This is the author's version of an article that has been published in IEEE.
The final version of record is available at https://doi.org/10.1109/IoTaIS50849.2021.9359721

A Rigorous Analysis of Biomedical Edge Computing: An Arrhythmia Classification Use-Case Leveraging Deep Learning

Sadman Sakib
Dept. of Computer Science
Lakehead University
Thunder Bay, Ontario, Canada
ssak2921@lakeheadu.ca

Mostafa M. Fouda
Dept. of Electrical and Computer Engineering
Idaho State University
Pocatello, ID, USA
mfouda@ieee.org

Zubair Md Fadlullah
Dept. of Computer Science
Lakehead University
Thunder Bay, Ontario, Canada
zubair.fadlullah@lakeheadu.ca

***Abstract*—Biomedical Edge computing is an exciting area of interdisciplinary research involving the Internet of Medical Things (IoMT) sensors and devices with lightweight Artificial Intelligence (AI) logic. To address the rapidly growing need for smart and portable biomedical devices with localized decision-making capability, we present a proof-of-concept logic-in-sensor design with an arrhythmia analytics use-case. Existing signal processing techniques for arrhythmia analytics such as discrete wave transform (DWT) and non-linear delay differential equation (DDE) lead to high complexity and computational burden on biomedical edge devices due to expensive pre-processing steps. As a solution, we propose a deep learning-based lightweight arrhythmia classification method leveraging a customized one-dimensional (1-D) convolutional neural network (CNN). A rigorous analysis of the proposed method's performances and generalization potential are assessed using four publicly available datasets.**

***Index Terms*—Internet of things (IoT), arrhythmia, deep learning, convolutional neural network (CNN), smart sensor.**

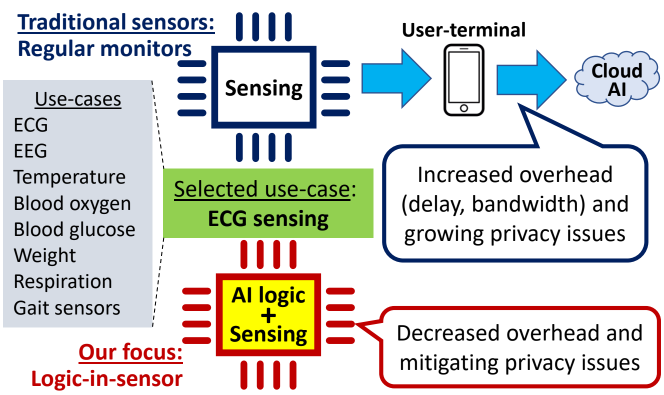


Fig. 1: The need for paradigm shift from traditional regular monitors relying on cloud-based medical analytics to biomedical edge devices with logic-in-sensor for localized analytics.

I. Introduction

The Internet of Things (IoT) [1] is anticipated to be a key enabler of the next generation smart society. With the surge of the IoT for biomedical applications [2] with an anticipated market value of $500 billion by 2025 [3], the need for a paradigm shift from the centralized, cloud computing toward the edge analytics for biomedical devices is rapidly growing [4]. In this paper, we address this pressing need by employing a logic-in-sensor concept [5], as depicted in Fig. 1, to bring localized intelligence to biomedical edge devices.

The contributions of this paper are summarized below.

1) We articulate the innate shortcoming of resource-constrained sensors in traditional biomedical devices and reveal the pressing need for a lightweight solution to move the AI analytics from the cloud to the edge sensors to facilitate continuous monitoring.
2) We choose the use-case of cardiac arrhythmia among a diverse set of use-cases. Arrhythmia is a major contributor of cardiovascular diseases [6] which account for 15–20% of global mortalities [7]. By demonstrating that the existing signal processing and Artificial Intelligence (AI) methods (i.e., machine learning) for analyzing time-series ECG (electrocardiogram) are not suitable for the resource-constrained edge devices, we present a one-dimensional (1-D) convolutional neural network (CNN)-based deep learning model, centrally trained and then transferred to the AI logic-in-sensor for arrhythmia classification.
3) Our proposed model can be used to classify heartbeats employing raw single-lead, and it does not require any noise-filtering of the ECG signal that makes the system lightweight and easy to integrate with the ultra-edge node.
4) Rigorous analysis of our proposed approach is presented and then corroborated by dataset-based empirical results. The viability of the proposed method is illustrated as a lightweight solution in an emulated biomedical edge device with several IoT setups. The model's accuracy, precision, F1-score, and particularly the generalization ability indicate that our approach can be extended toward other biomedical use-cases.

