Heterogeneous Sensor Data Analysis Using Efficient Adaptive Artificial Neural Network on FPGA Based Edge Gateway

Nikhil B. Gaikwad[1], Varun Tiwari[1], Avinash Keskar[1] and NC Shivaprakash[2]
[1] Department of Electronics and Communication Engineering, Visvesvaraya National Institute of Technology (VNIT), South Ambazari Road, Nagpur, India, 40010
[e-mail : nikhilgaikwad9423@gmail.com, varun.etrx@gmail.com, agkeskar@vnit.ac.in]
[2] Department of Instrumentation and Applied Physics, Indian Institute of Science (IISc), C V Raman Ave, Bengaluru, India, 560012
[e-mail : shiv@iisc.ac.in]
*Corresponding author : Nikhil B. Gaikwad

Received December 2, 2018; revised February 16, 2019; revised April 17, 2019; accepted May 1, 2019; published October 31, 2019

Abstract

We propose a FPGA based design that performs real-time power-efficient analysis of heterogeneous sensor data using adaptive ANN on edge gateway of smart military wearables. In this work, four independent ANN classifiers are developed with optimum topologies. Out of which human activity, BP and toxic gas classifier are multiclass and ECG classifier is binary. These classifiers are later integrated into a single adaptive ANN hardware with a select line(s) that switches the hardware architecture as per the sensor type. Five versions of adaptive ANN with different precisions have been synthesized into IP cores. These IP cores are implemented and tested on Xilinx Artix-7 FPGA using Microblaze test system and LabVIEW based sensor simulators. The hardware analysis shows that the adaptive ANN even with 8-bit precision is the most efficient IP core in terms of hardware resource utilization and power consumption without compromising much on classification accuracy. This IP core requires only 31 microseconds for classification by consuming only 12 milliwatts of power. The proposed adaptive ANN design saves 61% to 97% of different FPGA resources and 44% of power as compared with the independent implementations. In addition, 96.87% to 98.75% of data throughput reduction is achieved by this edge gateway.

Keywords: Real-time data analysis, field programmable gate array, adaptive artificial neural network, edge gateway, fog computing, smart wearables

1. Introduction

The wireless sensor networks have emerged as a prominent technology for many applications in the last two decades [1]. During these years, many important wireless sensor network concepts and algorithms are implemented and tested in the various application domains [2] [3]. The next generation enhancement in these networks has started in the direction of the internet of things (IoT). The transformation of traditional sensor networks into the smart environments is conceivable due to the rising IoT framework. The total internet-connected devices are rapidly increasing in recent years. According to the predictions, around 50 billion IoT devices will be connected to the internet across the globe by 2020 [4]. The increasing connections impose several limitations on the network performance like a real-time response, network security, reliability, energy-efficiency, interoperability, etc [5]. The decentralization of computing, networking and storage resources appear as an effective solution for these limitations [6]. The new emerging paradigm based on this approach is popularly known as fog computing or edge computing in the IoT domain. The fog computing technology is designed to achieve low network latency, high network security and efficient bandwidth utilization in IoT applications [7].

The fog computing operates in the fog layer present between the cloud layer and the device layer in the IoT infrastructure [6]. In the fog layer, the edge devices play a key role to ensure the real-time services and secure networking. The edge devices are equipped with sufficient computational power, which results in a considerable reduction in response latency and network traffic [8]. The gateway device is widely considered as the important edge platform for the implementation and execution of the fog layer that handles an essential local processing and services [9]. Traditionally, the gateway works as a protocol converter, routing-forwarding device and communication bridge point between network and the sensor [10]. The gateways are preferred because of the sufficient availability of computing resources, communication bandwidth, power consumption and it holds a strategic location in the network [5].

The implementation of the fog computing approach on the defense IoT application is presented in this work. The defense wearable gateway has been selected as an edge platform in this scenario. Traditionally, the wireless sensor networks in the military domain have to meet stringent system requirements like low latency and low power data processing, network security, real-time and reliable intelligence, low bandwidth availability, mobility, etc. However, these requirements become more challenging in case of difficult combat scenarios [11] [12]. These challenges can be potentially fulfilled by the adoption of fog computing approach in defense IoT domain.

In the proposed work, a real-time and intelligent data analyzer has been implemented for smart wearable edge gateway. Intellectual Property (IP) core of the adaptive artificial neural network (ANN) is developed for detection of soldier's physical activity, physiological status and chemical warfare threat. The adaptive ANN IP core has been tested on the heterogeneous hardware and then it is integrated seamlessly into the FPGA/SoC based edge gateway. The IP core is capable to classify four types of heterogeneous sensor features collected on the edge gateway in real-time. In this work, the selected heterogeneous sensor nodes are an accelerometer for current soldier activity detection, ECG for the abnormal cardiac activity detection, PPG for abnormal blood pressure detection and a gas sensor for toxic gas detection. The extensive analysis of each development stages, comparison with conventional designs and performance evaluation of edge gateway has been presented in this paper.

The selection of appropriate hardware platform plays a crucial role in the efficient working of edge gateway. The hardware architecture must be capable of parallel processing and dynamic reconfiguration with the low power requirements to ensure real-time performance [13]. The recent FPGA/SoCs provides high-performance to power ratio among all available parallel processing devices [14]. The new generation FPGA/SoCs are becoming a popular choice for the implementation of edge gateways [15] [16]. We used Xilinx Artix-7 FPGA for the IP designing and testing because all cost-optimized Xilinx SoC devices use Artix-7 FPGA fabric as Programmable Logic (PL). This also enables our adaptive ANN IP core to be seamlessly integrated into any cost-optimized Xilinx SoC Platforms [17].

The remainder of the research paper is organized as follows. In Section 2, we review the design and implementation of edge gateways for the various functionalities in the past few years, also we discuss the motivation behind this work. In Section 3, we focus on the modeling of adaptive artificial neural network IP Core for the edge gateway. The Section 3 is divided into two subsections, the first subsection shows the mathematical model of basic ANN design and second subsection is dedicated to mathematical modeling of adaptive ANN design. In Section 4, we describe the complete implementation and testing setup for the adaptive ANN IP core. Section 5 discusses the implementation and performance analysis of an ANN IP core. Finally, Section 6 concludes this work with important research findings and suggests some future direction.

2. Related Work

Among all edge platforms, the gateways are considered as the most important edge devices in fog computing paradigm. Hence, this attracted many researchers to investigate efficient edge gateway architecture in the last three to four years. The role of edge gateways in various IoT applications includes data dissemination, real-time services, decentralized computation, transient storage, security and privacy [6]. Some of the previously reported edge gateway architectures and implementations have been discussed below, which were developed for various fog assisted IoT applications.

