm is exhibited in the Algorithm 4. It receives the location of test ECG data for inference and returns the predicted class labels (y_{pred}) for the corresponding sample. After loading the testing ECG data from step 1, the pre-trained model (M_t) is loaded in the subsequent step. Step 3 is responsible for updating the test data by validating the data length from Algorithm 2. In step 4, the model M_t is used to predict the probabilities for a sample ECG test data to belong in each of the four classes. In step 5, the class with the highest probability is selected as the classified class for each sample data. Ultimately, in the last step, the collection of predictions for all the data is returned.

C. COMPUTATIONAL COMPLEXITY ANALYSIS IN TERMS OF MATHEMATICAL OPERATION

This section investigates the algorithm's complexity and the time cost to run the proposed deep learning-based lightweight ECG monitoring system to detect arrhythmia. We analyze the complexity of the proposed model's training and inference steps in terms of the number of different operations required by various stages of the model. The analysis primarily encompasses the mathematical analysis of the algorithm complexity in the training phase and inference phase through determining the recurrence of each operation (e.g., addition, subtraction, multiplication, and division, etc.). We express the addition and multiplication operations as ADD and MUL, respectively. In addition, we also analyze the occurrence of comparisons denoted as COMP.

1) TRAINING PHASE

The training phase comprises the DSV algorithm for data augmentation and the DL-LAC training phase for the proposed CNN model. In the training phase of DL-LAC, depicted in Algorithm 1, we perform computational overhead analysis, considering that the appropriate hyper-parameters of the proposed models are already selected after hyper-parameter tuning employing the grid search technique. We divide the overall analysis of the training phase, mainly into three different fragments, such as the required data size validation phase, feature extraction phase, and the classification phase. Therefore, the total computational complexity can be expressed as Eq. 5:

$$C(Training) = C(DSV) + C(AFE) + C(AC). \tag{5}$$

Here, $C(DSV)$, $C(AFE)$, and $C(AC)$ indicate the required computational overhead in the data size validation, automated feature extraction, and automated classification phases, respectively. For each of these three phases, the computation complexity is divided into three parts: the required number of additions, multiplications, and comparisons. In the first stage, to calculate the complexity of the data size validation phase, we mainly analyze the complexity of the Algorithm 2, which is invoked from the training procedure (Algorithm 1). The first four steps of the training algorithm are initializing steps; hence these do not require any mathematical operations (i.e., addition and multiplication). In step 5, the Algorithm 2 is invoked for validating the ECG data size. If the length of considered training ECG trace is $len(X_{train})$, then the required number of comparisons is also $len(X_{train})$ as the condition will be validated for each ECG trace.

In the next phase, the computational overhead is determined for the feature extraction phase manifested in the Algorithm 3 of the training procedure. For a particular layer (l^{th} layer) of the AFE module, if we consider that there are

N^l number of nodes for the convolution layer, then the number of required operations can be defined as Eqs. 6 and 7.

$$\begin{aligned} C(AFE_{ADD}) = nl_{AFE} * \xi * N^l * (len(x^l)/B) \\ *((len(k^l) * len(x^{l-1})) - (len(k^l) - \eta) + 1) \\ -(len(x^{l-1}) - (len(k^l) - \eta)) * z^l)), \end{aligned} \tag{6}$$

$$\begin{aligned} C(AFE_{MUL}) = nl_{AFE} * \xi * N^l * (len(x^l)/B) \\ *(z^l * ((len(k^l) * len(x^{l-1})) \\ -(len(k^l) - \eta) + 1)) + (nl_{AFE} - 1). \end{aligned} \tag{7}$$

Here, x^l, k^l, and z^l indicate the input, kernel, and the number of filters of layer l. The striding window length, the number of epoch, and batch sizes are denoted by η, ξ, and B, respectively.

Also, in terms of the AFE phase, the number of comparisons required for nl_{AFE} layers can be denoted as eq. 8. Here, x^l and z^l implies the input and the number of filters in the l^{th} layer of the AFE phase. For z^l number of filters, the number of comparisons required at the layer l due to passing the input x^l through the activation function (Ω) is $((z^l * len(x^l))$. In the sub-sampling layer (i.e., max-pooling layer), the number of comparisons required is $(len(x^l) - (z^l - 1))$.

$$\begin{aligned} (AFE_{COMP}) = \sum_{l=1}^{nl_{AFE}} (\xi * N^l * (len(x^l)/B) * ((z^l * len(x^l)) \\ +(len(x^l) - (z^l - 1)))) \end{aligned} \tag{8}$$

The extracted features set (F_x) of the AFE phase will be relinquished to the AC module of the proposed model for the classification task. For a particular layer denoted as l, if the output of the i^{th} layer is γ^i, then the computational complexity for the i^{th} layer can be $(len(\gamma^i) * (len(F_x) - 1))$ ADD, $(len(\gamma^i) * len(F_x))$ MUL. Thus considering the number of fully-connected layers to be nl_{AC}, the computational complexity of this phase can be denoted as Eqs. 9 and 10.

$$C(AC_{ADD}) = \sum_{i=1}^{nl_{AC}} (\xi + (len(\gamma^i)) * (len(F_x) - 1), \tag{9}$$

$$C(AC_{MUL}) = \sum_{i=1}^{nl_{AC}} (len(\gamma^i)) * len(F_x) * \xi. \tag{10}$$

In terms of the number of comparisons required in the AC phase, considering nl_{AC} layers, the cumulative comparisons due to the comparisons as are necessary for computing the activation functions can be denoted as Eq. 11.

