IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

FPGA SoC Implementation of Adaptive Deep Neural Network based Multimodal Edge Intelligence for Internet of Medical Things

NIKHIL B. GAIKWAD[1], SMITH K. KHARE[2], DINESH MENDHE[3], HASAN MIR[4], SOKOL KOSTA[1], AND U RAJENDRA ACHARYA.[5,6],

[1]the Department of Electronic Systems, Aalborg Universitet, Frederikskaj 12, Copenhagen, Denmark (e-mail: nikhilgaikwad9423@gmail.com)
[2]Applied Artificial Intelligence and Data Science Unit, The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Campusvej 55 DK-5230 Odense, Denmark (e-mail: smkh@mmmi.sdu.dk)
[3]Office of Research Computing, Rutgers University, Shrub Oak, New Jersey, 10588, USA
[4] Department of Electrical Engineering, American University of Sharjah, Sharjah, United Arab Emirates
[5]School of Mathematics, Physics and Computing, University of Southern Queensland, Springfield, Queensland, Australia
[6]Centre for Health Research, University of Southern Queensland, Springfield, Queensland, Australia

Corresponding author: Smith K. Khare (e-mail: smkh@mmmi.sdu.dk).

This work did not received any funding

ABSTRACT In emergency healthcare services, accurate and timely decision-making is critical for the patient's life and death. The emergence of edge intelligence enables these service goals achievable for Internet of Medical Things (IoMT) compared with cloud-centric approaches. To assist medical personnel in intensive care units (ICU), we present the design of a network edge gateway that performs resource-efficient, real-time data analytics. We develop a cloud-configurable deep neural network (DNN) intellectual property (IP) core with an adaptable hardware architecture that executes four different types of analysis on an edge gateway. Our developed IP core adaptively switches from one architecture to another only in one clock cycle, based on the type of input features. The proposed IP core analyzes raw multimodal signals such as ECG, PPG, accelerometer, and other to discover anomalies in critically ill patients and their surroundings. We have validated the robustness of our developed model by comparing it with benchmark machine learning models and their previous implementations. The results show that our adaptive DNN model has obtained a software accuracy of 99.2% for ECG, 91.4% for PPG, 95% for activity classification, and 98.7% for smoke detection with a five-fold cross-validation strategy. Three versions of adaptive DNN IP cores (8-bit, 16-bit, 24-bit) are implemented on SoC/FPGA and compared together to study the effect of bit precision on accuracy, resource utilization, and power consumption. The developed adaptive DNN IP cores with 16-bits require 680 nanoseconds with a power consumption of 309 milliwatts for a single inference with a speed of 1.47 mega samples per second. Our analysis shows that the decentralization of intelligence in the IP core reduces data size from 96.25% to 98.75%. This flexible IP core has achieved significant power and resource utilization performance compared to independent implementation without compromising latency and throughput.

INDEX TERMS FPGA SoC, Deep Neural Network, Edge Intelligence, electrocardiogram, photoplethysmography, blood pressure, Embedded AI, Internet of Medical Things.

I. INTRODUCTION

THE emergence and advancements of the Internet of Things (IoT) are revolutionizing multiple domains such as healthcare [1]–[3], automobile [4], industries [5], and defense [6]. The development of IoT in healthcare is known as the Internet of Medical Things (IoMT), which seamlessly integrates all resources and assets [7]. IoMT applications use electroencephalogram, electromyogram, electrocardiogram (ECG), photoplethysmography (PPG), and other modalities with an accelerometer to monitor crucial parameters of pa-

tients [8]–[12]. IoMT network collects and analyzes patients' data, which can help to make quick decisions in emergency situations, such as in intensive care units(ICU) [13]. Decision-making in the ICU includes processing and analysis of multimodal data from patients and their surroundings. Distributed sensors are used to acquire multimodal physiological indicators from patients, and their surroundings are enough for conventional IoMT technology [14]. However, accurate processing and decision making in real-time are essential to achieve quality of service (QoS) for IoMT networks, especially for ICU and other similar scenarios [13].

A. BACKGROUND

Figure 1 presents an overview of the IoMT scenario with edge computing capability. The ICU contains several critical patients with multiple sensor nodes that continuously collect physiological information from the patient's body and surroundings. The edge gateway analyzes this multimodal information in real-time during an emergency and directly informs medical staff for quick actions [15]. In Figure 1, patient 3 (P3), indicates abnormalities in the information collected. These abnormal activities detected by the edge gateway alert the staff before forwarding the information to the IoMT cloud. In normal working conditions, the edge gateway is a conventional gateway that forwards and saves critical information to the IoMT cloud. The EI improves the response time of the IoMT network by analyzing and processing this information at the network edge to handle ICU emergencies in real-time. The IoMT network to be efficient, the models must provide accurate and quick

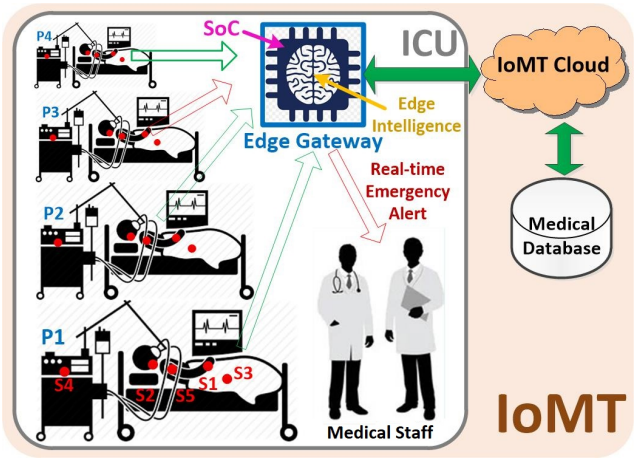


FIGURE 1. Multimodal Edge intelligence for ICU in internet of medical things.

decision-making. In addition, network performance depends on the flexible, real-time, and resource-efficient utilization of deployed EI [16]. Implementing the EI-based IoT network allows medical staff to act in real-time during emergency situation in the ICU. The EI is distributed intelligence responsible for generating and forwarding only meaningful information from the source (edge) of the network [17]. It significantly improves the performance of the IoMT network, specifically for ICU applications [18]. Advanced embedded devices like system on chip (SoC)/field programmable gate array (FPGA) are popular for handling such applications [19] [20], [21].

B. RELATED WORK

Several research studies have been published in recent years, focusing on the implementation of EI specifically on FPGA/SoC platforms [19]. Some of the important and most relevant EI based applications are discussed below. In computer vision applications, Hao et al. [22] introduced an FPGA/DNN co-design approach for object detection with higher IoU, lower power consumption, and improved energy efficiency. Similarly, Wisultschew et al. [23] demonstrated three DNN architectures on edge devices for object classification, favoring neural accelerators for a balance between performance and power consumption. In the privacy and security domain, Huang et al. [24] reported an adaptive crop growth monitoring system using EI, featuring adaptive cryptography for real-time data decryption, scalability, and an edge AI-based PDS estimator for crop detection. Likewise, Paul et al. [25] presents efficient LWE-based partially homomorphic encryption schemes integrated with IoT devices, demonstrating superior resource efficiency and throughput, making them ideal for edge-enabled IoT security. Similarly, Rebahi et al. [26] described the development of intrusion detection systems (IDS) for edge networks using FPGA-based acceleration. In the smart sensor area, Bartels et al. [27] proposed integration of advanced RNN into the TLM architecture for cow behavior using EI-enabled IoT. Gaikwad et al. [28] explore the implementation of FPGA/SoC-based EI for IoBT wearables, achieving promising results in the prediction of the gunshot angle with an MA-MLP model. In the healthcare domain, Alabdulhafith et al. [29] recently introduced a fog-computing-based clinical monitoring system for real-time remote prognosis and monitoring outside of the ICU within 30 days of discharge. After reviewing various application domains, it becomes evident that there are many other application areas, such as IoMT, where the use and deployment of EI is still unexplored. Specifically, a significant transformation is possible through the implementation of EI for the ICU scenario in IoMT. The proposed work is one of the first attempts to bridge this research gap and contribute significantly in this research domain.

The second important aspect is the effective deployment technique of EI; without it, EI is insignificant in high-end IoT applications. Below, we discuss some of the highly relevant implementation techniques and framework for different EI algorithms. Like, Lu et al. [30] demonstrated FPGA-based NDC systems with IMPFS, a POSIX-compliant file system optimized for neural networks, tested on a real PCM chip for low-energy EI support. Koutayni et al. [31] proposed a comprehensive framework, i.e. DeepEdgeSoC to optimize deep learning workflows. However, it lacks automated quantization bit selection and hardware optimization. Ajirlou et al. [32] introduced a novel concept of reusing a decoder as a

neural network (NN) to enable efficient edge intelligence (EI) on IoT devices, presenting a baseline Viterbi decoder with FPGA testing, a latency-efficient NN-over-decoder framework, and demonstrating its high performance scalability and low power overhead. Gao et al. [33] proposed EdgeDRNN that outperforms a 200W GPU+PC by running batch-1 RNNs significantly faster, boasting a power efficiency at least 4 times higher than commercial edge AI platforms and exploiting temporal sparsity with an effective 162 Op per clock cycle, while its scalability is facilitated through a standard AXI4 interface. Frasser et al. [34] introduced a compact and efficient architecture for a fully parallel SC-based CNN, which outperforms traditional binary logic and other SC implementations, demonstrating its effectiveness the implementation of complex edge-oriented CNNs with improved efficiency on a single FPGA chip. As edge gateways must process multimodal data due to the diverse sensory inputs in many IoT applications, EI implementation must support real-time multimodal data analytics. All of the above framework and architecture are unable to process diverse inputs. Therefore, the proposed Adaptive DNN architecture developed to overcome this challenge and open the opportunities for effective multimodal EI in other IoT applications.

C. RESEARCH GAPS AND MOTIVATION

Current studies suffer from multiple shortcomings, as conventional cloud-based IoT techniques limit quick and accurate decision-making, which is only possible with next-generation EI-based IoT networks [7], [15], [18]. Performance improvement and quick decision-making can be achieved by effective offloading from cloud to edge [35], [36]. However, the appropriate selection of the edge platform is crucial for executing the entire intelligence task on a single compact device [36]. IoMT offers multimodal data processing, but analysis of such data is challenging due to implementation challenges in multiple machine learning (ML) models [37]. Additionally, IoMT facilitates the minimization of bandwidth to improve response times in ICU environments [38]. Therefore, it is crucial to implement the ML algorithm with low computational latency [38]. To overcome this, edge ML implementations must support timely software and hardware upgrades in response to the rapid development of AI and IoT technology. Additionally, analyzing multimodal signals requires different architectures and topologies for decision-making. Developing and deploying such topologies on hardware requires a high degree of flexibility and substantial resources. Additionally, if a unified model is developed for analyzing such multimodal signals, it must be compact enough, with minimal nonlinear operations and a fixed number of weights and biases. However, designing such a unified architecture for classifying multi-modal signals on a single-edge device is challenging [35], [36]. This requires an adaptable and lightweight model with optimal layers for making quick and accurate decisions at the edge. Therefore, this paper presents the development of an adaptable deep neural network (DNN) to detect three physiological conditions and smoke. The model adaptively updates its weights and biases in conjunction with the network topology, depending on the type of classification task. Our developed adaptive DNN model is deployed on a single IP core chip with minimal resources and layers. The proposed work is one of the initial attempts to explore deployment techniques for flexible EI on SoC/FPGAs that handle multi-model data in IoMT scenarios [39], [40].