The rest of the paper is organized as follows. Section II concisely overviews the relevant work in the literature and presents a formal problem formulation. Our proposed methodology is presented in section III followed by extensive computational analysis and performance evaluation in section IV to corroborate the theoretical analysis. Concluding observations are presented in section V.

Z. M. Fadlullah is also affiliated with the Thunder Bay Regional Health Research Institute (TBRHRI).

This is the author's version of an article that has been published in IEEE.
The final version of record is available at https://doi.org/10.1109/IoTaIS50849.2021.9359721

II. Related Work and Problem Formulation

In the cloud-based ECG monitoring systems, several techniques are utilized, including feature extraction and classification. Discrete Wavelet Transformation (DWT) and Artificial Neural Network (ANN) were adopted for feature extraction, and for binary classification of heartbeat (normal or abnormal) are used in [8]. DWT and non-linear delay differential equations (DDE)-based optimization techniques [9], proposed in the literature for time-series ECG monitoring task, are unable to infer the system models in varying heart conditions adequately. An exhaustive exploration needs to be conducted to choose the most proper structure for the classification task using these approaches. Even a simple non-linear DDE for classifying ECG signals cannot be solved analytically as shown by the work in [10]. Because the selection of optimal time-delays and monomials is imperative for building an effective DDE-based classification system. To select an optimal DDE-based classification model, genetic algorithm was applied in [11]. In another study [12], sparse decomposition was employed for efficient feature extraction, and K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Probabilistic Neural Network, and Radial Basis Function Neural Network were applied for the classification task. However, these approaches are impractical in logic-in-sensors due to their significantly high computational complexity. Furthermore, as shown in Fig. 1, conventional bio-signal monitoring techniques (e.g., ECG, EEG (electroencephalogram), body temperature, blood oxygen, and so forth) require an Internet connection to interact with the servers for medical analytics. Hence, it consumes increased overhead (i.e., bandwidth and delay) for continuous biosignal monitoring, particularly when the number of users is high. Also, cloud-based analytics raises significant privacy issues regarding the collected health data. Therefore, the problem, in this paper, is to investigate how to design an automated, efficient, and lightweight system with localized intelligence capability and analyze its complexity to validate its deployment on biomedical edge devices.

III. Proposed Deep Learning-Based Lightweight Arrhythmia Classification System for Edge Devices

In this paper, we consider the use-case of arrhythmia classification in the edge. Inspired by the recent trend of AI-driven edge computing in other areas, we explore how to leverage deep learning as a viable approach [13], [14] for biomedical edge analytics. As depicted in Fig. 2, we construct a customized one dimensional CNN model, which does not demand any manual process for noise-filtering or feature extraction [15] unlike conventional techniques [16]. The reason behind using a 1-D CNN as the candidate technique is due to the notable ability in pattern recognition and the scalability of the trained model with continuous and large streams of biosignals from biomedical edge devices. CNN aggregates the ANN with the back-propagation technique, simplifying its complexity and decreasing the number of required parameters. Details of the proposed model's structure can be found in the authors' earlier work in [15]. The CNN-based deep learning paradigm automatically acquires unique patterns from the raw single-lead ECG and is able to deliver precise classification results for the unusual heartbeat detection. Note that this use-case can be generalized for various signals obtained from other biomedical edge devices also. The proposed ECG analysis model received an ECG signal ($X = [x_1, x_2, \ldots x_n]$) as the input , and outputs a sequence of labels presented as $Y = [y_1, y_2, \ldots y_n]$. Here, each y_i represents one of four different ECG heartbeat classes: {Normal, Supraventricular, Ventricular, Fusion}. Here, every class label is a part of the input ECG signal, and the output class labels combinedly resemble the full sequence of the signal.