$$C(AC_{COMP}) = \sum_{i=1}^{nl_{AC}} (\xi * len(\gamma^i)). \tag{11}$$

Hence, by substituting the equations, as mentioned earlier in the Eq. 5, the overall computational complexity in terms of the number of mathematical operations required in the training phase of the proposed DL-LAC algorithm can be expressed as Eq. 12. The number of comparisons needed in different stages of the DL-LAC algorithm's training phase is also considered in this equation.

$$C(Training) = \begin{cases} ADD: & C(AFE_{ADD}) + C(AC_{ADD}) \\ MUL: & C(AFE_{MUL}) + C(AC_{MUL}) \\ COMP: & len(X_{train}) + C(AFE_{COMP}) \\ & + C(AC_{COMP}). \end{cases} \tag{12}$$

2) INFERENCE PHASE

The inference/running phase is conducted to infer classes of each testing ECG data employing the pre-trained lightweight model (M_t) and then evaluating it using the unseen data. In correspondence with Algorithm 4, if we consider the test data to be X_{test}, and the size of test data after validating ECG signal is $len(X_{test})$, then the computational complexity of the inference phase can be denoted as follows:

$$C(Inference) = \begin{cases} ADD: & \sum_{i=1}^{nl_{AFE}} (len(x^i) - (\eta + 1)) \\ & + \sum_{j=1}^{nl_{AC}} (len(\gamma^j) - 1) \\ MUL: & \sum_{i=1}^{nl_{AFE}} (len(x^i) - \eta) \\ & + \sum_{j=1}^{nl_{AC}} (len(\gamma^j)) \\ COMP: & \sum_{i=1}^{nl_{AFE}} (len(x^i)) \\ & + \sum_{j=1}^{nl_{AC}} (len(\gamma^j)) \\ & + len(X_{test}) \end{cases} \tag{13}$$

Eq. 13 illustrates that, in the inference phase, the pre-trained model is able to produce results with considerably lower computational operations and in linear time (i.e., upper bound time-complexity of $O(len(X_{test})$, in Big O notation). The complexity analysis indicates that it can be utilized for lightweight arrhythmia classification at the resource-constrained ultra-edge IoT node.

VI. PERFORMANCE EVALUATION

In this section, we manifest the simulation results to establish the algorithmic analysis of the proposed lightweight DL-LAC method that is estimated in the previous section. Hence, the proposed method is compared with the traditional techniques in terms of classification performance, memory consumption, and required time for inference.

A. PERFORMANCE INDICATORS

To evaluate the classification results, we adopted the combination of three measurement indicators, accuracy, weighted precision, and weighted F1 score. The accuracy of a test is its ability to correctly differentiate the three cases. Considering, C = Number of classes in the considered classification task, $len(y_i)$ = number of samples in the i^{th} class, TP_i = the number of cases correctly identified to be in the i^{th} class, and $len(Y)$ = total number of samples in all the class, the accuracy can be denoted as Eq. 14:

$$\text{Accuracy} = \frac{\sum_{i=1}^{C}(TP_i)}{len(Y)}. \tag{14}$$

The weighted precision can be expressed as Eq. 15. It addresses how precise the model is out of those predicted to be in i^{th} class, how many of them are actually in i^{th} class, and the value is multiplied by the weight of the i^{th} class as follows:

$$\text{Weighted precision} = \sum_{i=1}^{C}\left(\frac{len(y_i)}{len(Y)} * \frac{TP_i}{TP_i + FP_i}\right). \quad (15)$$

Here, FP_i represents the number of cases incorrectly identified to be in the i^{th} class. Weighted F1 score is the weighted average of precision and recall. Hence, although we did not use recall directly as a performance measure, because of using the F1 score, it is implicitly used. The weighted F1 score can be obtained as follows:

$$\text{Weighted F1 score} = \sum_{i=1}^{C}\left(\frac{len(y_i)}{len(Y)} * 2\frac{P_i * R_i}{P_i + R_i}\right). \quad (16)$$

In the above equation, the precision and recall of i^{th} class are indicated by P_i and R_i, respectively. P_i can be expressed as $TP_i/(TP_i + FP_i)$ and R_i can be denoted as $TP_i/(TP_i + FN_i)$. FN_i denotes the number of cases incorrectly identified as a class other than the i^{th} class.

B. RESULTS AND DISCUSSION

We have conducted comprehensive experiments in a systematic approach to identify the optimal model. Here, the experimental results can be summarized as follows:

1) The first phase of the experiment encompasses the hyper-parameter tuning to find the optimal structure of the model. The selected hyper-parameters were applied in the proposed DL-based model.
2) In the second phase, we measured the model's performance employing the trained model obtained from DS1 and then tested it using MIT-BIH Arrhythmia Database (DS2), St Petersburg INCART 12-lead Arrhythmia Database (DS3), and Sudden Cardiac Death Holter Database (DS4).
3) In the third phase of the experiment, we evaluated the proposed model's generalization ability by utilizing each of the four datasets individually for training and testing purposes using k-fold cross-validation.
4) Finally, numerical analysis is carried out to assess the proposed CNN models' effectiveness in terms of execution time required and memory consumption in various IoT devices and compared to the traditional ML techniques.

We performed hyper-parameter tuning to select the optimal parameters for the proposed CNN-based model in the initial phase of the experiment. Figure 4 demonstrates the results of manual tuning for the number of convolution layers used in the model by varying the number from one to five. The experimental results illustrate that, for three convolution layers, the best performance is achieved with 96.26%, 0.9606, and 0.9604 accuracy, precision, and F1-score, respectively.