D. THESIS OF THE PAPER

This work demonstrates the end-to-end deployment strategy of EI for the IoMT gateway. The current work uses four types of multi-modal raw sensory data, which can be expanded according to the need. Physiological data including, ECG and PPG, along with data from an accelerometer can be collected from the patient body. General information like temperature, humidity, and pressure can also be collected from the surrounding. Edge gateway uses this multi-modal sensory information to generate real-time emergency alerts for medical staff. This research work incorporates four distinct alert types: detection of abnormal cardiac activity through ECG, classification of blood pressure using ECG and PPG, and identification of patient activities using a wearable accelerometer. Finally, smoke detection using temperature, humidity, and pressure. This sensory information is analyzed with the help of ML algorithms like decision trees, naive Bayes, support vector machine (SVM), k-nearest neighbor (KNN), and DNN classifier. The best-performing DNN classifier is selected for hardware deployment at the EI in IoMT [41]. The implementation of novel hardware architecture for a DNN IP core is done on the FPGA/SoC. The state-of-the-art Vitis Model composer design tool has been used for designing edge DNN IP core [42]. Three IP cores having different bit precision (8, 16, and 24-bit) are designed and implemented on the recently launched hardware board Kria KV260 AI starter kit [43]. The system performance including model evaluation, resource utilization, power consumption, latency performance, and EI performance are also been studied. All IP cores are highly flexible because of configurable weight and bias architecture. These DNN parameters can be configured from the cloud in the long run without modifying hardware configuration. All IP cores can be easily integrated with any AMD Xilinx-based device because of the light AXI interface that makes this DNN IP core scalable [44]. The switching between different inference types requires one clock cycle, eliminating the latency required for the parameter transfer from memory to FPGA/SoC in the multi-modal inference.

E. CONTRIBUTIONS AND SUMMARY OF THE PAPER

The key contributions of the present work are as follows:

- Developed heterogeneous ML models for classifying three physiological activities using ECG, PPG, and accelerometer data along with smoke detection.
- Generated an adaptive edge DNN classifier capable of performing four types of analysis within a cloud-

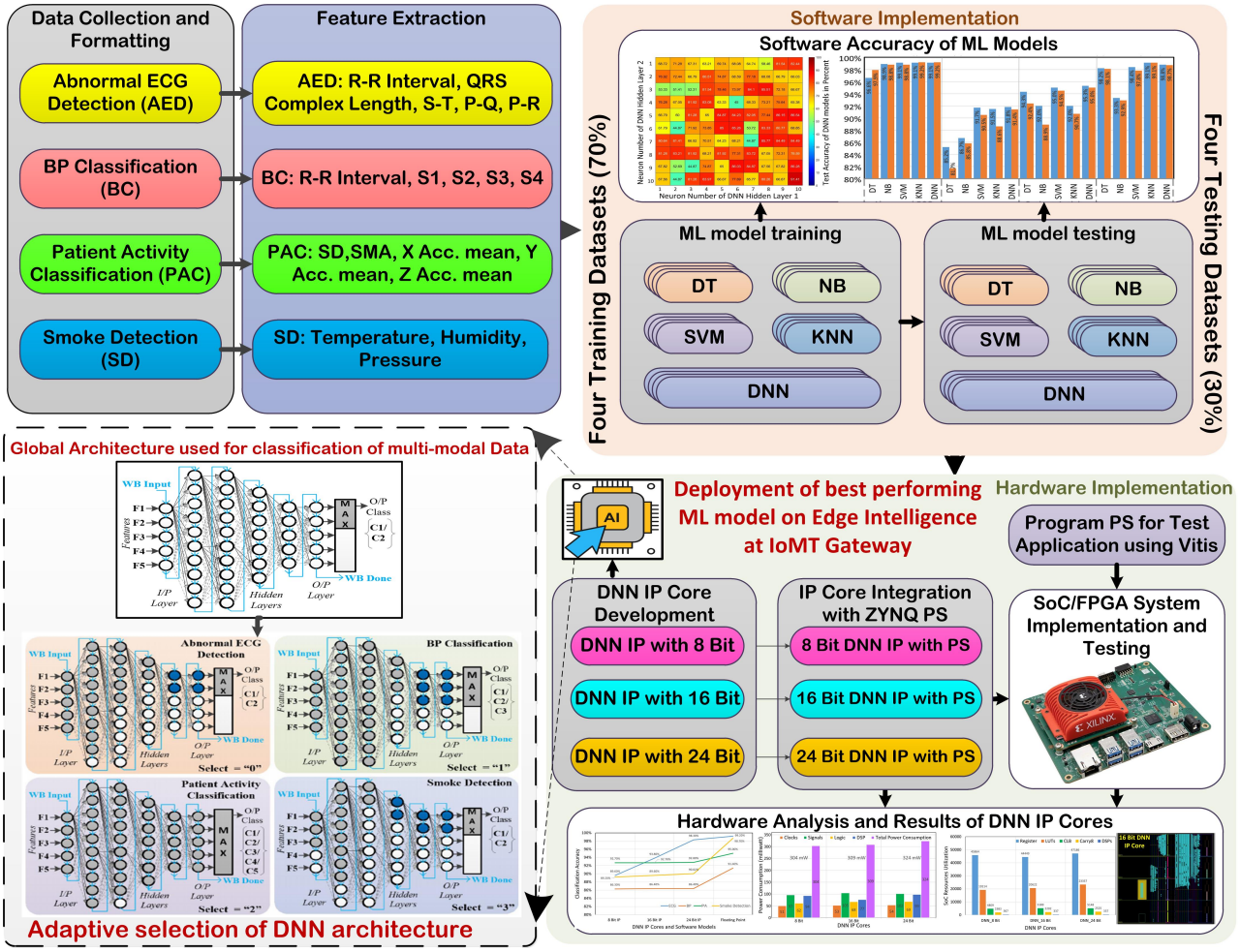


FIGURE 2. Schematic representation of DNN IP core development process.

configurable single DNN IP core with a single clock switching speed between different analysis.

- Presented the development and deployment of ML models for real-time edge intelligence for multimodal scenarios.
- To the best of our knowledge, this is one of the first attempt to present SoC/FPGA based development and deployment of an adaptive DNN IP core to deploy multimodal EI in IoMT.

The summary of findings on our developed adaptive DNN model are as follows:

- Obtained a software accuracy of 99.6%, 91.4%, 95%, and 98.7% using ECG signals, PPG signals, activity classification and smoke detection, respectively our developed DNN model.
- Achieved the inference speed of 1.47 mega samples per second by our DNN IP core.
- Developed model is fast and power efficient, as the DNN IP cores with 16 bits require 680 nanoseconds (ns) with 309 milliwatts (mW) power consumption for a single inference.
- Provides data size reduction of 96.25% to 98.75% with our developed DNN IP core.
- A reduction of 29% to 60% in resource utilization and a 47.76% decrease in power consumption have been achieved by the DNN IP core compared to independently implemented DNN models, all while maintaining the same level of accuracy and throughput reduction.

The reminder of the paper is organized as follows: Section II presents methodology for EI. Section III covers details of datasets, pre-processing, training, IP core integration, and SoC/FPGA system implementation. The results are discussed in Section IV. Finally, Section V concludes this paper.

II. METHODOLOGY

The proposed work emphasizes the independent analysis of several key physiological and behavioral parameters, namely heart condition (normal and abnormal), blood pressure (normal, low, and high), physical activities (walking, sitting, standing, lying, and transitioning), and smoke detection. Based on a detailed review of existing literature in each of these domains, appropriate algorithms were selected to ensure compatibility with an adaptive architecture while main-

taining high classification accuracy. For heart condition analysis, a variety of approaches have been reported with strong performance, including signal characteristic-based methods achieving up to 99% accuracy [45], SVM with 96% [46], CNN with 97.33% [47], LS-SVM with 98.21% [48], KNN with 98.40% [49], and DNN with 98.33% [50]. Blood pressure classification has also been explored using several techniques, such as MLP (i.e. DNN) with 91.30% accuracy [39] , AdaBoost with 92% [51], KNN variants reaching 86.80% [52] and 88.49% [53], CNN-LSTM with 66% [54], GRU and CNN with 78% [55], and HHT achieving 93.54% [56]. In the domain of activity classification, studies report multi-classifier frameworks with 91.70% accuracy [57], Random Forests with 86.20% [58], and MLP (DNN) models achieving 94.60% [21] and 94.13% [59]. More complex approaches, such as the hybrid EPS+LDA+MCSVM framework, have achieved an accuracy of 98.67% [60], whereas RCA has reached 81.32% [61]. For smoke detection, DNN models reached 97.52% accuracy [62], SVM 86% [63], ES-RNN 98% [64], Random Forests 86.11% [65], and Linear Regression as high as 99.60% [66]. Overall, these previous works highlight DNN as a consistently high-performing algorithm across all domains. Its ability to deliver strong results while supporting adaptive architectures makes it a particularly well-suited algorithm in this work. The steps and schematic representation of our proposed model is shown in Figure 2. The development of DNN-based EI analyzes four different types of raw sensory information in real-time.

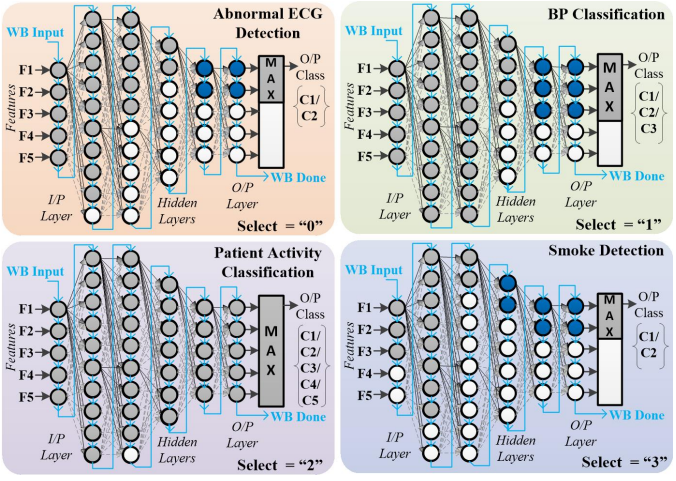


FIGURE 3. Illustration of hardware architecture of adaptive DNN with multi-modal analysis for EI in IoMT.