In order to enhance the performance of an IoT health monitoring system, Amir M. Rahmani et al. [5] proposed a smart e-Health gateway (UT-GATE). This work demonstrated local data processing based on fog computing approach for monitoring patient with acute illness. In the Industrial IoT (IIoT) domain, Ching-Han Chen et al. [18] designed a real-time and low power edge gateway for the data collection, communication and field-bus management. The modeling of edge gateway is done on FPGA by implementing the concept of multiple collaborative microcontrollers. The gateway also has to handle the internet communication and sensor data exchange, so the optimum resource management is a prime concern in its design; Roberto Morabito et al. addressed this in their work [10]. Their lightweight edge gateway (LEGIoT) achieved first service allocation, interoperability, high-energy efficiency and flexibility in the management of applications. The LEGIoT was implemented on the RPi2, RPi3, OC1+, OC2 hardware platforms and the performance were evaluated using 25 and 100 nodes.

Surabhi Abhimithra Karthikeya et al. [19] proposed the low-cost and power efficient solutions specific gateways (SSGWs). The SSGWs are specially designed for the remote area deployments, where devices are battery powered. Manuel Suárez-Albela et al. [9] discussed the energy efficient gateway. Nandor Verba et al. [20] proposed Raspberry Pi based open Service Gateway Interface (OSGI), which performs peripheral communication abstraction and

clustering of gateways. One of the interesting work by Blesson Varghese et al. [21] shows hidden potentials of fog computing. In the online game use case, the preliminary results of this work achieved 20% improvement in average response time and 90% reduction in cloud data traffic.

The various requirements of fog computing gateways have been achieved substantially in the above research works. We realized the need for low power and low latency local data analysis for the time-sensitive IoT applications like industry, transport and defense. The real-time data analysis is an important component of an edge gateway to detect emergency conditions, which also improves the system sensitivity, consistency, reliability and capability [5]. In addition, it increases the edge gateway performance and reduces the data throughput in time-sensitive applications [22]. In the defense IoT scenario, these parameters play the deciding role in the network-centric warfare circumstances. Therefore, this motivates us to implement real-time data analysis on the edge gateway in the defense IoT wearable [12].

To demonstrate real-time data analysis, we used four types of heterogeneous sensors data that transmits respective data features to the edge gateway. The important analysis is obtained from these features using the classification algorithm running on the edge gateway. The artificial neural network (ANN) is popularly used as a classification algorithm for all targeted sensor data analysis [23] [24] [25]. Therefore, the presented work has adopted the ANN as a classification algorithm. Thanks to ANN for its flexible architecture, we have successfully implemented a single ANN design, which is capable to classify all four types of sensor data using ANN adaptation concept. The adaptive ANN design gets configure into the respective type of classifier in real time.

3. Heterogeneous Sensors Data Classification

The basic details of ANN classifier is described in the first subsection and the second subsection is dedicated to the proposed adaptive ANN architecture. The adaptive ANN classifier is capable of analyzing diverse sensor data using limited configuration efforts.

3.1 Artificial Neural Network (ANN) Classifier

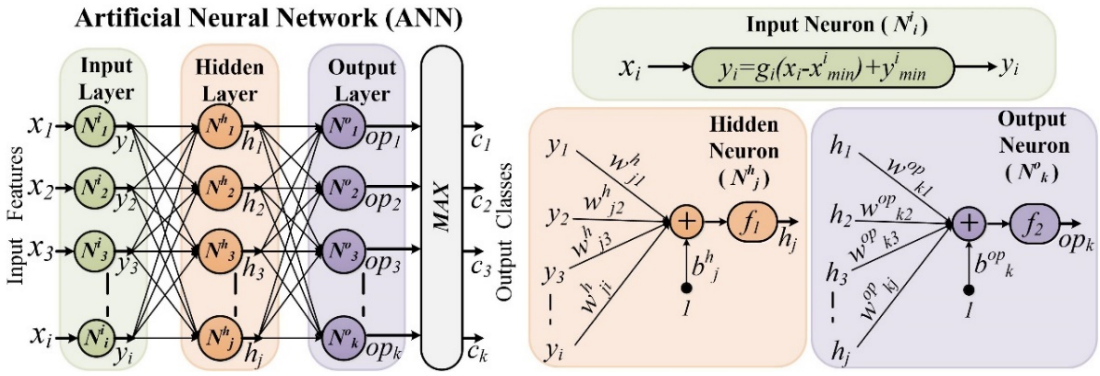

Fig. 1. Artificial Neural Network (ANN) with a single hidden layer.

The fully connected feed-forward multilayer perceptron with a hidden layer is used as an ANN design in this work. As shown in **Fig. 1**, an ANN design is divided into three layers. The output of each layer is calculated by the following equations.

- **Input Layer:**
  The input features are converted into a numeric range from 1 to -1 by the input layer, which is evaluated by Equation (1).

$$\boldsymbol{Y} = \{\boldsymbol{G} \circ (\boldsymbol{X} - \boldsymbol{X_{min}})\} + \boldsymbol{Y_{min}} \quad (1)$$

  Where, $\boldsymbol{Y}$ is the output matrix of the input layer, $\boldsymbol{X}$ is the input feature matrix; $\boldsymbol{X_{min}}$ is the matrix of all minimum values of respective features and it is subtracted from the $\boldsymbol{X}$ matrix. The resultant matrix is Hadamard Product ($\circ$) with the gain matrix $\boldsymbol{G}$. $\boldsymbol{Y_{min}}$ is the matrix of minimum values of output, as mentioned above output range is 1 to -1 so each element of $\boldsymbol{Y_{min}}$ is -1. The dimension of all these matrixes are i×1, where "i" is the total number of input features. The matrix $\boldsymbol{G}$ is calculated from the Equation (2).

$$\boldsymbol{G} = \left[\frac{2}{x^1_{max} - x^1_{min}} \quad \frac{2}{x^2_{max} - x^2_{min}} \quad \dots \dots \dots \quad \frac{2}{x^i_{max} - x^i_{min}}\right] \quad (2)$$

  Where, $x^1_{max}, x^1_{min}, x^2_{max}, x^2_{min} \dots\dots x^i_{max}, x^i_{min}$ are the maximum and minimum values of respective features. All of these constant matrices are evaluated during the training phase of the ANN classifiers.