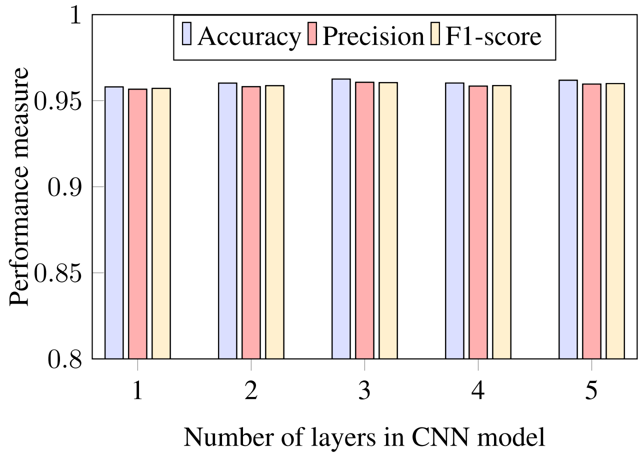


FIGURE 4. **Performance variation of the proposed/custom CNN model with varying numbers of layers.**

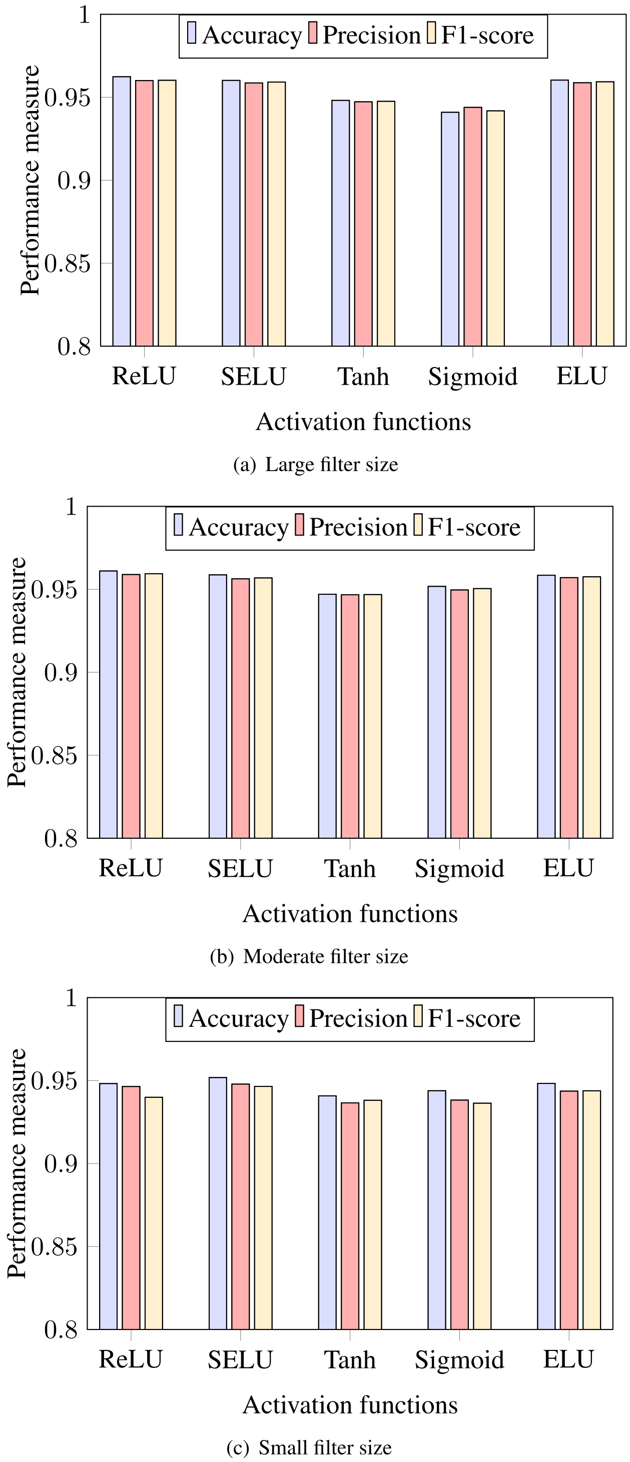
Therefore, we conducted further analysis using three number of convolution layers in the proposed DL-based architecture.

Furthermore, to select the optimal activation function (Ω) and the number of the initial filter size (γ), we performed a grid search technique. Figure 5 demonstrates the results obtained from the grid search where 5(a), 5(b), 5(c) represents the initial number of filter equals to large, moderate, and small, respectively. For the grid search, we considered three sizes for the filters of the first convolution layer, such as large, moderate, and small, with the value of 300, 150, and 50, respectively. For selecting activation function (Ω), we experimented with a set of five activation functions: relu, selu, elu, tanh, and sigmoid. According to performance, the best combination is evident when the activation is ReLU, and the number of filters is large with the accuracy, precision, and F1-score, respectively 96.23%, 0.96004, and 0.9601.

Additionally, to elect the optimal optimizer, batch size, dropout, and epochs, we performed a grid search, which is manifested in Table 3. We have conducted the grid search among six widely used optimizers such as Adadelta, Nadam (Nesterov-accelerated Adaptive Moment Estimation), SGD (Stochastic Gradient Descent), RMSprop (Root Mean Square Propagation), Adagrad (Adaptive Gradient Algorithm), and Adam (Adaptive Moment Estimation). For the batch size, we tuned the value employing a set of three different

TABLE 3. **Selected parameters for each optimizer after employing grid search.**

Optimizer	**Selected paramters**			**Accuracy**
	Batch size	Dropout rate	Epochs	
Adadelta	5000	0.2	100	88.55%
Nadam	2500	0.2	10	88.67%
SGD	5000	0.4	100	88.81%
RMSprop	1000	0.4	10	89.19%
Adagrad	1000	0.5	10	89.56%
Adam	5000	0.5	10	**91.85%**



(a) Large filter size

(b) Moderate filter size

(c) Small filter size

FIGURE 5. Performance comparison for different activation functions with respect to different filter size of the proposed CNN.

values, such as 1000, 2500, and 5000. For selecting the optimal dropout rate (α), we considered values from 0.1 to 0.5. We varied the number of epochs (ξ) using three values (i.e., 10, 50, and 100). The best performing combination is obtained for the Adam optimizer along with batch size 5000, dropout rate 0.5, and the number of epochs 10. Hence, for further analysis of the experiment, we employed these parameter values for the model.