Instead of relying solely on existing literature and its reported results, we decided to evaluate the performance of various machine learning techniques across all four targeted analyses ourselves. The outcome of this evaluation further validates and encourages the use of DNN in this work, as explained in detail in Section IV-A and illustrated in Figure 6. Most DNN architecture decisions are typically guided by a combination of expert judgment, empirical testing, and other heuristic methods to determine topological aspects such as the number of neurons and layer configurations [67] [68]. In our work, we performed a comprehensive evaluation by systematically exploring all possible combinations of DNN architectures with up to four hidden layers and a maximum of ten neurons per layer. The full set of evaluated models is provided in Appendices G–J of the supplementary material. Although several configurations achieved similar levels of accuracy, we intentionally selected the smallest architectures that provided the best trade-offs among accuracy, energy consumption, and resource utilization. This approach enabled efficient deployment on resource-constrained edge devices. The effectiveness of our selection strategy is further validated through a comparative analysis with 23 state-of-the-art methods, as presented in Table 2 (page 8), where our selected models consistently demonstrate superior accuracy. These results highlight the practicality and robustness of our systematic topology selection process.

The best-performing DNN model was deployed on hardware, utilizing adaptive switching of network topology along with weights and biases, and employing three different bit-width representations (8, 16, and 24-bit). For DNN hardware implementation, prior studies have adopted similar data precision levels-16-bit precision in [69] and [21], 32-bit in [70], and 8-bit in [71] and [39]. These works demonstrate that while higher precision (e.g., 32-bit) provides greater numerical stability [70], lower-precision implementations (8-bit and 16-bit) [21], [39], [69], [71] are increasingly preferred due to their efficiency in hardware acceleration. Accordingly, the 8-, 16-, and 24-bit representations were selected for exploration in this work, based on the proven performance-to-efficiency trade-offs reported in the literature and the demands of hardware deployment. Finally, results and analysis is performed in the final stage. The developed IP core can require approximately one-fourth of FPGA resources compared to an independent DNN hardware implementation, provided all four models have topologies that are the same or close to the final IP core topology. This condition leads to the optimum performance of the IP core with minimum resource utilization per inference type. The paper uses a bottom-up approach for hardware design, namely: architecture design test, and verification, and results with discussion. The elementary unit of the architecture is an adaptive neuron model. An adaptive maximum is the termination block of the network, and the integrated network of these neurons is an adaptive DNN hardware design. In this work, the final adaptive DNN architecture is modeled from elementary neurons using the Xilinx Vitis model composer toolbox in the MATLAB Simulink design environment. The four DNNs are integrated into a single dynamic, configurable, and adaptive DNN architecture having updating weight and biases as shown in Figure 2.

A. INTRODUCTION TO ADAPTIVE DNN HARDWARE DESIGN

Our developed DNN IP core classifies four heterogeneous feature vectors implemented on the edge gateway. The multi-modal sensor nodes are ECG for cardiac activities detection

[72], PPG for abnormal blood pressure detection [73], accelerometer for activity detection [74], and temperature, humidity, and pressure sensor data for smoke detection [75]. In the next step, features of these sensor data i.e., R-R interval, QRS complex length, S-T, P-Q, and P-R intervals, are used for abnormal ECG classification. For blood pressure detection, R-R interval and S1, S2, S3, and S4 are used as features. The features of standard deviation and signal magnitude area from the X-axis accelerometer and mean values from X-, Y-, and Z-axis accelerometers are used for patient activity detection. Finally, temperature, humidity, and pressure are used for smoke detection. These features are given to fined tuned independent DNN models for classification tasks by switching the pre-trained and pre-loaded weight/biases (WB) of the respective DNN model. The switching is done based on the types of feature vectors. The switching requires one clock cycle to select the desired functionality of the IP core, helping in reducing latency and increasing the throughput of ML computation specifically for the multi-modal data types in the constrained environment. It also eliminates the need for WB loading from memory during the inference, removing the bottleneck imposed due to chip data transfer.

In the first step, WB are initialized and loaded in the DNN design. As shown in Figure 3, the WB input is only a vector serial input port for DNN. It contains eleven elements, each of 16-bit with a decimal point at bit 14, covering a range of 1.99 to -1.99, respectively. As each layer can have a maximum of ten neurons, there are 10 elements in the WB vectors used to store weights (W), with the last element reserved for the bias (B) of the respective neuron. The sky blue colored arrow shows the flow of WB, where all WB are loaded at a desired location. The WB Done = 1, indicates completion of initialization. After that, the DNN IP core gets ready for the real-time edge inference. When inference starts, DNN checks for the data type using the select line input and configures the DNN design according to its status in a single clock cycle. The raw data generated from respective sensors is tagged with the sensor type number ranging from 0 to 3. These tags are automatically used to determine the select line status of the IP core during the inference. If the Select line = 0, the DNN design works as an abnormal ECG detector (5-9-5-2), for the Select line = 1, it performs BP classification (5-10-10-3). For the Select line = 2, the DNN performs patient activity detection (5-10-9-7-5-5), and when the Select line = 3, it works as a smoke detector (3-8-2). The DNN design enables the accommodation of all possible topologies under its maximum possible configuration i.e., 5-10-10-7-5-5. The MAX function accordingly is accordingly adapted to the select line status. As shown in Figure 3, three types of neurons are present in each configuration. Grey-colored neurons represent active neurons that participate in actual computation. White neurons represent deactivated neurons without any contribution in classification, and blue neurons acts as a buffer connected to the MAX block. Following subsection presents the detailed AMD Xilinx model composer-based hardware design of elementary neurons and DNN.

B. HARDWARE DESIGN OF A NEURON

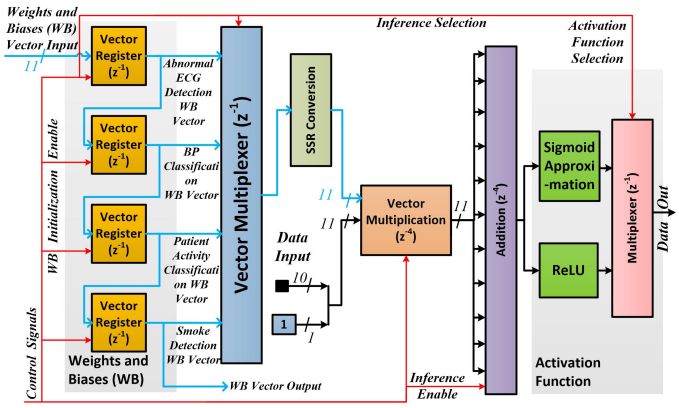


FIGURE 4. Hardware architecture of a neuron that work as an elementary block of the DNN IP core.

The architecture of an adaptive DNN is similar to the inherent DNN architecture designed using super sample rate (SSR) blocks available for vector operations in the model composer toolbox (See Appendix B, Figure 2) Figure 4 shows the hardware architecture of a neuron. As shown in Figure 4, the WB vector inserts eleven parallel elements into four registers used to analyze four classification scenarios. The last register passes the WB value of the next neuron using the WB output port. The control signal is composed of the select line input and the inference enable signal. The select line selects receptive WB register bank and activation function block while performing the inference calculation. The inference enable signal enables the multiplication and addition block. We internally use only two enabled signals in a neuron architecture to reduce hardware resource utilization. The correct WB register is selected using vector MUX control by an inference selection line connected externally to the select line port. As IP has to switch four types of WB, we have used a 4:1 multiplexer for the switching element. The WB vector is multiplied by an input vector having the same length using a vector multiplier. The resultant vector is added together in the next step and given to an activation function. The mathematical expression for each neuron used in our DNN model is denoted in (1).

$$N_o = PLAN\left(\sum_{i=1}^{n} I_i W_{ij} + B_j\right) \tag{1}$$

where N_o is the output, I_i is the input vector, W_{ij} is the weights matrix, and B_j is the bias value of the neuron.

The hardware design for a perfect nonlinear function requires huge FPGA/SoC hardware resources [46]. The PLAN activation function offers a flexible piecewise-linear approximation that outperforms ReLU in modeling complex nonlinear interactions while avoiding the vanishing gradients associated with sigmoid or tanh functions. Its computational simplicity allows for rapid training, and its interpretable segments make it ideal for crucial areas. As a result, we prefer

PLAN over alternative activation methods in our implementation process to reduce the utilization of hardware resources [46]. Equation (2) shows the mathematical formulation of the PLAN function [46].

$$\begin{aligned} PLAN\,(\beta) &= 0.25 \times |\beta| + 0.5 \quad for \quad 0 \le |\beta| < 1 \\ &= 0.125 \times |\beta| + 0.625 \quad for \quad 1 \le |\beta| < 2.375 \\ &= 0.03125 \times |\beta| + 0.84375 \quad for \quad 2.375 \le |\beta| < 5 \\ &= 1 \quad for \quad 5 \le |\beta|. \end{aligned} \tag{2}$$

C. HARDWARE ARCHITECTURE OF DNN

Figure 5 shows the architecture of the adaptive DNN IP core with appropriate control signals. The IP core includes an inference select line signal for selecting the desired classifier and other control signals i.e., WB vector, WB initialization clock, and inference enable. An extra register bank is included in adaptive DNN architecture to hold the feature inputs at the rising pulse of inference-enable signals. As adaptive DNN completes its operation, the inference done signal goes high, and the output appears on an inference output port. The type of inference is a delayed select line signal, used as the acknowledgment signal. The Vitis model composer token is used to set relevant parameters related to the IP cores like, targeted hardware, clock frequency, etc; the AXI interface is also selected using this token (See Appendix A, Figure 1).

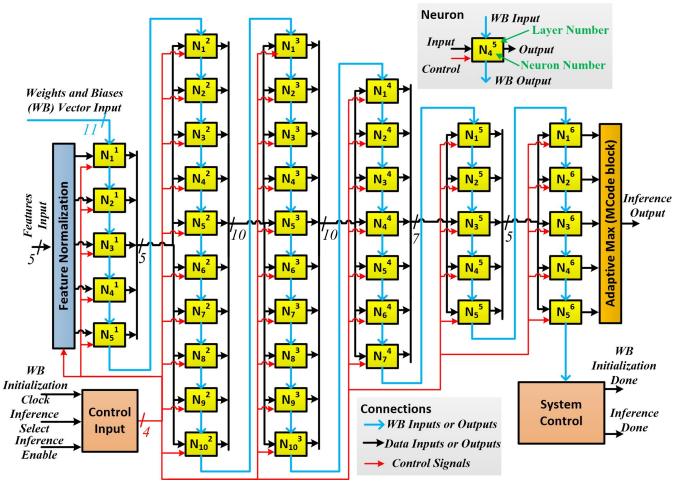


FIGURE 5. Hardware architecture of a DNN IP core for deployment of edge intelligence in IoMT.

A maximum of five features is used as input to the IP core. In conventional DNN, softmax function decides the output with the highest probability. In adaptive DNN, we have replaed softmax with a MAX function performing the same operation. The MAX function must be flexible to classify feature vectors according to the select line's status. As shown in Figure 5, the output of third-layer neurons is connected with available inputs of four MCode blocks. Each MCode is a maxima function designed independently for all four types of analysis. The number of adaptive maximum blocks depends on number of classes for each inference (two for ECG, three for BP, five for activity, and two for smoke detection). Other controls and status signals are updated as per the functionality of an adaptive DNN model. A total of 68 clock cycles are required to complete the DNN operation when the output class is set and inference goes high. This DNN supports six hidden layers with 5, 10, 10, 7, 5, and 5 maximum neurons in receptive layers. This DNN hardware design synthesized in an IP core that is integrable into any AMD Xilinx digital system.