- **Hidden Layer:**
  Total "j" neurons are present in the hidden layer; the output matrix of the hidden layer is calculated by the Equation (3).

$$\boldsymbol{H} = \boldsymbol{f_1}\{\boldsymbol{W^h} \times \boldsymbol{Y}\} + \boldsymbol{B^h} \quad (3)$$

  Where, $\boldsymbol{W^h}$ is the hidden layer weight matrix of dimension $j \times i$, $\boldsymbol{Y}$ is the output matrix of an input layer with the dimension of $i \times 1$, $\boldsymbol{B^h}$ is the bias matrix of a hidden layer with dimension $j \times 1$, and $\boldsymbol{f_1}$ works as activation function (nonlinear sigmoid). The hardware modeling of a perfect nonlinear function required enormous FPGA hardware resources. We implemented an efficient sigmoidal approximation known as a PLAN function to minimize the hardware resource utilization [26]. The Equation (4) shows the sigmoid function approximation (i.e. PLAN function).

$$\begin{aligned} PLAN\ (\beta) \quad &= 0.25 \times |\beta| + 0.5 && for\ 0 \le |\beta < 1, \\ &= 0.0125 \times |\beta| + 0.625 && for\ 1 \le |\beta| < 2.375, \\ &= 0.03125 \times |\beta| + 0.84375 && for\ 2.375 \le |\beta| \\ &= 1 && for\ 5 \le |\ \beta\ | \end{aligned} \quad (4)$$

- **Output Layer:**
  The basic ANN model classifies input features into the k[th] classes; therefore, the "k" total output neurons are present in the output layer. The final output matrix $\boldsymbol{O_P}$ is calculated from the Equation (5).

$$\boldsymbol{O_P} = \boldsymbol{f_2}\{\boldsymbol{W^{op}} \times \boldsymbol{H}\} + \boldsymbol{B^{op}} \quad (5)$$

  Where, $\boldsymbol{W^{op}}$ is weight matrix of output layer with dimension $k \times j$, $\boldsymbol{H}$ is the output matrix of a hidden layer with the dimension of $j \times 1$; $\boldsymbol{B^{op}}$ is the bias matrix of out layer with dimension of $k \times 1$. The function $\boldsymbol{f_2}$ is an activation function with the pure linear output response.

Generally, the ANN model is terminated with SOFTMAX function, this function is used for the calculation of error during ANN training [27]. However, it works as a regular MAX function in ANN testing. As this work is focused on the testing phase of ANN, we used MAX function to avoid the excess hardware resource utilization. After the evaluation of output matrix $\boldsymbol{O_P}$, the MAX block finds out an index of the maximum value in the output matrix. The calculated index is the output class of the ANN model.

As shown in **Fig. 1**, input parameters, weights and biases are calculated during the training of the ANN model. If the model is trained with a specific type of sensor features then it can only classify this respective sensor features. The classification of multiple heterogeneous sensor inputs is possible by developing separate and independent classifiers for all types of sensors. However, the hardware deployment of separate classifiers requires more hardware resources in a gateway and it increases linearly with number of heterogeneous sensors nodes. Indirectly, the excess hardware resource utilization also contributes to power consumption and cost of the gateway. Therefore, we designed an adaptive ANN model to minimize hardware resource utilization, power consumption and system cost of the gateway. The proposed adaptive ANN model is capable to classify all type of sensor features collected by the edge gateway. The adaptive ANN design gets configure according to the sensor type and classifies the respective features in real-time.

3.2 Adaptive ANN Classifier Architecture

Consider the scenario where the edge gateway is serving a time-sensitive application, which collects sets of features from “n” types of heterogeneous sensor nodes. The fog computing approach has been adopted for real-time data analysis to achieve quick gateway response. The important analysis is evaluated from the locally collected sensor features on the edge gateway itself. The implementation of separate pre-trained ANN classifiers for respective sensor analytics on a SoC/FPGA is one of the simplest possible approaches. However, it holds many drawbacks as explained above. Therefore, we proposed a novel adaptive ANN classifier implementation for the real-time data analysis in edge gateway.

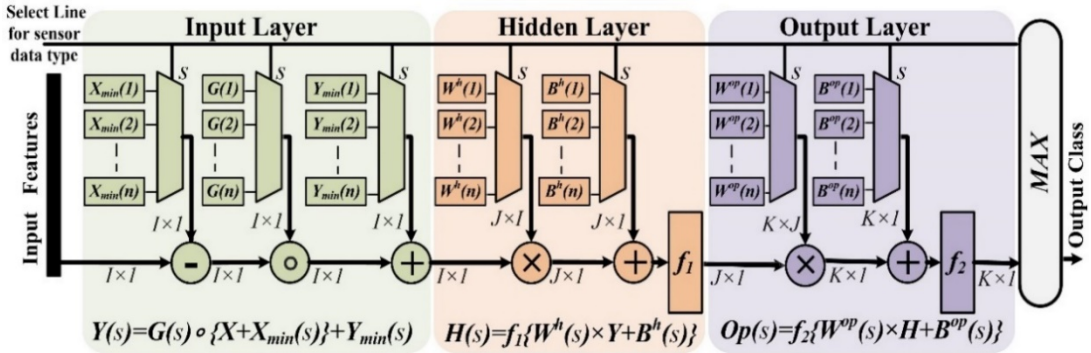

Fig. 2. The mathematical model of adaptive Artificial Neural Network (ANN) for heterogeneous sensor data classification.

As shown in **Fig. 1**, the observation of basic ANN architecture indicates that weight matrices, bias matrices and input layer constants define ANN classifiers functionality. Therefore, we designed the ANN model, which switches the respective ANN parameters according to the sensor type [28]. As shown in **Fig. 2**, the ANN parameters i.e. weights, biases etc. are supplied through the multiplexer to the ANN computational blocks. The selection of parameter matrix is done on basis of "*s*" select line input, which describes the sensor features type. The input features from sensors are heterogeneous and independent, therefore all pre-trained ANN classifiers are completely different from each other. The weight, bias and input layer constants matrices of all classifiers are calculated separately using the backpropagation algorithm during the training. The maximum dimensions among all the ANN classifiers have been selected as the dimension of the adaptive ANN so that it can accommodate all classifiers. The final topological dimension of an adaptive ANN are as

following, the input feature vector length (I), the number of hidden layer neurons (J) and the number of output layer neurons (K). The dimension of I, J and K are calculated from the following Equation (6). They are nothing but the maximum number of input, hidden and output neurons among all "n" topologies.

$$I = \max(i_1, i_2, i_3, \ldots. \ i_n),\ J = \max(j_1 j_2 j_3, \ldots. \ j_n)\ \&\ I = \max(k_1, k_2, k_3, \ldots. \ k_n) \quad (6)$$

Where, $i_1, i_2, i_3, \ldots. \ i_n$ are selected input feature vector lengths of all classifiers, $j_1 j_2 j_3, \ldots. \ j_n$ are all numbers of hidden layer neurons of "n" ANN classifiers, which is independently trained from "n" types of heterogeneous sensors features. Similarly, $k_1, k_2, k_3, \ldots. \ k_n$ are numbers of output layer neurons of all "n" ANN classifiers.