In the second phase of the experiment, we utilized DS1 as the training dataset and then tested the model's performance using DS2, DS3, and DS4 as test dataset. Table 4 illustrates the results for this phase. In all three test datasets, the proposed model outperformed the traditional ML methods (i.e., random forest, KNN) in terms of accuracy, precision, and F1-score. The proposed CNN achieved an accuracy of 94.07%, 92.04%, and 95.83%, while DS2, DS3, and DS4 are harnessed as the test dataset, respectively. The proposed custom CNN model is showing superior performance over traditional ML techniques because the combination of Convolution, sum-sampling, and regularization layers are able to capture the detailed features from the ECG signal automatically. Furthermore, due to the adaptive filter reduction in the deep convolution layers, the proposed model can identify significant points from the ECG with higher efficiency, and because of the use of the regularization layer, the proposed approach is able to avoid overfitting during training. However, the traditional methods are lacking the ability to automatically retrieve significant features from the ECG even after extensive manual pre-processing stages. Although the proposed model outperforms the traditional methods in terms of performance, the notable part is that the proposed technique can achieve great accuracy even with raw ECG signals, without adopting noise-filtering and manual feature extraction of the ECG. The results reveal that the custom CNN-based model is robust in detecting heartbeats with high accuracy and lightweight because of using raw single-lead ECG.

In the penultimate experimental phase (third phase), we experimented using each dataset individually, as manifested in Fig. 6, employing 3-fold stratified cross-validation to validate the generalization capability of the proposed model. Stratification is a method in which the samples are rearranged to have a stable representation of the whole dataset

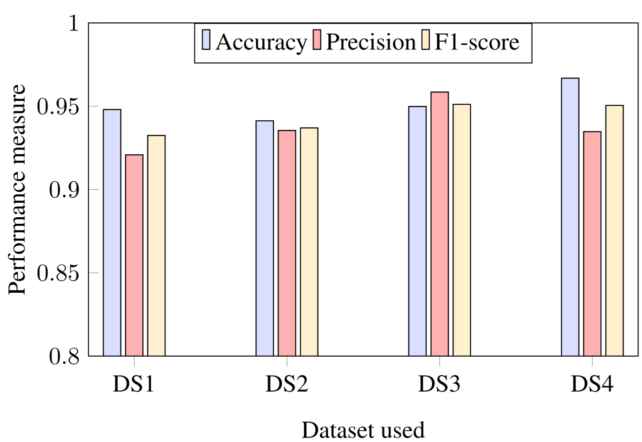
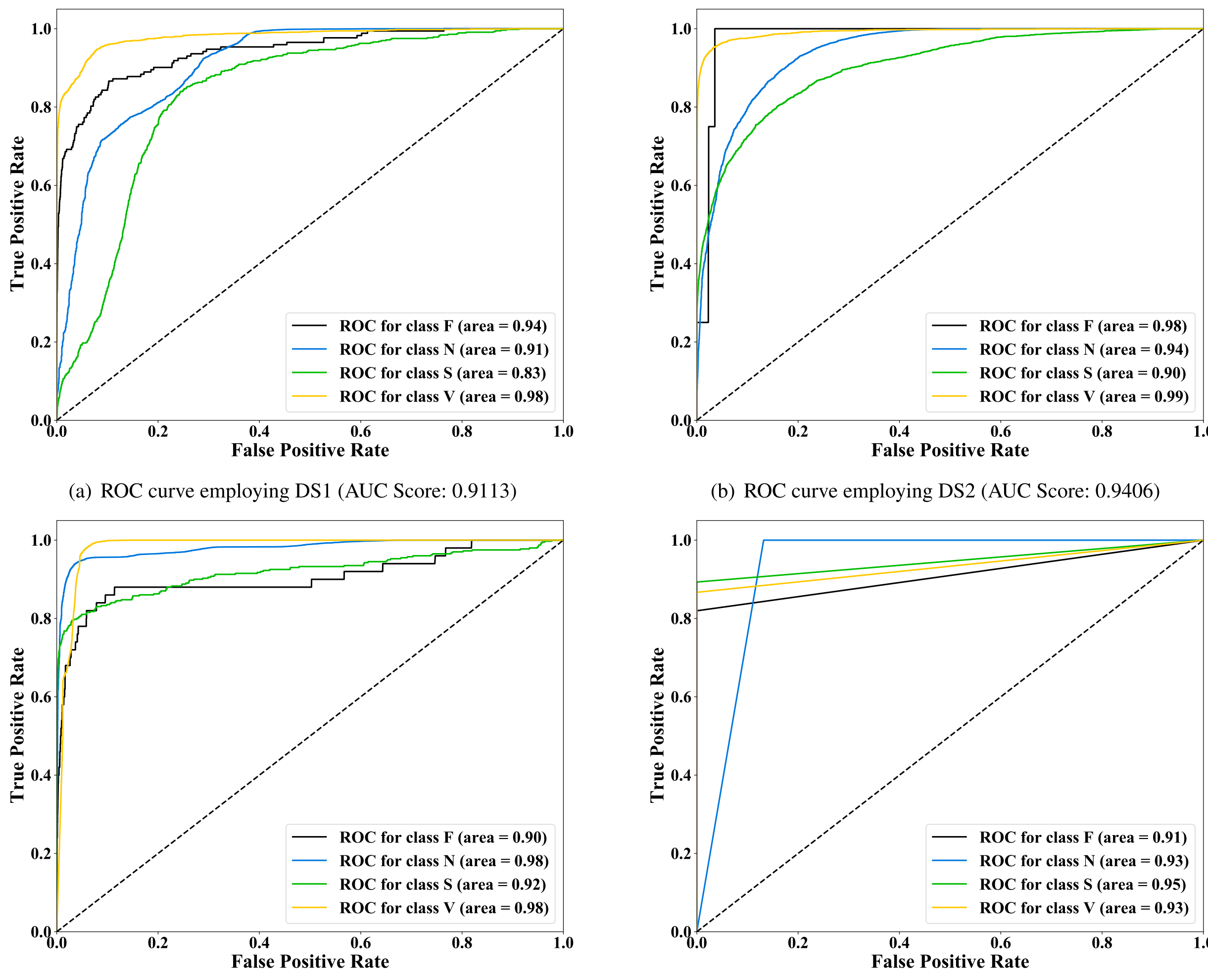