D. DYNAMIC SWITCHING OF DNN ARCHITECTURES

At the initiation phase of the IP core, the Weight and Biases (WB) initialize when the inference enable signal goes low, and the WB initialization clock is triggered at system boot of the FPGA/SoC. All values from the WB vector are then transferred into the DNN model. This WB string encapsulates the complete set of weights and biases for all four DNN models. It is formatted such that each WB value is loaded into its designated location within the architecture by the end of the initialization process. As illustrated in Figures 4 and 5, the Weight and Biases (WB) Vector Input serves as the input path for the WB string to enter the neurons and DNN structure. Similarly, the WB Vector Output serves as the output path, forwarding the WB string vectors to neighboring neurons until initialization is complete. Once WB initialization is done, the model becomes ready for inference. The normalization block scales the input feature vector to a range of -1 to 1. Edge inference begins when the select signal is loaded with a corresponding value indicating the type of inference to be performed. As shown in Figure 5, the Inference Select signal facilitates dynamic switching during the IP core's active operation. This signal is broadcast to all neurons in the network and is used to configure the "Vector Multiplexer" block in each neuron, as illustrated in Figure 4. The same signal also determines the activation function to be used. As the inference selection signal is physically connected to the multiplexers in each neuron, and since there is only a one-clock-cycle delay for multiplexer switching, this IP core achieves single-cycle inference switching latency. During each inference operation, a single vector register and a single activation function are selected based on the value of the select signal. For example, when Select = 0, the adaptive DNN is configured for abnormal ECG detection using a 5-9-5-2 architecture, where the first vector register and a PLAN function are selected based on the status of the select line. This mechanism of dynamic switching similarly supports the remaining classification tasks.

III. HARDWARE IMPLEMENTATION AND TESTING

This section discusses crucial aspects of the end-to-end deployment of EI for IoMT. The section contains dataset details, hardware (FPGA/SoC) deployment of the DNN IP core, and hardware test-bed setup for IP core evaluation.

TABLE 1. Details of the multi-modal datasets

Datasets	Abnormal ECG Detection	BP Classif.	Patient Activity Classif.	Smoke Detection
Dataset Source	MIT-BIH Arrhythmia [76]	Cuff-less BP Estimation [76] [51]	UCI Human Activities Data Set [77]	Smoke Detection Datasets [66]
Sensor Type	Modified Limb Lead II (MLII)	PPG and ECG Sensor	3-axial Accelero -meter	Temperature, Humidity, Pressure
Subjects	13	25	21	1
Training (70%)	560	2600	5390	43842
Testing (30%)	240	1820	2310	18790
Total Features	800	4420	7700	62632

A. DATASETS AND TRAINING

The paper focuses on the deployment of EI and the multi-modal datasets specifically for ICU scenarios. To the best of our knowledge, no open-source dataset is available with four distinct modalities. Collecting such a diverse dataset for ICU like scenarios is a challenging task. Therefore, in our work, we have analyzed independent publicly available datasets to develop the system. This work assumes that the dataset is collected from a multi-modal ICU setup. Table 1 shows the details of each dataset used in our work. These datasets are only used for the demonstration of DNN-based edge inferences for IoMT. The EGC, PPG probes, and accelerometer are placed on the patient body and smoke detection uses commonly available ICU monitoring parameters. The ECG dataset from MIT-BIH Arrhythmia [76] consists of 13 subjects and MLII signals, featuring 800 features derived from 560 training samples and 240 testing samples. Blood pressure classification utilizes a cuffless estimation dataset [76] [51] that combines PPG and ECG sensors from 25 subjects, resulting in 4,420 features across 2,600 training and 1,820 testing samples. For activity recognition, the UCI Human Activities dataset [77] captures 3-axis accelerometer data from 21 subjects, contributing 7700 features based on 5390 training and 2310 testing samples. The smoke detection dataset [66], based on environmental sensors that measure temperature, humidity, and pressure, stands out due to its data from a single subject, but with a high volume of samples: 43842 for training and 18790 for testing, totaling 62632 features.

In our work, we have used five features for the ECG, BP, and activity analyses while three features for smoke detection, as already discussed in Section II-A. All raw features are chosen based on literature study, which helps to reduce the size of DNN due to the fewer features [21] [39] [73] [66]. Four independent DNN models are trained using scaled conjugate gradient training function with variable learning rate using a five-fold cross-validation technique. The training procedure for software simulations of the DNN algorithm is given as follow:

1) WB initialization: All WB are initialized to random values between 1 to -1 in the DNN.
2) Forward propagation: The input features are propagated through the DNN layers. The result of each layer's output is passed to the next layer's input.
3) Loss calculation: The output of the final layer is compared with the desired output to evaluate the loss. In the current work, we used the cross-entropy loss function for training all four DNN models mentioned below.
4) Back-propagation: The loss is back-propagated through the network to evaluate the gradients of the WB to the loss.
5) Update WB: WB of each layer is updated using optimization to minimize the loss.
6) Repeat steps (b)-(e): All steps are repeated multiple times for many epochs until the model's performance gets saturated.
7) DNN testing: the trained DNN model can do predictions output class on the new features.

The Levenberg–Marquardt algorithm (trainlm) was used for training, providing fast convergence, with cross-entropy as the performance (loss) function for all classification tasks. Before training, the input data was normalized to the [-1, 1] range using mapminmax, and the output labels were one-hot encoded. For 5-fold cross-validation, the dataset was partitioned into five equal subsets. In each fold, four subsets were used for training and one for validation, with the rotation ensuring that each subset served as validation once. Key training hyperparameters included a maximum epoch limit of 1000 and a minimum gradient threshold of 1×10^{-7}, applied independently within each fold. Training in each fold was terminated based on reaching the epoch limit, achieving a predefined performance goal, or when minimal progress in the gradient was observed.

Loss Calculation: The output of the DNN final layer is compared with the desired output to evaluate the loss. In the current work, we utilized the cross-entropy loss function for training all four DNN models. In training deep DNN for classification, cross-entropy loss serves as a guide, assessing the disparity between the model's predicted class probabilities and the actual class labels. By minimizing this loss through optimization, the DNN adjusts its internal weights to align its predictions with the true data better, ultimately enhancing classification accuracy. Minimizing the cross-entropy loss during training is a common objective that guides the model in learning improved representations and enhancing its classification accuracy. There are two main types of cross-entropy: the first is Binary Cross-Entropy, which is used for binary classification problems with two possible outcomes, as mentioned in Equation (3). The second is Categorical Cross-Entropy, used for multiclass classification problems with more than two possible outcomes, given in Equation (4).

$$\text{Binary Cross Entropy (BCE)} = y \log(p) + (1-y)\log(1-p) \quad (3)$$

$$\text{Categorical Cross Entropy (CCE)} = \sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \quad (4)$$

where M is the number of classes, log is the natural logarithm, y is a binary indicator (0 or 1) representing the class label, c is the correct classification for observation o, and p is the predicted probability that observation o belongs to class c.

More detailed information on backpropagation is discussed in Appendix G of the Supplementary Material. We have used offline training with the benchmark ML models to test the robustness. The best-performing DNN model is optimized for hardware deployment. Finally, they are merged in a single DNN IP core for multi-modal EI.

B. IP CORE INTEGRATION AND SOC/FPGA SYSTEM IMPLEMENTATION

The WB of all four DNN models are organized in the IP core initialization file according to the IP core architecture. In the present work, we have designed three independent IP cores with different data width precision having the same architecture. AMD Xilinx model composer is used to develop DNN IP cores with 8-bit, 16-bit, and 24-bit. It helps to understand the implementation strategy and performance constraints for the DNN IP core. After designing and testing all DNN designs using the Simulink model-based tool, we generated DNN IP cores with an AXI light interface. We have created three independent digital systems using AMD Xilinx Vivado that uses ZYNQ UltraSCALE-based processing systems (PS). We integrated three DNN IP cores in three PS using Vivado IP integrator. The 16-bit DNN IP core is integrated with PS, external UART, and other PS subsystems (See Appendix C, Figure 3). The current IP cores are running on a 100 MHz clock frequency generated by the PS system. We have deployed DNN IP cores one by one on recently launched Kria K26 SOM hardware tested based on the test-bed setup (See Appendix E). The paper uses the AMD Xilinx Vitis software platform to write test application code for PS and debugged using Vivado hardware manager. We used the integrated logic analyzer (ILA) core to debug the DNN IP core, which helps to record the real-time latency performance of the DNN IP core while running on the actual Kria hardware board. External UART has been used for the feature data and output class transfer between hardware and LabVIEW-based user interface designed for system testing. Complete step by step hardware deployment flow for FPGA/SoC based edge intelligence is mentioned in Appendix D. Complete details of Test bed setup has been discussed in Appendix E of the supplementary material.

IV. RESULTS AND DISCUSSION

In the first subsection analysis of different ML models in software simulation is presented. The best-performing ML model is used for hardware implementation and analysis. After that, analysis of performance measurement in terms of hardware classification accuracy, FPGA resources utilization, and power consumption for different versions of DNN IP cores with various bit precision are presented. Finally, last two subsections discuss the classification latency and DNN IP core performance for EI.

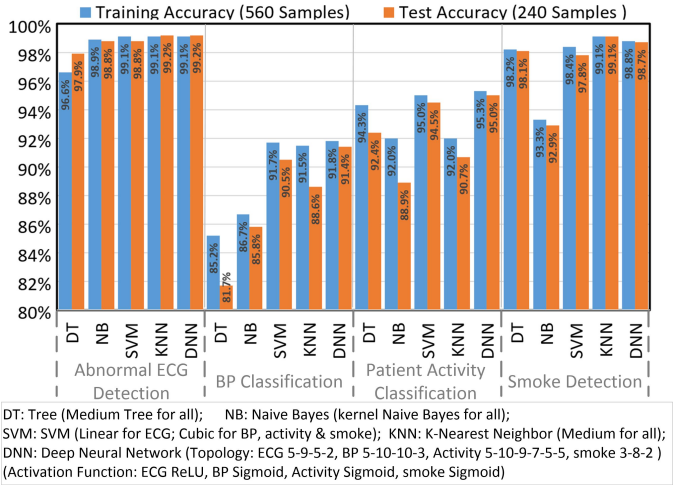


FIGURE 6. Performance comparison between different ML used for analysis of multi modal datasets using five-fold cross-validation technique.