The independent training of all ANN models for the respective sensors generates distinct ANN topologies. Because of its diverse dimensions of all models, the number of input, hidden and output neurons are different from each other. However, the designed adaptive ANN model has a fixed dimension so that it can accommodate all models in a single design. In the adaptive ANN design, all parameter matrix of each classifier is upended with zeros so that it achieves the dimension of adaptive ANN (I, J and K). After parameter matrix resizing of all classes, these are integrated into adaptive ANN seamlessly. The modeling methodology of the adaptive ANN implementation is discussed below.

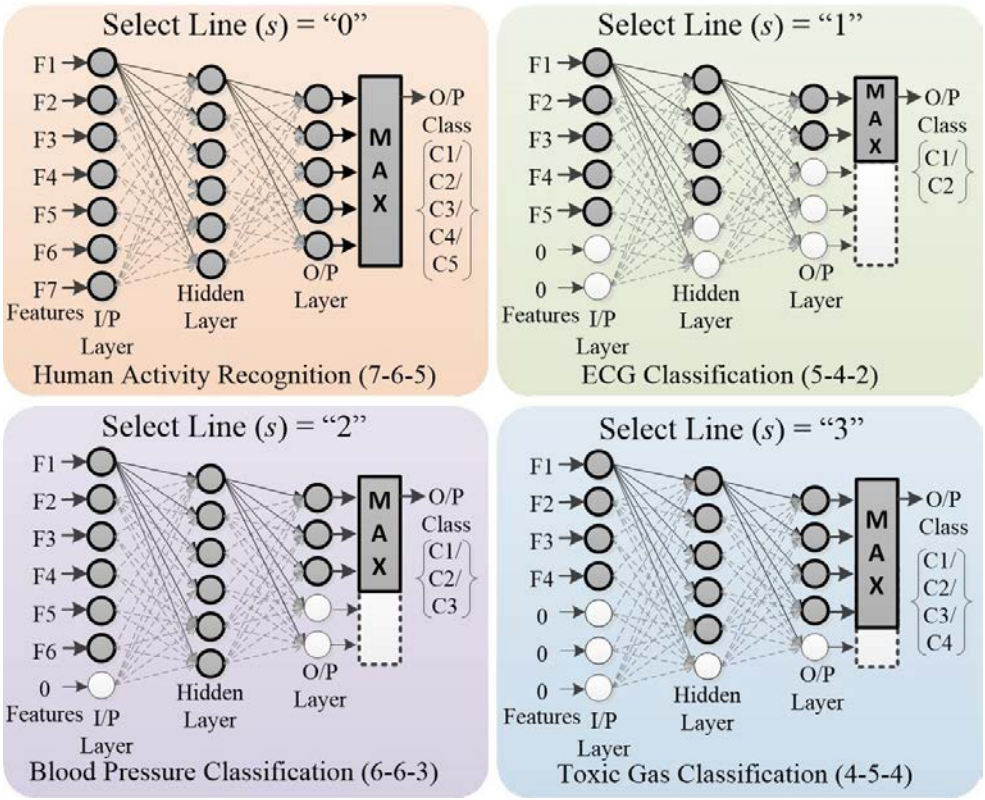

Fig. 3. The change in adaptive ANN architecture with respective select line status (s).

We selected four type of heterogeneous sensor nodes to evaluate the adaptive ANN performance in wearable edge gateway. The four independent ANN classifiers are trained from their respective data sets. Optimum ANN topologies with parameters (weights and biases) are obtained for all independent ANN models. Finally, all ANN models have been integrated into a single adaptive ANN model, which performs sequential classification according to the

select line(s) status. We selected four types of sensors, so "n" becomes 4 and select line(s) can be set to 0, 1, 2 or 3 which defines the type of classifications. **Fig. 3** shows the adaptive ANN architecture for all type of select line(s) inputs. At "$s = 0$", adaptive ANN classifier works as a human activity recognizer (7-6-5) with full connections. At "$s = 1$", adaptive ANN classifier is configured into ECG classifier (5-4-2) that uses five neurons input layer, four neurons in the hidden layer and two neurons at output layer. The neuron parameters (weight and biases and input layer constants) are switched according to the select line status so that it can perform ECG classification accurately. The remaining neurons are deactivated and the MAX function only accepts two neurons outputs during ECG classification. The similar approach works in case of blood pressure classification ($s = 2$) and toxic gas classification($s = 3$). As shown in **Fig. 3**, an adaptive ANN model is adapted according to the select line status. The same input and output connections of the model are used in all types of classifications. The following section describes each phase involved in the hardware implementation of adaptive ANN design.

4. Adaptive ANN Implementation and Testing

The four heterogeneous sensor ($n = 4$) inputs are classified by the adaptive ANN on edge gateway of defense wearables. All four-sensor nodes have been installed on the soldier wearables that generate raw heterogeneous sensors data. The respective sensor nodes perform preprocessing and feature extraction on raw data. The important and relevant classifications are done on the edge gateway by using hardware-based adaptive ANN in real time. The present work is focused on the development of adaptive ANN Intellectual Property (IP) core, which is integrated into the SoC/FPGA based wearable edge gateway.

4.1 Heterogeneous Sensor Data Collection

Table 1. Details of all four types of heterogeneous sensor datasets used in this work.

	Human Activity Recognition	**ECG Classification**	**BP Classification**	**Toxic Gas Classification**
Sensors	3-axial Accelerometer	Modified Limb Lead II (MLII)	PPG and ECG Sensors	Metal Oxide(MOX) Gas Sensors
Dataset	UCI Human Activities Data Set [29]	MIT-BIH Arrhythmia [30]	Cuff-Less Blood Pressure Estimation [30] [32]	UCI Gas Sensor Array Dataset [31]
Records Used	1,3,5,6,7,8,11,14, 15,16,17,19,21,2 3,25,26,27,28,29, 30	100,101,103,106, 112,113,109,114, 118,124,223,232, 234	29,30,35,71,77,79,2 97,390,383,436,439, 441,443,445,114,18 1,182,211,259,352,3 80,389,477,486,495	Channel CH2,CH3, CH5,CH7 of all 10 Batch's
Subjects	20	13	25	10 Batches
Sampling Rate	50 Hz	360 Hz	125 Hz	100 Hz
Window Size	0.56 sec	One ECG Interval	One ECG Interval	10 Milliseconds
Total Feature Vectors	7767	898	2629	20385
Training and Validation	7467	589	2329	20085
Testing	300	300	300	300

Four relevant machine-learning data sets of respective sensors have been used in this work for the development and testing of adaptive ANN model. **Table 1** shows detailed information about each dataset used for the respective sensor data analysis. The raw data streams are divided into small segments, the size of the segment is decided from the nature of sensor [29] [30] [31] [32]. Four sets of feature vectors are computed from these raw data segments. The details about the input features and output classes of each classifier are discussed in the next subsection. All four-feature vector sets are divided into the two parts, 300 feature vectors for the testing and remaining feature vectors for the training and validation of respective ANN design.