FIGURE 6. Performance of the proposed model for the third experimental setting employing the four datasets individually (3-fold stratified cross-validation). Here, DSi means the i^{th} dataset.

TABLE 4. Performance comparison of CNN with traditional ML methods for the second experimental setting using DS1 as the training dataset.

Method	Test Dataset	Accuracy	Precision	F1-Score	Pre-processing	Feature extraction	ECG type
KNN	DS2	89.83%	0.8541	0.8646	3-phased noise filtering	Required	Single-lead
	DS3	89.76%	0.9281	0.9124			
	DS4	56.53%	0.7383	0.6991			
RF	DS2	89.31%	0.9010	0.8957	3-phased noise filtering	Required	Single-lead
	DS3	90.21%	0.8921	0.8844			
	DS4	85.77%	0.9126	0.8947			
CNN	DS2	**94.07%**	0.907	0.9175	**Not required**	**Not required**	Single-lead
	DS3	**92.04%**	0.8991	0.9017			
	DS4	**95.83%**	0.9562	0.9572			



(a) ROC curve employing DS1 (AUC Score: 0.9113)

(b) ROC curve employing DS2 (AUC Score: 0.9406)

(c) ROC curve employing DS3 (AUC Score: 0.9796)

(d) ROC curve employing DS4 (AUC Score: 0.9340)

FIGURE 7. Area Under the Receiver Operating Characteristic (AUROC) curve derived for the third experimental settings utilizing 3-fold stratified cross-validation.

by preserving the portion of samples for each class [35]. The cross-validation is performed after splitting each of the four datasets into 80%-20% for training and testing purposes. On the testing part of the dataset, the accuracy values of the model are 94.79%, 94.12%, 94.97%, and 96.67%, respectively, for DS1, DS2, DS3, and DS4. The encouraging results illustrate the model's ability to generalize diverse types of ECG signals to classify arrhythmias. To investigate the classification efficiency for each class, we manifested the AUROC curve for each class. Figure 7 exhibits the ROC curves where 7(a), 7(b), 7(c), and 7(d) corresponds to the ROC curves for DS1, DS2, DS3, and DS4, respectively.

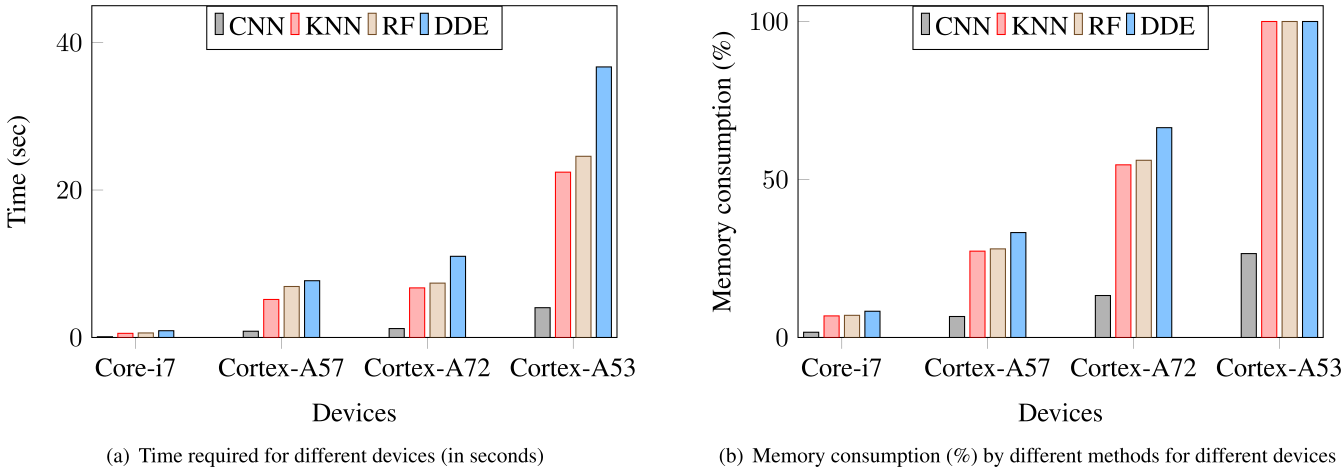


FIGURE 8. Required execution time and memory consumption of various methods on a workstation and different micro-controllers used as a proof-of-concept AI logic for the smart sensor.

For each dataset, the model demonstrated a high AUC score. The AUC scores for the four datasets are 0.9113, 0.9406, 0.9796, and 0.9340 for DS1 to DS4, respectively. The promising results prove the model's efficiency in distinguishing different classes of heartbeats to classify arrhythmia by employing a raw ECG signal.