Our proposed customized CNN model accepts raw one lead ECG as input and produces heartbeat classes as output, determining the standard or unusual heart status. It consists of two segments: an automated feature extraction (AFE) module consisting of nl_{AFE} 1-D convolution layers, and an automated classification (AC) module that processes the extracted features using nl_{AC} number of fully connected layer followed by the output layer. Systematic customization of our model leads us to reduce the filter size in each successive convolution layer by a factor of λ. Rectified Linear Unit (ReLU) is employed in the 1-D convolution layers as the activation function, while the softmax activation function is utilized in the output layer. A dropout layer is introduced to avoid data overfitting. After that, the batch normalization technique is employed to normalize the next layer's input for each mini-batch and accelerate the learning/training process.

The proposed model is trained in a centralized computational platform (e.g., cloud) and then the trained model is executed or run on the edge devices. A grid search with k-fold stratified cross-validation for hyper-parameter optimization is performed in the training phase to determine the optimal hyper-parameters (i.e., optimizer, and batch size, the number of epochs). Furthermore, hyper-parameter tuning is conducted on the number of layers and number of filters used in the convolution layers. On the other hand, in the inference or running phase on an edge device, the sensed biomedical data are locally classified using the trained model.

IV. Performance Analysis and Evaluation

In this section, we conduct a comprehensive performance analysis and performance validation of our proposed deep learning-based lightweight arrhythmia analytics on the edge IoT device.

A. Complexity Analysis

In this section, we conduct a rigorous analysis of the complexity and time-cost of our proposed deep learning-based lightweight bio-signal classification system, with the ECG monitoring use-case for arrhythmia detection. First, we analyze the complexity of our proposed model's training and inference steps, considering the number of mathematical operations needed by different modules of the model. The analysis primarily encompasses the mathematical analysis of the algorithm complexity in the training and inference stage through determining the recurrence of mathematical operations, such as addition (ADD), multiplication (MUL), and comparisons (COMP). Then, we also conduct the time complexity analysis of the inference and the training phases of the proposed model.

This is the author's version of an article that has been published in IEEE.
The final version of record is available at https://doi.org/10.1109/IoTaIS50849.2021.9359721

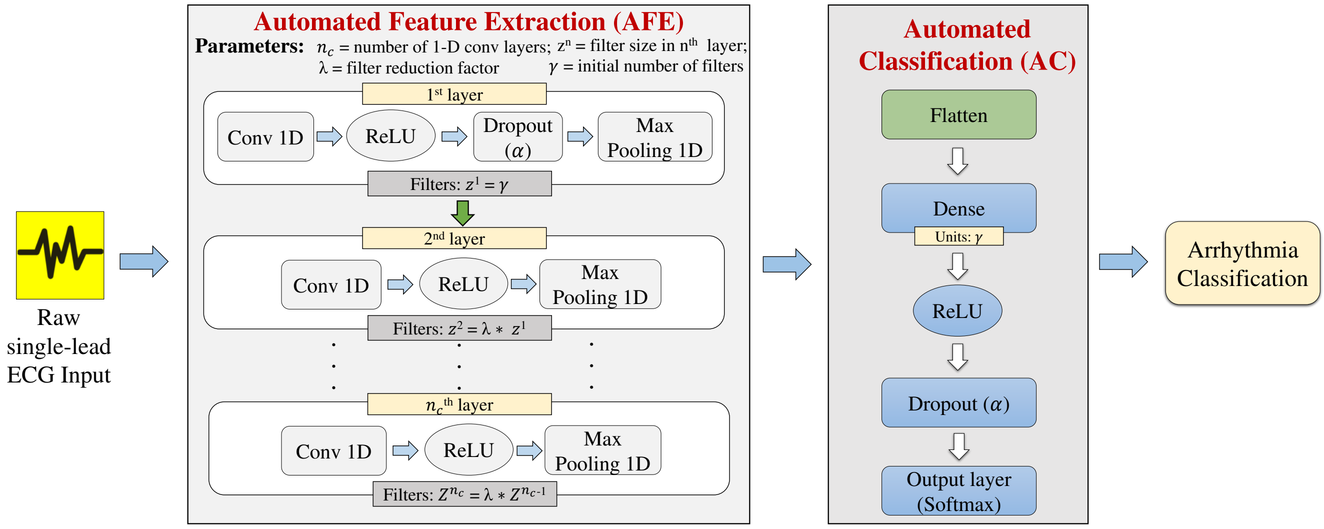


Fig. 2: Our envisioned customized training model for our considered biomedical edge computing use-case.