4.2 Feature Extraction and Training

The feature vectors are extracted from raw data segments of respective sensor nodes. We selected important and relevant features for the classifications, which used in many previous works. In human activity recognition, standard deviation along three axes [33], mean of gravity acceleration [34] [35] along three axes and Signal Magnitude Area (SMA) [36] [37] are extensively used features. Therefore, this work also used similar features for classification of five basic activates as shown in **Table 2**. In toxic gas classification, many works used normalized samples of gas sensors as feature inputs [25] [38]. Hence, the toxic gases like Acetaldehyde, Acetone, and Toluene has been classified from normalized sensor samples.

Table 2. Details of extracted feature vectors and selected output classes of all four classifiers.

Classification Type		Human Activity Recognition	Abnormal ECG Classification	BP Classification	Toxic Gas Classification
Input Features	F1	X axis Body Acc. Standard Deviation	R-R Interval	R-R Interval	Normalized Output of TGS2600
	F2	Y axis Body Acc. Standard Deviation	QRS Complex Length	S1	Normalized Output of TGS2602
	F3	Z axis Body Acc. Standard Deviation	S-T Interval	S2	Normalized Output of TGS2610
	F4	Body Acc. Signal Magnitude Area (SMA)	P-Q Interval	S3	Normalized Output of TGS2620
	F5	X axis Gravity Acc. Mean	P-R Interval	S4	-
	F6	Y axis Gravity Acc. Mean	-	(X/Y)*1000	-
	F7	Z axis Gravity Acc. Mean	-	-	-
Output Class	C1	Walking	Normal ECG	Normal BP	Acetaldehyde
	C2	Sitting	Abnormal ECG	Low BP	Acetone
	C3	Standing	-	High BP	Toluene
	C4	Laying	-	-	Other
	C5	Transition	-	-	-

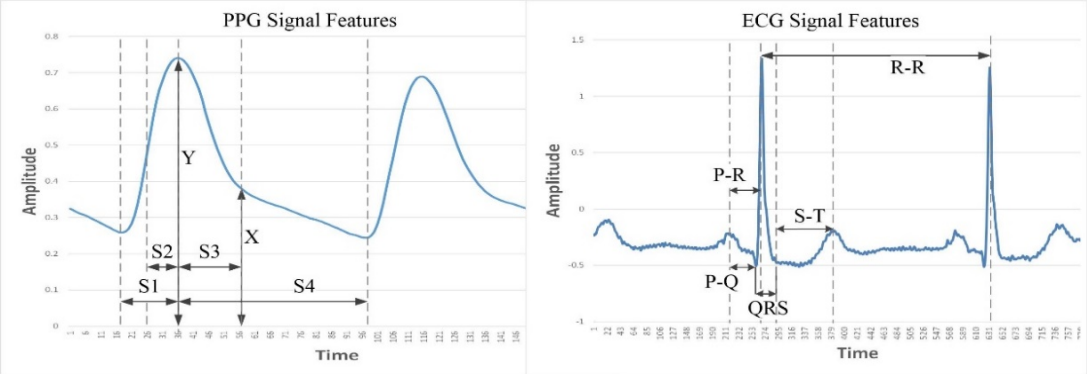

Fig. 4. PPG and ECG morphological features selected for the respective classifications.

As shown in **Fig. 4**, we have selected the morphological features of ECG and PPG signals for the physiological monitoring [24] [39]. We used five types of morphological features extracted from the ECG signal as shown in **Table 2**. ECG features are classified into two-output classes normal and abnormal. Similarly, Cuff-less Blood Pressure (BP) broadly has been classified into three classes normal, low and high, which is evaluated from six features. Five features are extracted from the PPG signal and one feature (R-to-R interval) is calculated from the ECG signal [31] [40]. The detailed deceptions of features and output classes of both physiological parameters are shown in **Fig. 4** and **Table 2**.

As shown in **Table 1**, the total feature vectors of each data set is divided into two parts, from which training and validation dataset is used for the supervised learning of four independent ANN classifiers. The training and validation is conducted independently using MATLAB. All ANN classifiers use fully connected multilayer perceptron (MLP) with a single hidden layer. The training of each classifier is done using the backpropagation algorithm, which uses the gradient descent approach [2]. The numbers of input neurons of each ANN models are decided from the input features of respective classifiers. Similarly, the numbers of output neurons are decided from the output classes of each model. For each ANN classifier, the optimum numbers of hidden layer neurons are estimated from the simulations results shown in **Fig. 6**. All ANN classifiers are trained and tested for all combinations of hidden layer neurons. The topology with maximum accuracy is selected for implementation in each ANN classifier. The weight matrices, bias matrices and input layer constants of all four ANN classifiers are calculated from training and validation, which are used finally in an adaptive ANN designing. The adaptive ANN classifier is implemented on hardware for the testing phase so that it avoids unnecessary hardware burden for training and validation in the present application.

4.3 Hardware Implementation

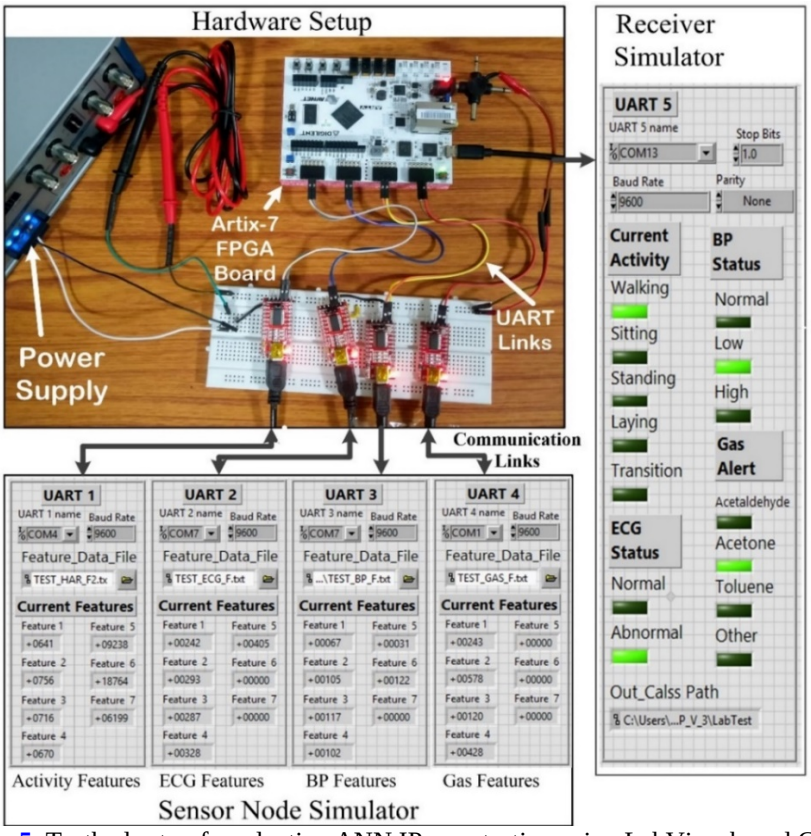

Fig. 5. Testbed setup for adaptive ANN IP core testing using LabView based GUI's.