Finally, we conducted the numerical analysis in terms of time delay and memory consumption (in percentage) of the proposed model and compared it with that of the traditional ML approaches (i.e., KNN, RF). Figure 8 illustrates the results obtained from the analysis. The initial experiment was conducted on a workstation with Intel Core i7, 3.00GHz CPU, 16 GB RAM, powered by Nvidia RTX 2060 GPU. We approximated the time and memory consumption required for different IoT devices to determine the model's potential to integrate with the logic-in-sensor. The microprocessor-based IoT devices we considered for the numerical analysis are Jetson Nano (Quad-core ARM A57 @ 1.43GHz), Raspberry Pi 4 (Quad-core Cortex-A72 @ 1.5GHz), and Raspberry Pi 3 (Quad-core Cortex-A53 @ 1.4GHz). Figs. 8(a) and 8(b) exhibit that the proposed CNN-based model can be beneficial for real-time analysis of the ECG signal as the model can perform efficiently with limited resources due to employing raw-ECG signal without any manual feature extraction.

The complexity analysis explained in Sec. V and the experimental outcomes presented in this section precisely confirm that the proposed lightweight ECG classification method can be considered as a viable solution for embedding intelligence into the resource-constrained ultra-edge IoT nodes. The proposed DL-LAC method's generalization aptitude was evaluated on four separate, publicly available real datasets by adopting multiple experimental settings. Promising experimental results signify that the proposed method performed with efficiency in all the experiments. Therefore, the model can be utilized for the ultra-edge IoT sensors to enhance healthcare services.

VII. CONCLUSION

Centralized cloud-based analytics and edge analytics on smart-devices are the traditional health monitoring approaches. To make smart health even smarter, in this paper, we focus on the necessity to go beyond the realms of conventional methods and investigate how to incorporate intelligence into the ultra-edge IoT sensors. As an example of the smart ultra-edge health monitoring, we selected arrhythmia (a cardiovascular disease) classification by analyzing the ECG signal. As the sensors are resource-constrained, we designed a deep learning-based lightweight heartbeat classification model named DL-LAC, that utilizes raw single-lead ECG to classify arrhythmia with encouraging efficiency. We compared the proposed method with traditional machine learning (e.g., KNN, random forest) and the DDE-based optimization technique. The proposed method's generalization ability was evaluated using four different datasets. The promising experimental outcomes manifest that the proposed deep learning model has the potential to be coupled with smart IoT sensors for ultra-edge computing to enhance the existing ECG monitoring system. Therefore, this research can be considered as a pioneering footprint to encourage the sensor foundries to consider embedding intelligence into IoT devices, and if it can be produced in mass production, the fabrication cost of the intelligent sensors can be significantly reduced.

REFERENCES

[1] A. Mohsen, M. Al-Mahdawi, M. M. Fouda, M. Oogane, Y. Ando, and Z. M. Fadlullah, "AI aided noise processing of spintronic based IoT sensor for magnetocardiography application," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, Jun. 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9148617.

[2] S. Mendis, P. Puska, and B. Norrving, "Global atlas on cardiovascular disease prevention and control. WHO," World Heart Fed., World Stroke Org., Geneva, Switzerland, Tech. Rep., Jan. 2011.

[3] S. Berrouiguet, M. L. Barrigón, J. L. Castroman, P. Courtet, A. Artés-Rodríguez, and E. Baca-García, "Combining mobile-health (mHealth) and artificial intelligence (AI) methods to avoid suicide attempts: The smartcrises study protocol," *BMC Psychiatry*, vol. 19, no. 1, pp. 1–9, Dec. 2019, doi: 10.1186/s12888-019-2260-y.

[4] Q. Yao, R. Wang, X. Fan, J. Liu, and Y. Li, ''Multi-class arrhythmia detection from 12-lead varied-length ECG using attention-based time-incremental convolutional neural network,'' *Inf. Fusion*, vol. 53, pp. 174–182, Jan. 2020, doi: 10.1016/j.inffus.2019.06.024.

[5] S. Sahoo, M. Dash, S. Behera, and S. Sabut, ''Machine learning approach to detect cardiac arrhythmias in ECG signals: A survey,'' *Innov. Res. Biomed. Eng.*, vol. 41, pp. 185–194, Jan. 2020, doi: 10.1016/j.irbm.2019.12.001.

[6] P. Kamble and A. Birajdar, ''IoT based portable ECG monitoring device for smart healthcare,'' in *Proc. 5th Int. Conf. Sci. Technol. Eng. Math. (ICONSTEM)*, Chennai, India, Mar. 2019, pp. 471–474, doi: 10.1109/ICONSTEM.2019.8918776.

[7] H. Kim, S. Kim, N. Van Helleputte, A. Artes, M. Konijnenburg, J. Huisken, C. Van Hoof, and R. F. Yazicioglu, ''A configurable and low-power mixed signal SoC for portable ECG monitoring applications,'' *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 2, pp. 257–267, Apr. 2014, doi: 10.1109/TBCAS.2013.2260159.

[8] C. H. Tseng, ''Coordinator traffic diffusion for data-intensive zigbee transmission in real-time electrocardiography monitoring,'' *IEEE Trans. Biomed. Eng.*, vol. 60, no. 12, pp. 3340–3346, Dec. 2013, doi: 10.1109/TBME.2013.2266373.

[9] M. Bansal and B. Gandhi, ''IoT & big data in smart healthcare (ECG monitoring),'' in *Proc. Int. Conf. Mach. Learn., Big Data, Cloud Parallel Comput. (COMITCon)*, Faridabad, India, Feb. 2019, pp. 390–396, doi: 10.1109/COMITCon.2019.8862197.