1) Analysis of the training phase in terms of mathematical operations: We perform computational overhead analysis in the training phase by assuming that the proper hyper-parameters are previously chosen by employing the grid search technique. In the data preparation (i.e., DP) phase, to load the training ECG data, the algorithm does not perform any mathematical operation; hence, we do not consider the computational overhead of the DP phase in this part of the analysis. Therefore, we divide the training phase's complexity analysis into three different divisions, namely, the required data-size validation (DSV) step, automated feature extraction (AFE) step, and the automated classification (AC) step. Thus, the overall computational complexity is given by:

$$C(Training) = C(DSV) + C(AFE) + C(AC). \quad (1)$$

To calculate the complexity of the ECG data-size validation (DSV) step, we mainly analyze the complexity of the ECG data validation. If the length of the considered training ECG trace is $len(X_{train})$, then the required number of comparisons is also $len(X_{train})$ since the condition will be validated for each ECG sample.

The computational overhead is determined for the feature extraction phase of the training procedure. The recurrence of addition (ADD), multiplication (MUL), and comparisons (COMP) instructions are considered in the analysis. For nl_{AFE} number of convolution layers, the number of required operations can be defined as eqs. 2 and 3.

$$\begin{aligned} C(AFE_{ADD}) = nl_{AFE} * \xi * (len(x^l)/B) \\ *((len(k^l)*len(x^{l-1})) - (len(k^l) - \eta) + 1) \\ - (len(x^{l-1}) - (len(k^l) - \eta)) * z^l)), \end{aligned} \quad (2)$$

$$\begin{aligned} C(AFE_{MUL}) = nl_{AFE} * \xi * (len(x^l)/B) * (z^l \\ *((len(k^l)*len(x^{l-1})) - (len(k^l) - \eta) + 1)) \\ + (nl_{AFE} - 1), \end{aligned} \quad (3)$$

where η, ξ, and B are the striding window value, the number of epoch, and batch sizes, respectively. x^l, k^l, and z^l denote the input, kernel, and the number of filters of layer l. Also, the number of comparisons required for the nl_{AFE} layers in the AFE phase can be denoted as eq. 4.

$$\begin{aligned} C(AFE_{COMP}) = \sum_{l=1}^{nl_{AFE}} (\xi * (len(x^l)/B) \\ *((z^l * len(x^l)) + (len(x^l) - (z^l - 1)))), \end{aligned} \quad (4)$$

where x^l implies the input of the l^{th} layer, z^l indicates the number of filters in the l^{th} layer of the AFE module. 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 needed is $(len(x^l) - (z^l - 1))$.

The extracted features set (F_x) of the AFE module is then relinquished to the AC module for the classification task. If the output of the i^{th} layer is γ^i, the computational complexity in terms of number of required mathematical operations for the i^{th} layer can be expressed as $(\gamma^i * F_x)MUL, (\gamma^i * (F_x - 1))ADD$. Thus, considering the number of fully-connected layers to be nl_{AC}, the computational complexity of this phase is given by eqs. 5 and 6.

$$C(AC_{ADD}) = \sum_{i=1}^{nl_{AC}} (\gamma^i * (F_x - 1)), \quad (5)$$

$$C(AC_{MUL}) = \sum_{i=1}^{nl_{AC}} (\gamma^i * F_x). \quad (6)$$

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. 7.

$$C(AC_{COMP}) = \sum_{i=1}^{nl_{AC}} (\gamma^i). \quad (7)$$

This is the author's version of an article that has been published in IEEE.
The final version of record is available at https://doi.org/10.1109/IoTaIS50849.2021.9359721

Hence, by substituting the equations into eq. 1, the overall computational complexity in terms of the number of mathematical operations and comparisons required in the training phase of the proposed algorithm can be expressed as eq. 8.

$$C(Training) = \begin{cases} ADD: & C(AFE_{ADD}) + C(AC_{ADD}) \\ MUL: & C(AFE_{MUL}) + C(AC_{MUL}) \\ COMP: & X_s^{train} + C(AFE_{COMP}) \\ & + C(AC_{COMP}). \end{cases} \tag{8}$$