Generally, the classification algorithms are computationally heavy and efficient implementation of these algorithms on hardware is still a challenging problem. The new generation system on chip solutions is the best choice for the fulfillment of tough system requirements. The five versions of adaptive ANN Intellectual Property (IP) with different data perception (32, 24, 16, 12 and 8 bits) has been implemented and tested on the hardware. The adaptive ANN classifiers are implemented on Xilinx Artix-7 series FPGA's, which works as programmable logic (PL) in the all cost-optimized Xilinx SoC devices. These IP cores of adaptive ANN design gets easily integrate with any Xilinx FPGA or SoC with sufficient hardware resources. The IP cores are designed using Xilinx system generator platform (high-level hardware description tool) that minimized the design time [28].

Each version of IP core is integrated with separate Microblaze processor-based hardware test designs using Vivado IP integrator. Microblaze is programmed with test application using Xilinx software development kit (SDK). All five versions of IP core are tested using testbed setup shown in below. The hardware architectures of all versions are similar except data precision that has a significant influence on hardware performance. The effect of data precision on hardware performances of all five IP cores has been analyzed in subsection 5.2.

All IP cores are operated at 10 MHz operating frequency to optimize power consumption and classification latency. Sigmoid function approximation is designed in the hardware as shown in Equation (4) [26]. The computation of each neuron output in a layer of adaptive ANN executes parallel on the FPGA fabric. For these implementations, we selected Xilinx XC7A35TICSG324-1L FPGA that provides around 30% reduction in power consumption by

switching fabric voltage level from 1 volt to 0.95 volts [41]. Each IP core test design are equipped with five UART connections, four of them are used to collect heterogeneous sensor features and remaining is used to send the analyzed data.

4.4 Testbed Setup

As shown in **Fig. 5**, the practical edge gateway scenario has been created using LABVIEW based sensors and receiver simulators. In the present application, all four heterogeneous sensor nodes communicate with the gateway by using the UART link, which is frequently used in short-range communication applications. In the present application, feature extraction is done on the wearable sensor nodes. Therefore, the similar approach has been used in adaptive ANN IP core testing. The similar ANN test samples (300 per sensor nodes) of four types of heterogeneous sensor nodes are sent to the gateway. Each type of features set is transferred to the adaptive ANN IP core through UART link and Microblaze with appropriate select line (s) input. IP core classifies input features according to select line status and forwards the output class to the receiver simulator. The classification accuracy of all five versions of adaptive ANN IP core implementations are tested in the same way.

5. Results and Analysis

This section covers the complete analysis of adaptive ANN classifier performance in all design phases. The results are divided into three parts from which first two parts are dedicated to simulations and implementation results of adaptive ANN IP core. The final subsection analyzes performance improvement of edge gateway due to the integration of adaptive ANN IP core.

5.1 Simulation Results and Analysis

The collective accuracy analysis of all independent ANN models and the comparison of best models with the previously reported work are discussed in the subsequent subsubsections.

5.1.1 Simulation Classification Accuracy

All ANN models used feedforward multilayer perceptron with the single hidden layer. The number of neurons in the input layer and the output layer is decided from the input features and output classes respectively. The number of hidden layer neurons is decided from accuracy analysis done on all possible ANN topologies. The seven different ANN designs with distinct hidden layer neurons are trained and tested independently on the respective datasets. The classification accuracy obtained from testing of all topologies are shown in **Fig. 6**. The topologies of all four classifiers have been decided similarly using respective heterogeneous datasets. The final topology with maximum classification accuracy is selected for hardware implementation of adaptive ANN classifiers. The weight matrices, bias matrices and input layer constants of selected classifiers have been preserved for the adaptive ANN hardware implementation.

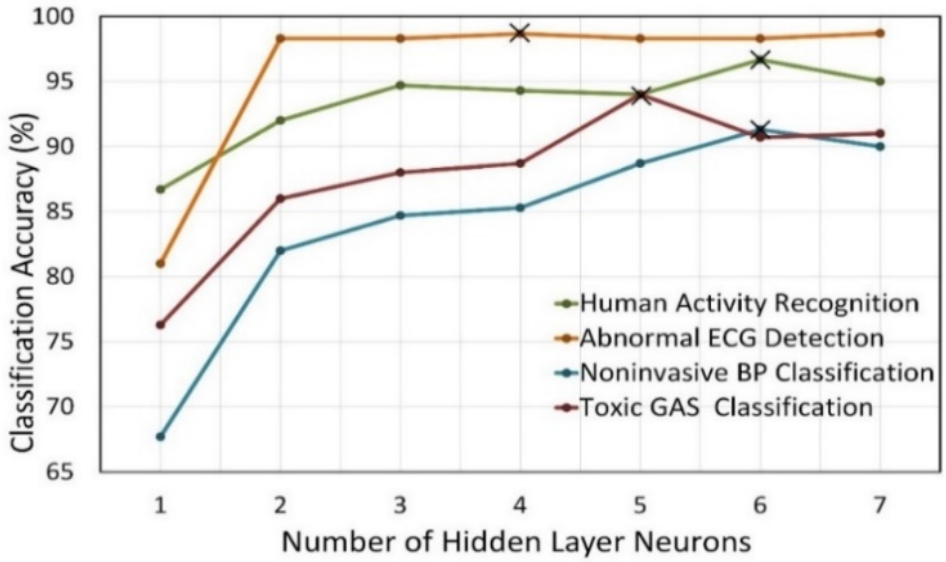

Fig. 6. The classification accuracies comparison of the respective ANN models with different combinations of hidden layer neurons.

5.1.2 Accuracy Comparison

To validate the performance of our proposed method, we compared our work with previously reported results in the literature. Most of them used MLP as a classifier except BP classification. **Table 3** shows the comparison of simulation result with existing works of respective domains. In human activity recognition, the classification accuracy of this work is more than [23] , although the output classes used in this work are less than [23]. In the abnormal ECG classification, the classification accuracy of this work is more than the others except [24]. The BP classification accuracy achieved by this work is almost similar to [31]. Similarly, in the gas classification, the achieved accuracy is intermediate among the mentioned work. The classification accuracy achieved using our proposed MLP approach is very encouraging as it is very close to the best results achieved among most of the mentioned works. In addition, the use of this classifier enables us to integrate all the different classifiers into one single adaptive ANN.