[10] Z. Yang, Q. Zhou, L. Lei, K. Zheng, and W. Xiang, ''An IoT-cloud based wearable ECG monitoring system for smart healthcare,'' *J. Med. Syst.*, vol. 40, no. 12, p. 1476, Oct. 2016, doi: 10.1007/s10916-016-0644-9.

[11] A. Rahman, T. Rahman, N. H. Ghani, S. Hossain, and J. Uddin, ''IoT based patient monitoring system using ECG sensor,'' in *Proc. Int. Conf. Robot., Elect. Signal Process. Techn. (ICREST)*, Dhaka, Bangladesh, Jan. 2019, pp. 378–382, doi: 10.1109/ICREST.2019.8644065.

[12] J. Bogatinovski, D. Kocev, and A. Rashkovska, ''Feature extraction for heartbeat classification in single-lead ECG,'' in *Proc. 42nd Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, Opatija, Croatia, May 2019, pp. 320–325, doi: 10.23919/MIPRO.2019.8757135.

[13] M. Alfaras, M. C. Soriano, and S. Ortín, ''A fast machine learning model for ECG-based heartbeat classification and arrhythmia detection,'' *Frontiers Phys.*, vol. 7, Jul. 2019, Art. no. 103, doi: 10.3389/fphy.2019.00103.

[14] H. Shi, H. Wang, Y. Huang, L. Zhao, C. Qin, and C. Liu, ''A hierarchical method based on weighted extreme gradient boosting in ECG heartbeat classification,'' *Comput. Methods Programs Biomed.*, vol. 171, pp. 1–10, Apr. 2019, doi: 10.1016/j.cmpb.2019.02.005.

[15] M. Baza, A. Salazar, M. Mahmoud, M. Abdallah, and K. Akkaya, ''On sharing models instead of data using mimic learning for smart health applications,'' in *Proc. IEEE Int. Conf. Informat., IoT, Enabling Technol. (ICIoT)*, Doha, Qatar, Feb. 2020, pp. 231–236, doi: 10.1109/ICIoT48696.2020.9089457.

[16] H. Yang and Z. Wei, ''Arrhythmia recognition and classification using combined parametric and visual pattern features of ECG morphology,'' *IEEE Access*, vol. 8, pp. 47103–47117, Mar. 2020, doi: 10.1109/ACCESS.2020.2979256.

[17] C. Lainscsek and T. J. Sejnowski, ''Electrocardiogram classification using delay differential equations,'' *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 23, no. 2, Jun. 2013, Art. no. 023132, doi: 10.1063/1.4811544.

[18] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng, ''Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network,'' *Nature Med.*, vol. 25, no. 1, pp. 65–69, Jan. 2019, doi: 10.1038/s41591-018-0268-3.

[19] M. Wess, P. D. S. Manoj, and A. Jantsch, ''Neural network based ECG anomaly detection on FPGA and trade-off analysis,'' in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Baltimore, MD, USA, May 2017, pp. 1–4, doi: 10.1109/ISCAS.2017.8050805.

[20] S. Saadatnejad, M. Oveisi, and M. Hashemi, ''LSTM-based ECG classification for continuous monitoring on personal wearable devices,'' *IEEE J. Biomed. Health Informat.*, vol. 24, no. 2, pp. 515–523, Feb. 2020, doi: 10.1109/JBHI.2019.2911367.

[21] S. Singh, S. K. Pandey, U. Pawar, and R. R. Janghel, ''Classification of ECG arrhythmia using recurrent neural networks,'' *Procedia Comput. Sci.*, vol. 132, pp. 1290–1297, Jan. 2018, doi: 10.1016/j.procs.2018.05.045.

[22] P. Pławiak and U. R. Acharya, ''Novel deep genetic ensemble of classifiers for arrhythmia detection using ECG signals,'' *Neural Comput. Appl.*, vol. 32, no. 15, pp. 11137–11161, Aug. 2020, doi: 10.1007/s00521-018-03980-2.

[23] S. Sakib, M. M. Fouda, Z. M. Fadlullah, and N. Nasser, ''Migrating intelligence from cloud to ultra-edge smart IoT sensor based on deep learning: An arrhythmia monitoring use-case,'' in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Limassol, Cyprus, Jun. 2020, pp. 595–600, doi: 10.1109/IWCMC48107.2020.9148134.

[24] A. Daamouche, L. Hamami, N. Alajlan, and F. Melgani, ''A wavelet optimization approach for ECG signal classification,'' *Biomed. Signal Process. Control*, vol. 7, no. 4, pp. 342–349, Jul. 2012, doi: 10.1016/j.bspc.2011.07.001.

[25] C. Lainscsek, P. Rowat, L. Schettino, D. Lee, D. Song, C. Letellier, and H. Poizner, ''Finger tapping movements of Parkinson's disease patients automatically rated using nonlinear delay differential equations,'' *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 22, no. 1, Mar. 2012, Art. no. 013119, doi: 10.1063/1.3683444.

[26] S. D. Greenwald, R. S. Patil, and R. G. Mark, ''Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information,'' in *Proc. Comput. Cardiol.*, Chicago, IL, USA, Sep. 1990, pp. 461–464, doi: 10.1109/CIC.1990.144257.

[27] G. B. Moody and R. G. Mark, ''The impact of the MIT-BIH arrhythmia database,'' *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, May 2001, doi: 10.1109/51.932724.

[28] S. D. Greenwald, ''The development and analysis of a ventricular fibrillation detector,'' Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 1986.

[29] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, ''PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals,'' *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000.

[30] F. Bouaziz, D. Boutana, and H. Oulhadj, ''Diagnostic of ECG arrhythmia using wavelet analysis and K-nearest neighbor algorithm,'' in *Proc. Int. Conf. Appl. Smart Syst. (ICASS)*, Medea, Algeria, Nov. 2018, pp. 1–6, doi: 10.1109/ICASS.2018.8652020.