A. SOFTWARE ACCURACY OF ML MODELS FOR MULTI-MODAL DATASETS

Various DT, Naive Bayes, SVM, KNN, and DNN-based ML models have been compared to decide the best-performing classifier for edge deployment. The paper used five-fold cross-fold validation, with 70% data utilized for training and 30% data for testing. The topology of proposed DNN models has been selected based on extensive analysis of all possible combinations of DNN models with maximum four hidden layers and maximum ten neurons in each layer (See Appendix G, H, I and J of supplementary material). The training and testing accuracies obtained for ML models is shown in Figure 6.

For abnormal ECG detection, KNN and DNN produced the highest accuracy of 99.1% in training and testing, respectively. For BP classification, DNN is the best-performing classifier with a training and testing accuracy of 91.8% and 91.4%, respectively. For patient activity classification, DNN classifier yields the highest training and testing accuracy of 95.3% and 95%, respectively. Finally, for smoke detection, the KNN outperformed other ML algorithms with an accuracy of 98.8% and 98.7% during training and testing, respectively. As shown in Figure 6, the performance of DNN is consistent for most of the analysis. Therefore, we choose DNN classifier for edge deployment on the IoMT edge gateway. The fine-tuned DNN models have different topology and activation functions to achieve the highest accuracy for respective applications. As demonstrated in Table II, the proposed DNN models have been compared with ML algorithms previously published in the literature for chosen applications. The baseline performance comparison reveals that the

TABLE 2. Comparison of Independent DNN Models with Existing Works Utilizing Various ML Algorithms for Respective Applications

Applications	Works	Algorithm	Accuracy
Abnormal ECG Detection (Binary Class Output)	[45]	Signal Characteristics	99%
	[46]	SVM	96.00%
	[47]	CNN using Inception V3	97.33%
	[48]	LS-SVM	98.21%
	[49]	KNN	98.40%
	[50]	DWT + DNN	98.33%
	This Work	DNN (5-9-5-2)	99.20%
BP Classification (Three Class Output)	[39]	MLP	91.30%
	[51]	AdaBoost	92%
	[54]	CNN-LSTM	66%
	[52]	KNN	86.80%
	[56]	HHT and 2D CNN	93.54%
	[53]	K-NN	88.49%
	[55]	GRU and CNN	78%
	This Work	DNN (5-10-10-3)	91.4%
Patient Activity Classification (Five Class Output)	[57]	Multi-classifier	91.70%
	[58]	Random Forests	86.20%
	[21]	MLP	94.60%
	[59]	MLP	94.13%
	[60]	EPS+LDA+MCSVM	98.67%
	[61]	RCA	81.32%
	This Work	DNN (5-10-9-7-5-5)	95%
Smoke Detection (Binary Class Output)	[62]	DNN	97.52%
	[63]	SVM	86%
	[64]	ES-RNN	98%
	[65]	Random Forests	86.11%
	[66]	Linear Regression	99.60%
	This Work	DNN (3-8-2)	98.70%

proposed DNN models outperform other recently published works. Furthermore, the performance of these DNN models remains consistent across all applications, which motivates us to integrate all DNN models into a single IP core for signal processing. However, implementing each DNN model independently requires higher hardware resources. Therefore, we decided to implement only one hardware design i.e DNN IP core that leverages the advantage of inherent DNN architecture. DNN IP core initialized its WB parameters of all four inferences, when it boots up. It minimizes the need of WB movements from memory to processing blocks during each multi-modal inference. The DNN IP core performs all four types of analysis in a single IP core, reducing hardware resource utilization and power consumption.

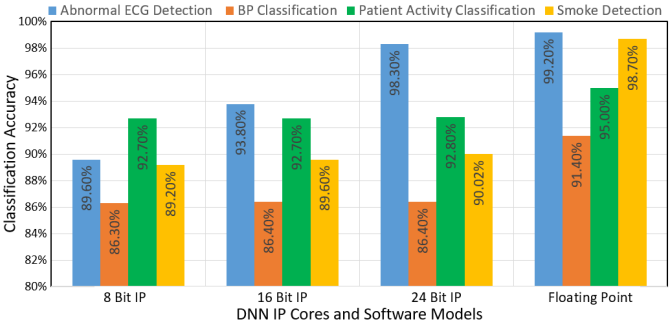


FIGURE 7. Comparison of 8 bit, 16 bit, 24 bit, and floating point precision DNN IP cores for multi-modal edge inference in IoMT.

B. HARDWARE ACCURACY OF DNN IP CORES

As stated earlier, different bit precision helps us to decide the optimum IP core for overall performance. Therefore, we have compared the accuracy obtained for floating point and 8-, 16-, and 24-bit precision to get optimum IP code design, as shown in Figure 7. From Figure 7, it is clear that the accuracy of DNN models with floating point precision is the highest. But floating point precision requires higher hardware resources eliminating its usability for edge hardware deployment. Therefore, the proposed work uses fixed point precision to implement the DNN IP core. It is observed that the accuracy drop of about 0.9% to 8.68% results due to fixed point precision due to quantization error and activation function approximation. For smoke detection, the accuracy loss is potentially the highest because of fewer features involved in the classification. In the case of abnormal ECG detection, the accuracy loss seems insignificant over others.

In the current implementation, we utilized the basic Post-Training Quantization (PTQ) technique, wherein we trained all DNN models with floating-point precision and subsequently quantized these models to fixed-point precision [78]. The accuracy of these fixed-point DNN models can be further enhanced by employing more advanced quantization techniques, such as Quantization-Aware Training (QAT), during the offline training phase [78]–[82]. Among the various DNN IP cores, 16-bit precision shows considerable accuracy for all four types of inferences. Therefore, DNN IP with 16-bit is the right choice for EI with decent accuracy in IoMT.

C. HARDWARE RESOURCE UTILIZATION OF DNN IP CORES

All versions of DNN IP cores are synthesized for K26 SOM hardware after the final hardware implementation of the IP cores is complete. As shown in Figure 8, the register utilization of both 8-bit and 16-bit IP cores is almost the same, but the 24-bit version requires higher registers. The implementation reveals that the area required for the DNN IP with 8-bit precision is less than that for the 16-bit and 24-bit IP cores. The utilization of look-up tables is slightly reduced from 24- to 8-bit IP cores. The main components of the implementation are configurable logic block (CLB) and

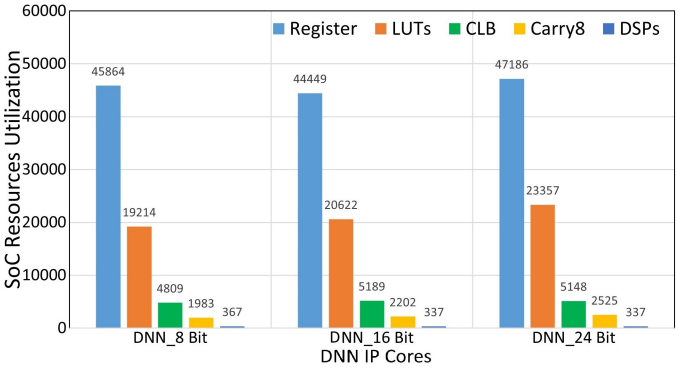


FIGURE 8. FPGA/SoC resource utilization of DNN IP core with 8, 16, & 24 bit.

digital signal processing (DSP) block, covering most of the SoC/FPGA area. The utilization of CLB and DSP is consistent across the IP cores without any variation. However, DNN IP cores with 16-bit precision show balanced resource utilization compared to other IP cores, considering accuracy and power consumption. Additionally, Figure 9 illustrates the hardware implementation of all three IP cores deployed in register transfer level (RTL), reflecting the resource utilization of designs into the area utilization in the FPGA fabric.

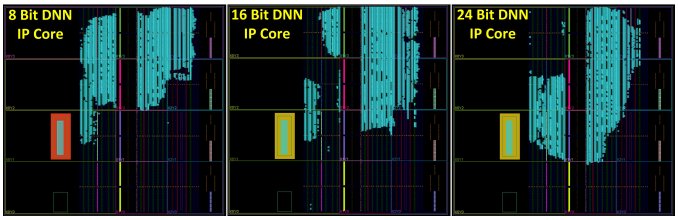


FIGURE 9. FPGA/SoC area utilization of DNN IP core with 8, 16 & 24 Bit.

D. POWER CONSUMPTION RESULTS OF DNN IP CORES

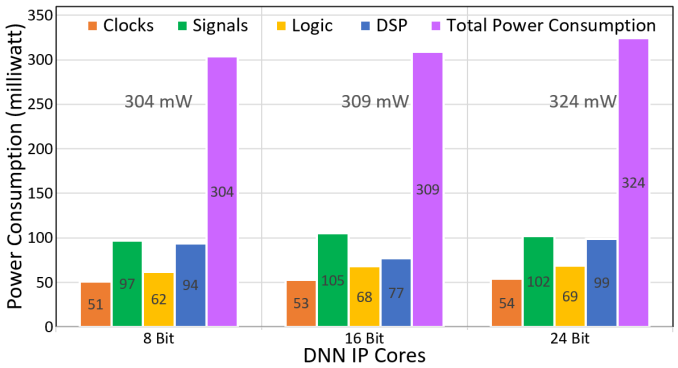


FIGURE 10. Power consumption of DNN IP core with 8, 16 & 24 bit precision.

All DNN IP cores are operating at a frequency of 100 MHz to cope with the real-time performance of IP cores. As shown in Figure 10, the total power consumption for a 24-bit DNN IP core is higher than 8-bit and 16-bit. The difference between the power consumption of 8-bit and 16-bit IP cores is 5 mW. Figure 10 shows the breakout of total power consumption, where DSP power consumption varies more across IP cores. Clock, signal, and logic changes are relatively less for all IP cores. The result shows that the power consumption of 8- and 16-bit precision DNN IP cores is lower than 24-bit precision DNN IP cores. A narrower data path generally requires less power due to reduced switching activity, while wider data paths contribute to higher power consumption. DSP power consumption in the 16-bit setting differs from the 24-bit IP core due to data path width, despite using the same DSP resources. Signal power consumption focuses on dynamic power associated with signal transitions, with key contributors being switching activity, high fanout, interconnect length, clocking networks, data buses, and design structure. Post-synthesis parameters of DNN IP cores vary in interconnect length, clocking networks, and design structure among different models, making this the main reason for the highest signal power consumption of 16 bit IP cores.

E. LATENCY PERFORMANCE OF DNN IP CORE

To monitor the AXI bus signals, we used the ILA core and debugging facility provided by AMD Xilinx tools to check the latency of the DNN IP core. As shown in Figure 11, four values are set on the `Inference_Select` line with respective features on the `Feature_Input` line. After setting the data and control signals, the DNN IP core starts computing edge inference once the inference done `Inference_Done` signal goes high. `Inference_Out` port shows the output class for the respective features. A total of 680 ns is required for each inference which is fast for EI requirements. 2.72 μs is required for all four types of multi-model analysis, where only 10 ns are required for the switching between different modalities.