Table 3. The simulation comparison of MLP designs with reported works for respective data analysis.

Application	Works	Feature Used	Classification Algorithm	Output Classes	Classification Accuracy
Human Activity Recognition	[23]	High-level Features (PCA)	MLP	7	89.20%
	This Work	Low-level Features	MLP	5	96.67%
Abnormal ECG Classification	[42]	Normalize Convolute Normalize (NCN)	MLP	2	96.70%
	[24]	DWT (48) and morphological (16) Features	MLP	2	100%
	[39]	Morphological Features	MLP	2	86.33%
	This Work	Morphological Features	MLP	2	98.6%

Noninvasive BP Classification	[31]	Morphological features of PPG and ECG	AdaBoost	3	92%
	This Work	Morphological features of PPG and ECG	MLP	3	91.3%
Gas Classification	[25]	NA	MLP	1	97.40%
	[38]	Sensor values	MLP	5	93.75%
	This Work	Steady-state Feature	MLP	4	94%

5.2 Hardware Results and Analysis

The adaptive ANN is capable to classify any four types of sensor feature sets by switching a select line status. Five hardware versions of adaptive ANN with the different data precision are implemented on the FPGA. The architecture of all versions are kept similar, but the data precision of each neuron is set to 32 bit, 24 bit, 16 bit, 12 bit and 8 bit respectively. The best adaptive ANN hardware design in the context of hardware resource utilization, classification accuracy and power consumption is determined from the classification results and hardware implementation reports of all five versions. All five hardware versions of the adaptive ANN classifier are synthesized, implemented and tested on FPGA.

5.2.1 Hardware Classification Accuracy

Four heterogeneous sets of 300 feature vectors are used for the hardware testing as mentioned in the testbed setup (Section 4.4). The classification accuracies of all five adaptive ANN IP core are computed on the FPGA. **Fig. 7** shows the classification accuracies of all IP cores with respective sensors. The classification results show that the classification accuracy is almost constant for the all IP cores with different data precision selected in this work. A slight reduction in classification accuracy is observed in the case of noninvasive BP classification and toxic gas classification compared with simulation accuracy. This effect is observed due to the sigmoid function approximation and fixed-point operations. There is negligible change in classification accuracy of human activity recognition and abnormal ECG detection compared with simulation results. The human activity recognition and abnormal ECG detection are more robust compared with BP and gas classification.

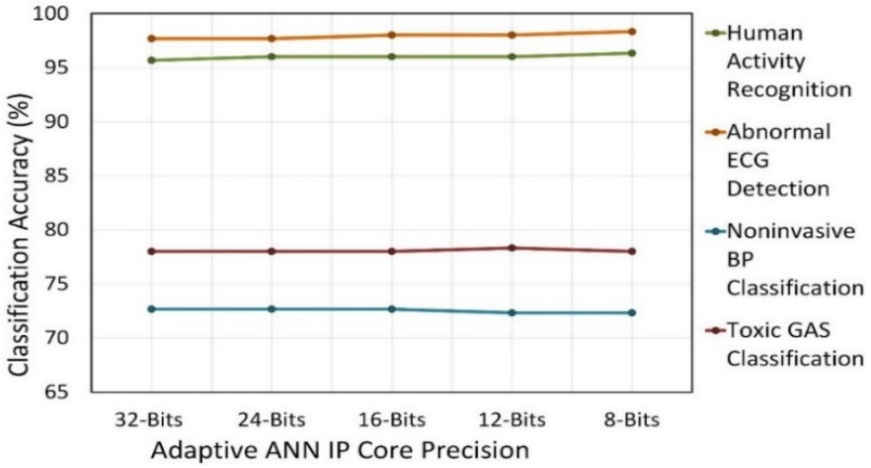

Fig. 7. The classification accuracies comparison of adaptive ANN IP cores versions for respective sensors.

5.2.2 FPGA Resource Utilization

All version of ANN IP cores are integrated into separate Microblaze test systems, and then these designs are synthesized and implemented on the FPGA. The competitive analysis of implementation reports of all IP cores is shown in **Fig. 8**. The important hardware resources like LUT Slices, Register Slices, LUT Flip Flop Pairs and DSP are shown in this analysis. The adaptive ANN with 8-bit data precision requires minimum numbers of FPGA resources. Although, DSP slices utilization is almost constant from 24-bit to 8-bit precision due to the fixed size of DSP Slices (DSP48E1). As mentioned in the above Section 5.2.1, the classification accuracy of all IP cores are almost constant. Therefore, the adaptive ANN with 8-bit precision is more area and resources efficient without compromising the classification accuracy.

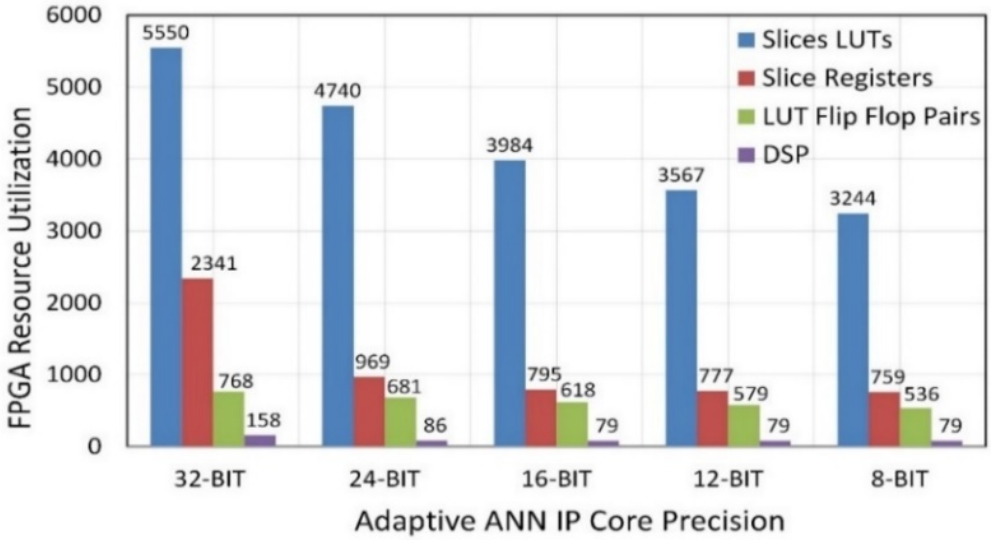

Fig. 8. Effect of change in precision of adaptive ANN IP cores versions on FPGA resource utilization.