[31] G. Chen, Z. Hong, Y. Guo, and C. Pang, ''A cascaded classifier for multi-lead ECG based on feature fusion,'' *Comput. Methods Programs Biomed.*, vol. 178, pp. 135–143, Sep. 2019, doi: 10.1016/j.cmpb.2019.06.021.

[32] *Testing and Reporting Performance Results of Cardiac Rhythm and ST Segment Measurement Algorithms*, Standard ANSI/AAMI EC57:1998/(R)2008, 2008.

[33] A. F. Agarap, ''Deep learning using rectified linear units (ReLU),'' Feb. 2019, *arXiv:1803.08375*. [Online]. Available: http://arxiv.org/abs/1803.08375

[34] Z. J. Wang, R. Turko, O. Shaikh, H. Park, N. Das, F. Hohman, M. Kahng, and D. Horng Chau, ''CNN explainer: Learning convolutional neural networks with interactive visualization,'' Aug. 2020, *arXiv:2004.15004*. [Online]. Available: http://arxiv.org/abs/2004.15004

[35] R. Kohavi, ''A study of cross-validation and bootstrap for accuracy estimation and model selection,'' in *Proc. 14th Int. Joint Conf. Artif. Intell. (IJCAI)*, Montreal, QC, Canada, Aug. 1995, pp. 1137–1145. [Online]. Available: https://dl.acm.org/doi/10.5555/1643031.1643047

SADMAN SAKIB received the B.Sc. degree in CSE from the Ahsanullah University of Science and Technology, Dhaka, Bangladesh, in 2017. He has been with the Department of Computer Science, Lakehead University, Thunder Bay, ON, Canada, as a Graduate Student, since 2019. His research interest includes intelligent computing/communication systems for mobile health (mHealth) applications for providing better health outcomes in northern Ontario and beyond. He was the Winner of the Best Student's Poster Award of the Canadian Institutes of Health Research (CIHR) category in the graduate conference held at the Graduate Student Conference—Poster Presentation, Research and Innovation Week, Lakehead University, 2020. He also received an internship opportunity through the Mitacs Accelerate Research Grant in 2020.

MOSTAFA M. FOUDA (Senior Member, IEEE) received the Ph.D. degree in information sciences from the Graduate School of Information Sciences, Tohoku University, Japan, in 2011. He has served as an Assistant Professor with Tohoku University. He was a Postdoctoral Research Associate with Tennessee Technological University, USA. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Idaho State University, ID, USA. He is also an Associate Professor with Benha University, Egypt. He has published more than 30 papers in IEEE conference proceedings and journals. His research interests include cyber security, machine learning, blockchain, the IoT, 5G networks, smart healthcare, and smart grid communications. He has served on the technical committees of several IEEE conferences. He is also a reviewer of several IEEE transactions and magazines and an Associate Editor of IEEE ACCESS.

ZUBAIR MD. FADLULLAH (Senior Member, IEEE) was an Associate Professor with the Graduate School of Information Sciences (GSIS), Tohoku University, Japan, from 2017 to 2019. He is currently an Associate Professor with the Department of Computer Science, Lakehead University, and the Research Chair of the Thunder Bay Regional Health Research Institute (TBRHRI), Thunder Bay, ON, Canada. His main research interests include emerging communication systems, such as 5G new radio and beyond, deep learning applications on solving computer science and communication system problems, UAV-based systems, smart health technology, cyber security, game theory, smart grid, and emerging communication systems.

Dr. Fadlullah is a Senior Member of the IEEE Communications Society (ComSoc). He received several best paper awards at conferences, including IEEE/ACM IWCMC, IEEE GLOBECOM, and IEEE IC-NIDC. He is currently an Editor of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY (TVT), *IEEE Network Magazine*, IEEE ACCESS, IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, and *Ad Hoc & Sensor Wireless Networks* (AHSWN) journal.

NIDAL NASSER (Senior Member, IEEE) received the Ph.D. degree from the School of Computing, Queen's University, Kingston, ON, Canada, in 2004. He worked at the School of Computer Science, University of Guelph, Guelph, ON, Canada, where he was the Founder and the Director of the Wireless Networking and Mobile Computing Research Lab. He is currently a Professor of Software Engineering with the College of Engineering, Alfaisal University, Saudi Arabia. He is also the Founder and the Director of the Telecommunications Computing Research Lab, Alfaisal University. He has authored 180 journal publications, refereed conference publications, and book chapters in the area of wireless communication networks and systems. He has been a member of technical programs and organizing committees of several international IEEE conferences and workshops. He received several outstanding research awards as well as a number of best paper awards. He is also serving as an Associate Editor for *IEEE Wireless Communications Magazine*, the *International Journal of Communication Systems* (Wiley), and IEEE COMMSOFT E-LETTER.

WALEED ALASMARY (Senior Member, IEEE) received the B.Sc. degree (Hons.) in computer engineering from Umm Al-Qura University, Saudi Arabia, in 2005, the M.A.Sc. degree in electrical and computer engineering from the University of Waterloo, Canada, in 2010, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, Canada, in 2015. During his Ph.D. degree, he was a Visiting Research Scholar with the Network Research Laboratory, UCLA, in 2014. He was a Fulbright Visiting Scholar with the CSAIL Laboratory, MIT, from 2016 to 2017. Subsequently, he joined the College of Computer and Information Systems, Umm Al-Qura University, as an Assistant Professor of Computer Engineering, where he currently holds an associate professor position. His THE MOBILITY IMPACT IN IEEE 802.11P INFRASTRUCTURELESS VEHICULAR NETWORKS article is among the most cited ad hoc networks journal articles list from 2016 to 2018. He is also an Associate Editor for the *Array* journal.