2) Algorithmic Time Complexity Analysis of the Training Phase: Next, we analyze the algorithmic time complexity of the training phase. The upper bound of the time complexity can be determined as the combination of the data preparation (DP), data-size validation (DSV), automated feature extraction (AFE), and the automated classification (AC) steps. Considering the optimal hyper-parameter tuning is performed *a priori*, the overall time complexity of the training phase can be expressed as:

$$O(Training) = O(DP) + O(DSV) + O(AFE) + O(AC). \tag{9}$$

Assume that the initialization of the training step involves constant time, $O(1)$. Considering that there is X_s number of ECG samples in the training data location, the time cost can be expressed as $O(DP)$. Hence, the time complexity of the data preparation step is given by:

$$O(DP) = 2 * O(X_s) \approx O(X_s). \tag{10}$$

In the next step, the input vector X is verified by comparing with a threshold for ECG data size, δ, and if the length of the validated ECG signal is X_s^δ, then the time complexity can be expressed as Eq. 11:

$$O(DSV) = O(X_s^\delta). \tag{11}$$

The time complexity of the segment of the training phase that invokes the proposed 1-D CNN structure can be derived as the combination of time complexity of AFE and that of AC modules, defined in eqs. 12 and 13, respectively.

$$O(AFE) = O(k * \sum_{j=1}^{nl_{AFE}} (Z^j * (X_s^j * K_s^j)). \tag{12}$$

Here, k denotes the number of folds in the adopted stratified cross-validation, nl_{AFE} is the number of layers in the AFE module. Z^j, X_s^j, and K_s^j denote the number of filters, the input size, and the filter size for the j^{th} convolution layer, respectively. The extracted feature set will then be employed for the classification task in the AC module. For k number of folds, the time complexity in the AC phase can be denoted as eq. 13.

$$O(AC) = O(k * \sum_{j=1}^{nl_{AC}} (F_s^j * W_s^j) + N_b^j)), \tag{13}$$

where nl_{AC} is the number of layers in the AC module of the proposed model, F_s^j denotes the size of the features passed as input to the j^{th} layer, W_s^j represents the size of the weights utilized at each node of the j^{th} layer, and N_b^j indicates the number of nodes in the j^{th} layer.

Lastly, to express the training phase's overall time cost/complexity, we can substitute the corresponding values from eqs. 10, 11, 12, and 13 into eq. 9.

3) Algorithmic Time Complexity Analysis of the Inference Phase: The inference or running phase is conducted to infer classes of each testing ECG data employing the pre-trained model and then evaluating it. Assuming that the number of test data is X_s^{test} and the size of test data after validating the ECG signal is X_s^δ, the time complexity of the testing phase can be denoted as follows:

$$O(Inference) = O(X_s^\delta) + O(X_s^{test}). \tag{14}$$

Eq. 14 illustrates that the proposed DL-based model can generate results in linear time, which indicates that it can be utilized for lightweight arrhythmia classification on biomedical edge devices.

B. Performance Evaluation

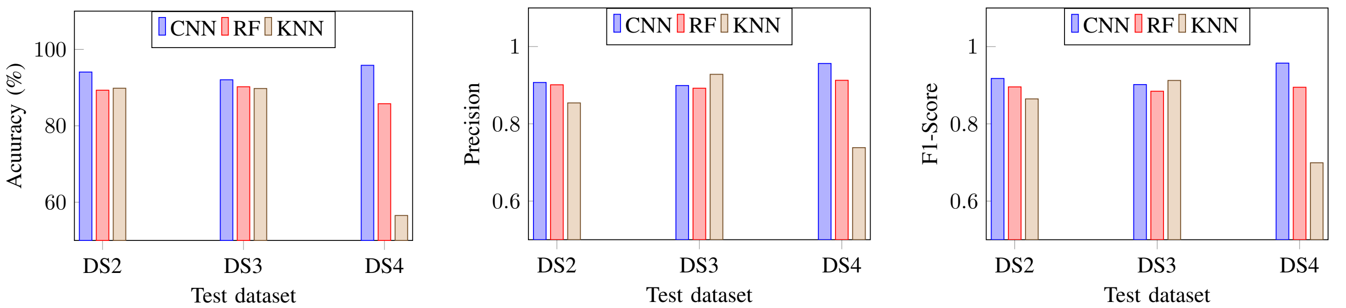
To corroborate the complexity analysis, we empirically evaluated the performance of the proposed deep learning model using the MIT-BIH Supraventricular Arrhythmia database, referred to as DS1 [17], with hyper-parameters tuning. The ECG signal samples were re-sampled at 257Hz before passing to the proposed model as input. 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. Therefore, we conducted further analysis using three convolution layers in the proposed deep learning-based architecture. Then, the model was extensively tested using three other public datasets from MIT-BIH repository, INCART 12-lead Arrhythmia database, and Sudden Cardiac Death Holter database, referred to as DS2 [18], DS3 [19], and DS4 [20], respectively. Then, our proposed model's generalization ability was verified by utilizing each of the four datasets individually for training and testing purposes using k-fold cross-validation. Fig. 3 illustrates the results for this phase. The proposed CNN-based model outperformed the traditional techniques as it achieves accuracy values of 94.07%, 92.04%, and 95.83% with DS2, DS3, and DS4 used as the test datasets, respectively, as indicated in Fig. 3a. A similar trend to the accuracy values is also observed in the case of precision and F1-Score values as illustrated in Fig. 3b and 3c.