F. ADAPTIVE DNN IP CORES PERFORMANCE FOR EI

All useful and significant information has been analyzed at the network edge itself because of the distributed intelligence. DNN IP core based EI contributes to boosting IoMT performance in all possible directions, like a huge reduction in cloud storage requirements, communication security enhancement because of small data size, and reduction in communication power requirements. As shown in Table 3, the DNN IP core can perform four different types of inferences in a single IP core with an adaptable topology. This novel DNN architecture saves 47.76% of power consumption and 29.26% to 60.54% of hardware resources utilization compared with independent implementation of DNN models. While achieving this performance, the percent throughput reduction and classification accuracy are same in both cases. Also, the saving of hardware resources and power consumption indirectly contributes to reduction in system and cooling cost. Throughput reduction is critical in IE-enabled IoMT, especially for real-time performance, data security, and cloud storage when billions of sensor nodes are streaming data

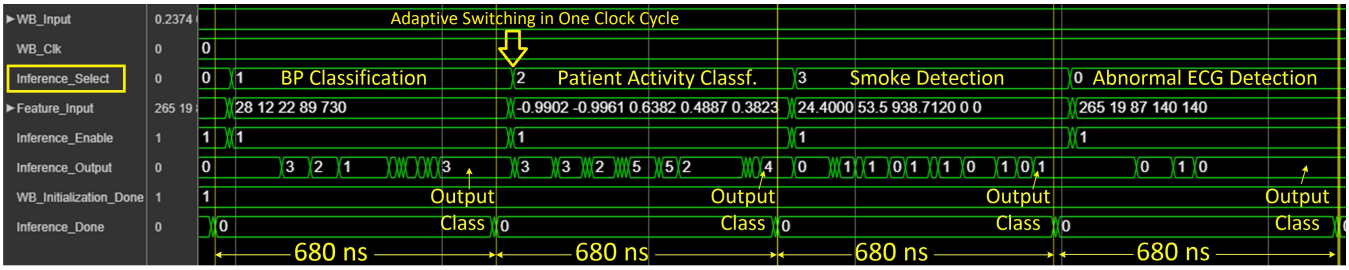


FIGURE 11. Signal and data variations in AXI interface of DNN IP core captured for multi-modal inference running on the hardware.

TABLE 3. Performance Comparison of 16 Bit Adaptive DNN IP core with Independent Hardware Deployment independent DNN models.

Performance Parameters		Independent DNN models Deployment	Adaptive DNN IP Deployment	Percent Saving
Resource Utilization	Registers	62830	44449	29.26%
	LUTs	42455	20622	51.43%
	CLB	9830	5189	47.21%
	Carry8	5578	2202	60.52%
	DSPs	854	337	60.54%
Estimated Total Power Consumption (milliwatt)		580	303	47.76%
Percent Throughput Reduction	Abn. ECG Detection	98.75%		NA
	BP Classif.	97.50%		NA
	P. Activity Classif.	96.25%		NA
	Smoke Detection	97.92%		NA
Hardware Classif. Accuracy	Abn. ECG Detection	93.80%		NA
	BP Classif.	86.40%		NA
	P. Activity Classif.	92.70%		NA
	Smoke Detection	89.60%		NA

simultaneously. In our work, we have evaluated the throughput reduction in edge-to-cloud communication traffic for the four types of analyses achieved due to the DNN IP core, as shown in Table 3. The edge gateway has achieved 96.25% to 98.75% throughput reduction due to the adaptive DNN IP core. Percentage throughput reduction is the ratio of the difference between raw features throughput and inference throughput to the raw features throughput [83] (Details in Appendix F). The most important advantage is the real-time response of the network because it eliminates network delay during cloud communication. The DNN IP core required 680 ns by consuming only a 309 mW power for a single inference. It indicates that the IE-enabled IoMT can alert medical staff about emergencies with less than a microsecond delay.

Finally, we have compared the critical parameters of the proposed hardware model with the existing models as shown in Table 4. As already mentioned in the introduction, the present work is one of the first attempts of multi-modal EI development for IoMT. Therefore, it is very difficult to compare our DNN IP core with available research on exactly the same ground. However, we compare different neural networks implemented for various applications with different topologies, data precision, operating frequency and other parameters in Table 4. It is evident from Table 4 that our designed DNN IP cores have obtained better latency and throughput performance. The DNN IP core is the most flexible IP core that supports adaptive topology for multi-modal inference with flexible WB that helps to configure DNN from the cloud, which is not possible in other designs. The power consumption and resource utilization of DNN IP cores are relatively high compared to existing implementations due to their large topology and data precision. Therefore, the operational flexibility of this IP core is achieved at the cost of some designed overhead, as reflected in both the power consumption and resource utilization of the IP core. Most of the existing models implemented single network topology with fixed hardware architecture, which do not support EI upgrades after deployments. The real scenarios require multimodal analysis but these single network topologies designs cannot be sufficient to perform efficient multimodal inferencing at the network edge. Also, employing different topologies for different types of classification demands higher hardware resources, limiting the applicability of DNN in IoMT scenarios. Our DNN IP core overcomes these limitations by using a single IP core capable of adapting different network topologies and WB according to data modalities. In addition, The DNN IP core neurons and layers can be easily scaled up as per requirements because of the easy design and troubleshooting technique provided by Vitis Model Composer. This IP core can be smoothly integrated in any AMD Xilinx FPGA SoC device with sufficient hardware resources because of standard AXI-Lite interface. Therefore, This makes our model robust, effective, and resource-efficient compared to existing models for real-time implementation in critical ICU environments with IoMT.

V. CONCLUSION

The developed system has provided an effective hardware solution for real-time and power-efficient EI inference. The proposed model performs the classification of abnormal ECG signals, BP detection, patient activities, and smoke detec-

TABLE 4. Comparison of 16 Bit DNN IP cores with existing neural networks implementation.

Performance Parameters		[69]	[21]	[70]	[71]	[39]	This Work
Topology		14-19-19-7	7-6-5	8-16-12-8-4	4-8-3-3	7-6-5	5-10-10-7-5-5
SOC FPGA Hardware		XC6SLX 45CSG	Artix-7 35T	ZYNQ7020	EP4CE C7N	Artix-7 35ticsg324-1L	KV260 SOM
Data Precision		16 Bit	16 Bit	32 Bit	8 Bit	8 Bit	16 Bit
Frequency (MHz)		67	100	100	77.59	10	100
Latency (microseconds)		146500	0.27	3.254	1.16	31	0.68
4* Resource Utilization	LUT	1032	3466	15738	0	3244	20622
	DSP	7	81	176	0	79	337
	Register	4590	1069	18740	8582	759	44449
	CLBS	2175	NA	NA	42499	NA	5189
Power (Watt)		0.294	0.241	NA	NA	12	0.309
Throughput (MSPS)		7×10-6	3.7	0.307	862	0.032	1.47
Support Adaptive Topology		No	No	No	Yes	Yes	Yes
Weight and Biases		Fixed	Fixed	Fixed	Fixed	Fixed	Flexible
Support Multi-model Inference		No	No	No	No	Yes	Yes

tion by different number of hidden neurons and layers. The DNN model proved accurate and consistent over other ML models in software simulation. Our generated system has yielded an accuracy of 91.4% to 99.1% for multimodal signal classification. We have integrated these independent DNN models into single DNN IP cores with flexible topology. The adaptive DNN IP cores with 16-bit precision have obtained a balance performance compared to 8- and 24-bit precision. The analysis revealed that our developed adaptive DNN IP core with 16-bit requires 680 ns with 309 mW power for a single inference. The decentralization of intelligence of the DNN IP core has resulted in a data size reduction of 96.25% to 98.75%. This indirectly helped to reduce communication overload and data storage required in the cloud and improved the security and privacy during data transfer. Our developed adaptive DNN model is fast with an inference speed of DNN IP core of 1.47 mega samples per second. Thus, we can conclude that our proposed adaptive DNN IP core is flexible and efficient for critical healthcare EI applications. In addition, our presented DNN IP core can be easily integrated into other EI applications with minimal changes in hardware design. However, this study is limited by the absence of validation on real-world datasets, as all experiments were conducted on controlled or benchmark data. Additionally, the scalability of the proposed framework, particularly in terms of performance and resource requirements, has not yet been fully explored on larger and more heterogeneous datasets. Another limitation is that the addition of new data modalities could affect the current system design and its integration capabilities, potentially requiring architectural modifications or retraining. In future work, we aim to address these gaps by applying the method to extensive, real-world datasets from diverse domains, systematically analyzing its scalability, and evaluating the impact of incorporating additional data modalities to ensure robustness and adaptability in practical scenarios. In addition, our goal will be to explore pruning and fixed-precision training (such as QAT) during software simulation to improve the accuracy performance of DNN models. The same DNN IP core can be directly used in real-time ICU scenarios containing large numbers of patients and data modalities with just appropriate fine-tuning of the DNN IP core. In addition, the developed DNN IP core can be easily used for other EI applications with smaller topologies because of its flexible WB configuration.

REFERENCES

[1] M. A. Scrugli, P. Meloni, C. Sau, and L. Raffo, "Runtime adaptive iomt node on multi-core processor platform," Electronics, vol. 10, no. 21, p. 2572, 2021.

[2] S. K. Khare and U. R. Acharya, "An explainable and interpretable model for attention deficit hyperactivity disorder in children using EEG signals," Computers in Biology and Medicine, vol. 155, p. 106676, 2023.

[3] S. K. Khare, V. Bajaj, and U. R. Acharya, "SchizoNET: a robust and accurate margenau-hill time-frequency distribution based deep neural network model for schizophrenia detection using EEG signals," Physiological Measurement, 2023.

[4] N. B. Gaikwad, S. K. Khare, N. Satpute, and A. G. Keskar, "Hardware implementation of high-performance classifiers for edge gateway of smart automobile," in 2022 1st International Conference on the Paradigm Shifts in Communication, Embedded Systems, Machine Learning and Signal Processing (PCEMS), 2022, pp. 74–77.

[5] Q. Wang, X. Zhu, Y. Ni, L. Gu, and H. Zhu, "Blockchain for the iot and industrial iot: A review," Internet of Things, vol. 10, p. 100081, 2020.

[6] N. B. Gaikwad, H. Ugale, A. Keskar, and N. C. Shivaprakash, "The internet-of-battlefield-things (iobt)-based enemy localization using soldiers location and gunshot direction," IEEE Internet of Things Journal, vol. 7, no. 12, pp. 11 725–11 734, 2020.

[7] A. Ghubaish, T. Salman, M. Zolanvari, D. Unal, A. Al-Ali, and R. Jain, "Recent advances in the internet-of-medical-things (iomt) systems security," IEEE Internet of Things Journal, vol. 8, no. 11, pp. 8707–8718, 2020.

[8] S. Khare, A. Nishad, A. Upadhyay, and V. Bajaj, "Classification of emotions from EEG signals using time-order representation based on the S-transform and convolutional neural network," Electronics Letters, vol. 56, no. 25, pp. 1359–1361, 2020.

[9] S. K. Khare and V. Bajaj, "A self-learned decomposition and classification model for schizophrenia diagnosis," Computer Methods and Programs in Biomedicine, vol. 211, p. 106450, 2021.