Finally, in Table I, a numerical investigation is carried out to assess the proposed model's effectiveness in terms of execution time required and memory consumption in different IoT devices and compared the results with the traditional ML methods (i.e., KNN and RF). The initial experiment was conducted on a workstation with Intel Core i7, 3.00GHz CPU with the RAM of 16 GB, powered by Nvidia RTX 2060 GPU. We approximated the time and memory consumption required for different edge devices to corroborate the reasonable computational complexity of the proposed model for integration with the logic-in-sensor. The edge IoT devices considered for the numerical analysis are Jetson Nano (expressed as Cortex-A57), Raspberry Pi 4 (defined as Cortex-A72), and Raspberry Pi 3 Model B+ (Cortex-A53). Table I demonstrates the proposed model's lightweight nature since it requires a considerable amount of lower time and memory in diverse IoT configurations. While the proposed model outperforms

This is the author's version of an article that has been published in IEEE.
The final version of record is available at https://doi.org/10.1109/IoTaIS50849.2021.9359721

TABLE I: Required execution time and memory consumption of the proposed method and traditional ML techniques.

Method	Device	Time (sec)	Memory (%)	Pre-processing	Feature extraction	ECG lead
CNN	Core-i7	0.10	1.65	Not required	Not required	Single lead
	Cortex-A57	0.84	6.63			
	Cortex A72	1.21	13.27			
	Cortex-A53	4.04	26.55			
RF	Core-i7	0.61	7.01	3-phased noise filtering	Required	
	Cortex-A57	6.91	28.04			
	Cortex A72	7.37	56.08			
	Cortex-A53	24.57	100.00			
KNN	Core-i7	0.56	6.83	3-phased noise filtering	Required	
	Cortex-A57	5.15	27.32			
	Cortex A72	6.72	54.64			
	Cortex-A53	22.42	100.00			



(a) Accuracy values obtained employing the testing datasets.

(b) Precision values obtained employing the testing datasets.

(c) F1-Score obtained employing the testing datasets.

Fig. 3: Classification efficiency comparison of proposed model with traditional machine learning techniques using DS1 as the training dataset.

traditional methods in terms of performance and resource consumption, the notable part is that the proposed technique can achieve these higher performances without noise-filtering and manual feature extraction of the ECG. The results reveal that the model is robust in detecting heartbeats with high accuracy and lightweight because of using raw single-lead ECG. Thereby, the experimental results ensure that it is useful for real-time and efficient ECG analytics under limited resource conditions without the need for any manual feature extraction.

V. Conclusion

As smart health technologies continue to evolve, there is a pressing need for incorporating localized decision making capability into resource-constrained sensors on board various biomedical devices. In this paper, we addressed this issue by considering a specific use-case of arrhythmia classification by automated analysis of ECG and presented a customized one-dimensional convolutional neural network (1-D CNN) model. We have compared the proposed method to traditional techniques such as KNN and RF in ECG classification efficiency, time, and memory requirements. In order to assess the generalization efficacy of the proposed model, the proposed method is also evaluated on four different publicly open datasets. We demonstrated the viability of our proposed deep learning-based logic-in-sensor architecture (in terms of efficiency and generalizability compared to existing techniques) to be deployed in resource-constrained sensors in biomedical edge devices. Therefore, this research's encouraging outcomes can be heralded as an innovative start that will inspire sensor manufacturers to envision installing localized AI intelligence into the resource-constrained ultra-edge IoT sensors for the mass-scale, low-cost production of truly smart sensors.