[10] V. Jahmunah, E. Ng, R.-S. Tan, S. L. Oh, and U. R. Acharya, "Uncertainty quantification in densenet model using myocardial infarction ECG signals," Computer Methods and Programs in Biomedicine, vol. 229, p. 107308, 2023.

[11] H. W. Loh, S. Xu, O. Faust, C. P. Ooi, P. D. Barua, S. Chakraborty, R.-S. Tan, F. Molinari, and U. R. Acharya, "Application of photoplethysmography signals for healthcare systems: An in-depth review," Computer Methods and Programs in Biomedicine, vol. 216, p. 106677, 2022.

[12] M. Coskun, O. Yildirim, Y. Demir, and U. R. Acharya, "Efficient deep neural network model for classification of grasp types using sEMG signals," Journal of Ambient Intelligence and Humanized Computing, vol. 13, no. 9, pp. 4437–4450, 2022.

[13] B. S. Egala, S. Priyanka, and A. K. Pradhan, "Shpi: smart healthcare system for patients in icu using iot," in 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). IEEE, 2019, pp. 1–6.

[14] L. Sun, X. Jiang, H. Ren, and Y. Guo, "Edge-cloud computing and artificial intelligence in internet of medical things: architecture, technology and application," IEEE Access, vol. 8, pp. 101 079–101 092, 2020.

[15] A. Banerjee, B. K. Mohanta, S. S. Panda, D. Jena, and S. Sobhanayak, "A secure iot-fog enabled smart decision making system using machine learning for intensive care unit," in 2020 International Conference on Artificial Intelligence and Signal Processing (AISP), 2020, pp. 1–6.

[16] U. U. Tariq, H. Ali, L. Liu, J. Panneerselvam, and X. Zhai, "Energy-efficient static task scheduling on vfi-based noc-hmpsocs for intelligent edge devices in cyber-physical systems," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 10, no. 6, pp. 1–22, 2019.

[17] S. H. A. Shah, D. Koundal, V. Sai, and S. Rani, "Guest editorial: Special section on 5g edge computing-enabled internet of medical things," IEEE Transactions on Industrial Informatics, vol. 18, no. 12, pp. 8860–8863, 2022.

[18] S. U. Amin and M. S. Hossain, "Edge intelligence and internet of things in healthcare: A survey," IEEE Access, vol. 9, pp. 45–59, 2021.

[19] Z. Chang, S. Liu, X. Xiong, Z. Cai, and G. Tu, "A survey of recent advances in edge-computing-powered artificial intelligence of things," IEEE Internet of Things Journal, vol. 8, no. 18, pp. 13 849–13 875, 2021.

[20] C. Xu, S. Jiang, G. Luo, G. Sun, N. An, G. Huang, and X. Liu, "The case for fpga-based edge computing," IEEE Transactions on Mobile Computing, vol. 21, no. 7, pp. 2610–2619, 2020.

[21] N. B. Gaikwad, V. Tiwari, A. Keskar, and N. C. Shivaprakash, "Efficient fpga implementation of multilayer perceptron for real-time human activity classification," IEEE Access, vol. 7, pp. 26 696–26 706, 2019.

[22] C. Hao, X. Zhang, Y. Li, S. Huang, J. Xiong, K. Rupnow, W.-m. Hwu, and D. Chen, "Fpga/dnn co-design: An efficient design methodology for iot intelligence on the edge," in Proceedings of the 56th Annual Design Automation Conference 2019, 2019, pp. 1–6.

[23] C. Wisultschew, A. Pérez, A. Otero, G. Mujica, and J. Portilla, "Characterizing deep neural networks on edge computing systems for object classification in 3d point clouds," IEEE Sensors Journal, vol. 22, no. 17, pp. 17 075–17 089, 2022.

[24] C.-H. Huang, B.-W. Chen, Y.-J. Lin, and J.-X. Zheng, "Smart crop growth monitoring based on system adaptivity and edge ai," IEEE Access, vol. 10, pp. 64 114–64 125, 2022.

[25] B. Paul, T. K. Yadav, B. Singh, S. Krishnaswamy, and G. Trivedi, "A resource efficient software-hardware co-design of lattice-based homomorphic encryption scheme on the fpga," IEEE Transactions on Computers, vol. 72, no. 5, pp. 1247–1260, 2022.

[26] Y. Rebahi, F. Catal, N. Tcholtchev, L. Maedje, O. Alkhateeb, V. K. Elangovan, and D. Apostolakis, "Towards accelerating intrusion detection operations at the edge network using fpgas," in 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC). IEEE, 2020, pp. 104–111.

[27] J. Bartels, A. Hagihara, L. Minati, K. K. Tokgoz, and H. Ito, "An integer-only resource-minimized rnn on fpga for low-frequency sensors in edge-ai," IEEE Sensors Journal, 2023.

[28] N. B. Gaikwad, S. K. Khare, H. Ugale, D. Mendhe, V. Tiwari, V. Bajaj, and A. G. Keskar, "Hardware design and implementation of multi-agent mlp regression for the estimation of gunshot direction on iobt edge gateway," IEEE Sensors Journal, pp. 1–1, 2023.

[29] M. Alabdulhafith, H. Saleh, H. Elmannai, Z. H. Ali, S. El-Sappagh, J.-W. Hu, and N. El-Rashidy, "A clinical decision support system for edge/cloud icu readmission model based on particle swarm optimization, ensemble machine learning, and explainable artificial intelligence," IEEE Access, vol. 11, pp. 100 604–100 621, 2023.

[30] J. Lu, X. Chen, S. Li, X. Qian, A. Yuemaier, and Z. Song, "A novel nvm memory file system for edge intelligence," IEICE Electronics Express, vol. 19, no. 8, pp. 20 220 079–20 220 079, 2022.

[31] M. R. Al Koutayni, G. Reis, and D. Stricker, "Deepedgesoc: End-to-end deep learning framework for edge iot devices," Internet of Things, vol. 21, p. 100665, 2023.

[32] A. F. Ajirlou, F. Kenarangi, E. Shapira, and I. Partin-Vaisband, "Nod: A neural network-over-decoder for edge intelligence," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 30, no. 10, pp. 1438–1447, 2022.

[33] C. Gao, A. Rios-Navarro, X. Chen, S.-C. Liu, and T. Delbruck, "Edgedrnn: Recurrent neural network accelerator for edge inference," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 10, no. 4, pp. 419–432, 2020.

[34] C. F. Frasser, P. Linares-Serrano, I. D. de Los Rios, A. Moran, E. S. Skibinsky-Gitlin, J. Font-Rossello, V. Canals, M. Roca, T. Serrano-Gotarredona, and J. L. Rossello, "Fully parallel stochastic computing hardware implementation of convolutional neural networks for edge computing applications," IEEE Transactions on Neural Networks and Learning Systems, 2022.

[35] I. Ben Dhaou, M. Ebrahimi, M. Ben Ammar, G. Bouattour, and O. Kanoun, "Edge devices for internet of medical things: technologies, techniques, and implementation," Electronics, vol. 10, no. 17, p. 2104, 2021.

[36] H.-N. Dai, Y. Wu, H. Wang, M. Imran, and N. Haider, "Blockchain-empowered edge intelligence for internet of medical things against covid-19," IEEE Internet of Things Magazine, vol. 4, no. 2, pp. 34–39, 2021.

[37] K. Kakhi, R. Alizadehsani, H. D. Kabir, A. Khosravi, S. Nahavandi, and U. R. Acharya, "The internet of medical things and artificial intelligence: trends, challenges, and opportunities," Biocybernetics and Biomedical Engineering, 2022.

[38] R. Gupta, D. Reebadiya, and S. Tanwar, "6g-enabled edge intelligence for ultra-reliable low latency applications: Vision and mission," Computer Standards & Interfaces, vol. 77, p. 103521, 2021.

[39] N. B. Gaikwad, V. Tiwari, A. Keskar, and N. Shivaprakash, "Heterogeneous sensor data analysis using efficient adaptive artificial neural network on fpga based edge gateway," KSII Transactions on Internet and Information Systems (TIIS), vol. 13, no. 10, pp. 4865–4885, 2019.

[40] M. Ben Ammar, I. Ben Dhaou, D. El Houssaini, S. Sahnoun, A. Fakhfakh, and O. Kanoun, "Requirements for energy-harvesting-driven edge devices using task-offloading approaches," Electronics, vol. 11, no. 3, 2022.

[41] C. Gao, A. Rios-Navarro, X. Chen, S.-C. Liu, and T. Delbruck, "Edgedrnn: Recurrent neural network accelerator for edge inference," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 10, no. 4, pp. 419–432, 2020.

[42] "Vitis model composer [online] :"https://docs.xilinx.com/r/en-us/ug1483-model-composer-sys-gen-user-guide/overview"," AMD Xilinx, Tech. Rep., 2022. [Online]. Available: https://docs.xilinx.com/r/en-US/ug1483-model-composer-sys-gen-user-guide/Overview

[43] "Kria kv260 ai starter kit [online] :"https://docs.xilinx.com/r/en-us/ug1089-kv260-starter-kit"," AMD Xilinx, Tech. Rep., 2022. [Online]. Available: https://docs.xilinx.com/r/en-US/ug1089-kv260-starter-kit

[44] M. R. Azghadi, C. Lammie, J. K. Eshraghian, M. Payvand, E. Donati, B. Linares-Barranco, and G. Indiveri, "Hardware implementation of deep network accelerators towards healthcare and biomedical applications," IEEE Transactions on Biomedical Circuits and Systems, vol. 14, no. 6, pp. 1138–1159, 2020.

[45] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7457–7469, 2020.

[46] A. Tisan and J. Chin, "An end-user platform for fpga-based design and rapid prototyping of feedforward artificial neural networks with on-chip backpropagation learning," IEEE Transactions on Industrial Informatics, vol. 12, no. 3, pp. 1124–1133, 2016.

[47] E. B. Panganiban, A. C. Paglinawan, W. Y. Chung, and G. L. S. Paa, "Ecg diagnostic support system (edss): A deep learning neural network based classification system for detecting ecg abnormal rhythms from a low-powered wearable biosensors," Sensing and Bio-Sensing Research, vol. 31, p. 100398, 2021.

[48] H. Karnan, N. Sivakumaran, and R. Manivel, "An efficient cardiac arrhythmia onset detection technique using a novel feature rank score algorithm," Journal of medical systems, vol. 43, pp. 1–8, 2019.

[49] V. G. Rangappa, S. Prasad, and A. Agarwal, "Classification of cardiac arrhythmia stages using hybrid features extraction with k-nearest neighbour classifier of ecg signals," learning, vol. 11, pp. 21–32, 2018.

[50] N. Raghu, "Arrhythmia detection based on hybrid features of t-wave in electrocardiogram," in Deep Learning Techniques and Optimization Strategies in Big Data Analytics. IGI Global, 2020, pp. 1–20.

[51] M. Kachuee, M. M. Kiani, H. Mohammadzade, and M. Shabany, "Cuffless blood pressure estimation algorithms for continuous health-care monitoring," IEEE Transactions on Biomedical Engineering, vol. 64, no. 4, pp. 859–869, 2016.