References

[1] S. Sakib, T. Tazrin, M. M. Fouda, Z. M. Fadlullah and N. Nasser, "An Efficient and Light-weight Predictive Channel Assignment Scheme for Multi-Band B5G Enabled Massive IoT: A Deep Learning Approach," *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2020.3032516.

[2] S. Vadrevu and M. S. Manikandan, "A New Quality-Aware Quality-Control Data Compression Framework for Power Reduction in IoT and Smartphone PPG Monitoring Devices," *IEEE Sensors Letters*, vol. 3, no. 7, pp. 1–4, Jul. 2019, Art. no. 6001404, doi: 10.1109/LSENS.2019.2920849.

[3] "IoT in Healthcare Market Worth $534.3 Billion By 2025—CAGR: 19.9%." [Online]. Avail-

able: https://www.grandviewresearch.com/press-release/global-iot-in-healthcare-market

[4] R. K. Pathinarupothi, P. Durga, and E. S. Rangan, "IoT-Based Smart Edge for Global Health: Remote Monitoring With Severity Detection and Alerts Transmission," in *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2449–2462, Apr. 2019, doi: 10.1109/JIOT.2018.2870068.

[5] 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 International Conference on Communications (ICC)*, Dublin, Ireland, Jun. 2020, doi: 10.1109/ICC40277.2020.9148617.

[6] S. Mendis, P. Puska, and B. Norrving, "Global atlas on cardiovascular disease prevention and control," *WHO, World Heart Federation and World Stroke Organization*, 2011.

[7] N. T. Srinivasan and R. J. Schilling, "Sudden Cardiac Death and Arrhythmias," *Arrhythmia & Electrophysiology Review*, vol. 7, no. 2, pp. 111–117, Apr. 2018, doi: https://doi.org/10.15420/aer.2018:15:2.

[8] K. Balaskas and K. Siozios, "ECG Analysis and Heartbeat Classification Based on Shallow Neural Networks," in *Proc. International Conference on Modern Circuits and Systems Technologies (MOCAST)*, Thessaloniki, Greece, May 2019, doi: 10.1109/MOCAST.2019.8742072.

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

[10] C. Lainscsek and T. Sejnowski, "Electrocardiogram classification using delay differential equations," *Chaos (Woodbury, N.Y.)*, vol. 23, no. 2, Jun. 2013, Art. no. 023132, doi: 10.1063/1.4811544.

[11] 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: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 1, Art. no. 013119, Mar. 2012, doi:10.1063/1.3683444.

[12] R. Sandeep and K. C. Ray, "Sparse representation of ECG signals for automated recognition of cardiac arrhythmias," *Expert Systems with Applications*, vol. 105, pp. 49–64, Sep. 2018, doi: 10.1016/j.eswa.2018.03.038.

[13] Z. M. Fadlullah *et al.*, "State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, Fourthquarter 2017, doi: 10.1109/COMST.2017.2707140.

[14] F. Tang, Z. M. Fadlullah, B. Mao and N. Kato, "An Intelligent Traffic Load Prediction-Based Adaptive Channel Assignment Algorithm in SDN-IoT: A Deep Learning Approach," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5141–5154, Dec. 2018, doi: 10.1109/JIOT.2018.2838574.

[15] 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 *2020 International Wireless Communications and Mobile Computing (IWCMC)*, Limassol, Cyprus, Jun. 2020, pp. 595–600, doi: 10.1109/IWCMC48107.2020.9148134.

[16] U. Satija, B. Ramkumar, and M. Sabarimalai Manikandan, "Real-Time Signal Quality-Aware ECG Telemetry System for IoT-Based Health Care Monitoring," in *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 815–823, Jun. 2017, doi: 10.1109/JIOT.2017.2670022.

[17] S. D. Greenwald, R. S. Patil, and R. G. Mark, "Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information," in *[1990] Proceedings Computers in Cardiology*, Chicago, IL, USA, Sept. 1990, pp. 461–464, doi: 10.1109/CIC.1990.144257.

[18] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH Arrhythmia Database," in *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, May 2001, doi: 10.1109/51.932724.

[19] A. L. Goldberger, L. A. 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, doi: 10.1161/01.cir.101.23.e215.

[20] S. D. Greenwald, "The development and analysis of a ventricular fibrillation detector," Master of Science (MSc) thesis, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, May 1986.