[52] H. Tjahjadi and K. Ramli, "Noninvasive blood pressure classification based on photoplethysmography using k-nearest neighbors algorithm: a feasibility study," Information, vol. 11, no. 2, p. 93, 2020.

[53] Y. Liang, Z. Chen, R. Ward, and M. Elgendi, "Hypertension assessment via ecg and ppg signals: An evaluation using mimic database," Diagnostics, vol. 8, no. 3, p. 65, 2018.

[54] I. Kuzmanov, A. M. Bogdanova, M. Kostoska, and N. Ackovska, "Fast cuffless blood pressure classification with ecg and ppg signals using cnn-lstm models in emergency medicine," in 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO). IEEE, 2022, pp. 362–367.

[55] I. Kuzmanov, M. Kostoska, and A. Madevska Bogdanova, "Blood pressure class estimation using cnn-gru model," 2022.

[56] X. Sun, L. Zhou, S. Chang, and Z. Liu, "Using cnn and hht to predict blood pressure level based on photoplethysmography and its derivatives," Biosensors, vol. 11, no. 4, p. 120, 2021.

[57] Y. Chen and C. Shen, "Performance analysis of smartphone-sensor behavior for human activity recognition," Ieee Access, vol. 5, pp. 3095–3110, 2017.

[58] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," ACM SigKDD Explorations Newsletter, vol. 12, no. 2, pp. 74–82, 2011.

[59] S. Chernbumroong, A. S. Atkins, and H. Yu, "Activity classification using a single wrist-worn accelerometer," in 2011 5th international conference on software, knowledge information, industrial management and applications (SKIMA) proceedings. IEEE, 2011, pp. 1–6.

[60] R. Ahmed Bhuiyan, N. Ahmed, M. Amiruzzaman, and M. R. Islam, "A robust feature extraction model for human activity characterization using 3-axis accelerometer and gyroscope data," Sensors, vol. 20, no. 23, p. 6990, 2020.

[61] K. Chen, L. Yao, D. Zhang, X. Wang, X. Chang, and F. Nie, "A semisupervised recurrent convolutional attention model for human activity recognition," IEEE transactions on neural networks and learning systems, vol. 31, no. 5, pp. 1747–1756, 2019.

[62] S. Jana and S. K. Shome, "Hybrid ensemble based machine learning for smart building fire detection using multi modal sensor data," Fire Technology, vol. 59, no. 2, pp. 473–496, 2023.

[63] A. I. Ahmad, R. Mustapha, and K. Haruna, "A framework for predicting household fire using internet of things (iot) technology."

[64] L. An, L. Chen, and X. Hao, "Indoor fire detection algorithm based on second-order exponential smoothing and information fusion," Information, vol. 14, no. 5, p. 258, 2023.

[65] C. Nagolu, C. Cheekula, D. S. K. Thota, K. Padmanaban, and D. Bhattacharyya, "Real-time forest fire detection using iot and smart sensors," in 2023 International Conference on Inventive Computation Technologies (ICICT). IEEE, 2023, pp. 1441–1447.

[66] S. Blattmann, "Real-time smoke detection with ai-based sensor fusion [online] "https://github.com/blatts01/ai-smokedetector"," 2022. [Online]. Available: https://github.com/Blatts01/ai-smokedetector

[67] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016, http://www.deeplearningbook.org.

[68] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural networks, vol. 61, pp. 85–117, 2015.

[69] K. Basterretxea, J. Echanobe, and I. del Campo, "A wearable human activity recognition system on a chip," in Proceedings of the 2014 Conference on Design and Architectures for Signal and Image Processing. IEEE, 2014, pp. 1–8.

[70] M. Dendaluce Jahnke, F. Cosco, R. Novickis, J. Perez Rastelli, and V. Gomez-Garay, "Efficient neural network implementations on parallel embedded platforms applied to real-time torque-vectoring optimization using predictions for multi-motor electric vehicles," Electronics, vol. 8, no. 2, p. 250, 2019.

[71] J. G. Oliveira, R. L. Moreno, O. de Oliveira Dutra, and T. C. Pimenta, "Implementation of a reconfigurable neural network in fpga," in 2017 International Caribbean Conference on Devices, Circuits and Systems (ICCDCS). IEEE, 2017, pp. 41–44.

[72] S. K. Khare, N. B. Gaikwad, and V. Bajaj, "Vhers: A novel variational mode decomposition and hilbert transform-based eeg rhythm separation for automatic adhd detection," IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1–10, 2022.

[73] G. Thambiraj, U. Gandhi, U. Mangalanathan, V. J. M. Jose, and M. Anand, "Investigation on the effect of womersley number, ecg and ppg features for cuff less blood pressure estimation using machine learning," Biomedical Signal Processing and Control, vol. 60, p. 101942, 2020.

[74] N. B. Gaikwad, A. G. Keskar, V. Tiwari, and N. Shivaprakash, "Fpga implementation of real-time soldier activity detection based on neural network classifier in smart military suit," in 2019 IEEE Bombay Section Signature Conference (IBSSC). IEEE, 2019, pp. 1–6.

[75] R. Tomar and R. Tiwari, "Information delivery system for early forest fire detection using internet of things," in Advances in Computing and Data Sciences: Third International Conference, ICACDS 2019, Ghaziabad, India, April 12–13, 2019, Revised Selected Papers, Part I 3. Springer, 2019, pp. 477–486.

[76] G. L. H. J. I. P. M. R. M. J. M. G. P. C. S. H. Goldberger AL, Amaral LA, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals." Circulation, vol. 101(23):E215-20, 2000 Jun 13.

[77] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," Neurocomputing, vol. 171, pp. 754–767, 2016.

[78] E. Soufleri and K. Roy, "Network compression via mixed precision quantization using a multi-layer perceptron for the bit-width allocation," IEEE Access, vol. 9, pp. 135 059–135 068, 2021.

[79] M. Pistellato, F. Bergamasco, G. Bigaglia, A. Gasparetto, A. Albarelli, M. Boschetti, and R. Passerone, "Quantization-aware nn layers with high-throughput fpga implementation for edge ai," Sensors, vol. 23, no. 10, p. 4667, 2023.

[80] J. Yang, S. Hong, and J.-Y. Kim, "Fixar: A fixed-point deep reinforcement learning platform with quantization-aware training and adaptive parallelism," in 2021 58th ACM/IEEE Design Automation Conference (DAC). IEEE, 2021, pp. 259–264.

[81] M. Kirtas, N. Passalis, A. Oikonomou, M. Moralis-Pegios, G. Giamougiannis, A. Tsakyridis, G. Mourgias-Alexandris, N. Pleros, and A. Tefas, "Mixed-precision quantization-aware training for photonic neural networks," Neural Computing and Applications, vol. 35, no. 29, pp. 21 361–21 379, 2023.

[82] T. Allenet, D. Briand, O. Bichler, and O. Sentieys, "Disentangled loss for low-bit quantization-aware training," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 2788–2792.

[83] S. T. Muhammad, M. A. El-Moursy, A. A. El-Moursy, and H. F. Hamed, "Architecture level analysis for process variation in synchronous and asynchronous networks-on-chip," Journal of Parallel and Distributed Computing, vol. 102, pp. 175–185, 2017.

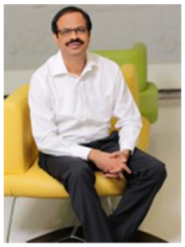
NIKHIL B. GAIKWAD was born in 1989. He received his B.E. degree in Electronics and Communication Engineering from Nagpur University, India, in 2012, and his M.Tech. degree from VJTI, Mumbai, in 2015. He completed his Ph.D. at Visvesvaraya National Institute of Technology (VNIT), Nagpur, in 2021, under the supervision of Dr. A. Keskar. He is currently a Postdoctoral Researcher at Aalborg University, Denmark. His research interests include edge intelligence, FPGA/SoC and GPU-based AI acceleration, smart sensors, AI hardware, distributed systems, the Internet of Things (IoT), and heterogeneous computing.

SMITH K. KHARE is an Assistant Professor in the SDU Applied AI and Data Science, The Maersk Mc-Kinney Møller Institute, University of Southern Denmark, Denmark, and worked as a Postdoctoral researcher in the Aarhus University, Denmark. He received his doctoral degree in Electronics and Communication Engineering at the Indian Institute of Information Technology, Design and Manufacturing Jabalpur (IIITDMJ), India in 2022. He has authored more than 50+ research papers in various reputed international Journals such as IEEE Transactions. Smith is listed in the top 2% Scientists in the World (2023, 2024), according to Elsevier.

DINESH MENDHE is a computer and information research scientist and entrepreneur who loves technology and research. He founded Modern Softwares LLC, a company that offers cutting-edge software solutions for various domains. He also leads the software development effort for Institute for Health's research grants, where he develops secure and innovative research and statistical applications, algorithms, and tools for healthcare research. In addition, he has a wealth of experience in handling complex datasets such as Medicaid, Medicare, SEER, and SPARCS. His expertise spans AI/ML, genomics, bioinformatics, precision medicine, big data algorithms and frameworks, data structures, automated AI/ML genomics pipelines, mobile apps, sensor and IoT devices, sensor data classification, high-performance computing, and HIPAA-compliant data centers.

HASAN MIR (Senior Member, IEEE) received the B.S. (cum laude), M.S., and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, WA, USA, in 2000, 2001, and 2005, respectively. From 2005 to 2009, he was with the Air Defense Technology Group, Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA, USA. Since 2009, he has been with the Department of Electrical Engineering, American University of Sharjah, Sharjah, United Arab Emirates, where he is currently a Professor.

SOKOL KOSTA (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the Sapienza University of Rome, Rome, Italy, in 2006, 2009, and 2013, respectively. He was a Postdoctoral Researcher of Computer Science with the Sapienza University of Rome and a Visiting Researcher with The Hong Kong University of Science and Technology, Hong Kong, in 2015. He is an Associate Professor with the Department of Electronic Systems, Aalborg University, Aalborg, Denmark. His research includes networking, distributed systems, security, and edge computing. Dr. Kosta has won the Best Ph.D. Student Paper Award by the Computer Science Department of Sapienza University in 2012, the IEEE INFOCOM and IEEE SECON Best Demo Awards in 2013, and the IEEE INFOCOM Test of Time Paper Award in 2024.



U. RAJENDRA ACHARYA FIAPR and FAAIA, is a Professor of Artificial Intelligence in Healthcare at the School of Mathematics, Physics, and Computing at the University of Southern Queensland, Australia. His research interests include biomedical imaging and signal processing, data mining, and visualization, as well as applications of biophysics for better healthcare design and delivery. His funded research has accrued cumulative grants exceeding six million Singapore dollars. He has authored over 800 publications, including 750 in refereed international journals, 42 in international conference proceedings, and 17 books. He has received over 95,000 citations on Google Scholar (with an h-index of 154). According to the Essential Science Indicators of Thomson, he has been ranked in the top 1% of the highly cited researchers for the last seven consecutive years (2016–2022) in computer science.