5.2.3 Estimated Power Consumption

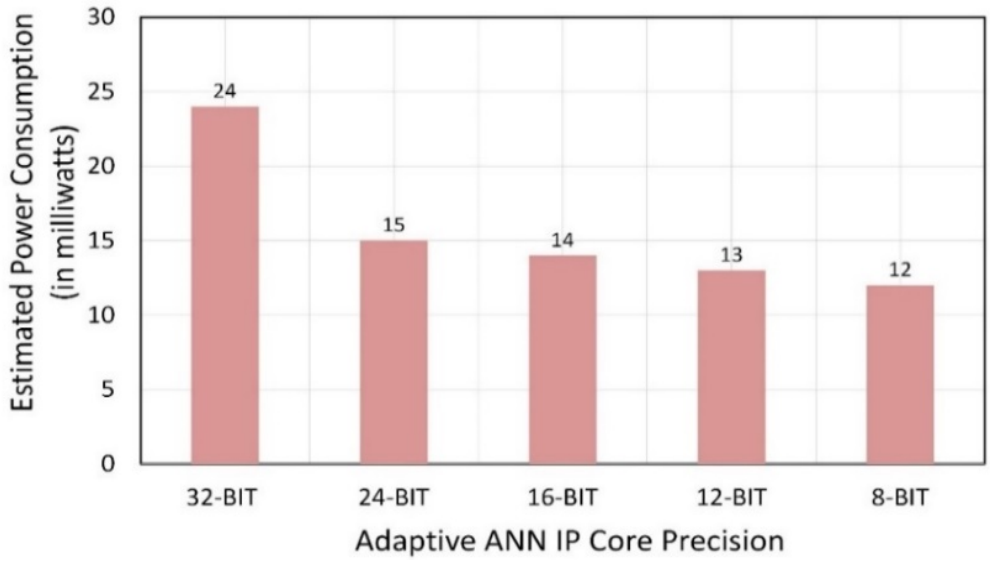

Fig. 9. Effect of change in precision of adaptive ANN IP cores versions on power consumption.

As the edge Gateway is battery powered, therefore minimal power consumption has been achieved by keeping the operating frequency of the adaptive IP cores at 10 MHz but it increases the classification latency. The power estimation of all IP cores are generated independently using Vivado power estimation toolbox, all environmental conditions are kept constant for IP cores. The estimated power comparison of all five adaptive IP cores is shown in **Fig. 9**. The DSP slices are the majority contributor to the total power consumption in all IP cores. As 32-bit IP core uses 158 DSP slices, therefore it uses 11 milliwatt power out of 24 mill watts. In remaining cores, DSP slices contribute approximately half of the total power. The estimated power consumption is proportional to the FPGA resource utilization. Therefore, the adaptive ANN with 8-bit precision is the most efficient adaptive ANN IP core without compromising on classification accuracy.

5.2.4 Classification Latency

The classification latency of the adaptive ANN IP core is evaluated by analyzing the bus transitions of AXI interface that connects the IP core with the test system. **Fig. 10** shows the important signals transitions of the AXI bus during actual hardware testing of the adaptive ANN IP core. It shows one cycle of classification for all four types of sensor feature inputs. These transitions are acquired by using the Vivado system debugger during the hardware run. As shown in figure **Fig. 10**, AXI write bus shows writing operations of respective features in an IP core, also it sends the sensor type (s) and control signals. The evaluated classification outputs are sent back to Microblaze using AXI read bus. AXI write ready and AXI read ready singles are bus control signals required for AXI communications. A segment of these transitions is zoomed in **Fig. 10**, which shows classification delay involved in an ECG classification. As the architecture of all adaptive ANN versions is similar, therefore, the classification latency of all IP cores are also the same i.e. 31 microseconds. This classification latency can be easily reduced by increasing the operating frequency of the adaptive ANN, but the IP core power consumption also increases proportionally.

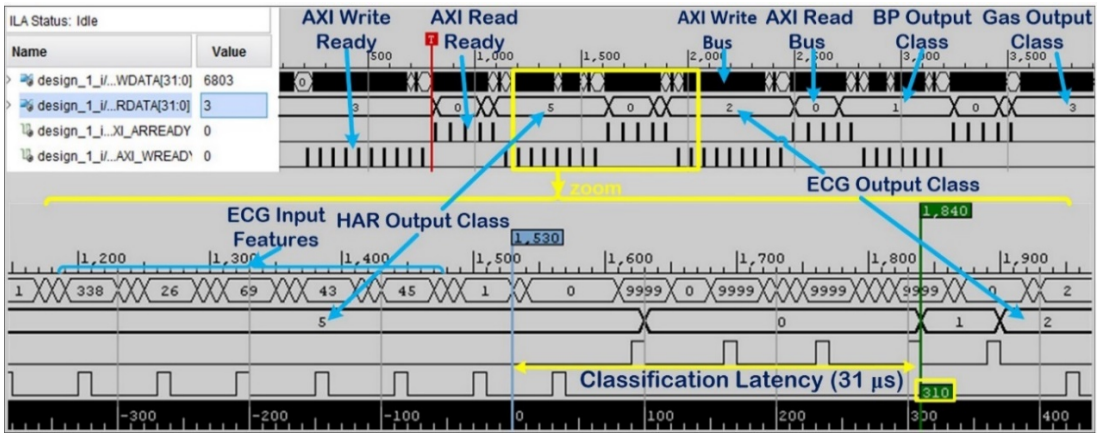

Fig. 10. Timing response of adaptive ANN IP core during live FPGA run for all four heterogeneous classifications.

5.2.5 FPGA Resource Utilization Comparison

Table 4. FPGA resource utilization comparison of adaptive ANN with previously reported ANN implementations.

Works	Implementations Platform	ANN Topology	Data Precision (Bits)	FPGA Resource Utilization			
				LUT	DSP 48	Register Slices	BRAM
[43]	XC5VSX50T	10-3-1	12	8043	70	2243	NA
[25]	Zynq-7000 XC7	12-3-1	24	4032	28	NA	2
[44]	Artix-7 xc7a100t	12-7-3	24	21,658	219	6931	2
[45]	Cyclone V5CEFA9	8-6-1	14	2189	92	3839	NA
[23]	XC6SLX45	6-9-9-7	16	5432	19	4590	65
[38]	XC4020 Ehq208-4	8-4-5	11	1356	NA	NA	86
This Work	Artix-7 Xc7a35ticsg324-1L	Adaptive Topology	8	3244	79	759	0

The hardware design with the least resources requirements provides advantage in terms of power efficiency and cost. Many attempts have been made to implement ANN classifiers on FPGA for various applications. All of these designs have fixed topologies which are only capable of classify single type of sensor features. The comparison of hardware resource utilization of an adaptive ANN IP core (8-bit) with the previously reported ANN implementation is shown in **Table 4**. The data precision of all of these designs are more compared to the adaptive ANN, therefore most of them require more hardware resources. All of these designs are implemented for single application and are not adaptive. These limitations are removed using our proposed adaptive ANN approach. Although the proposed IP core is designed to classify four types of sensor features, still this IP core outperforms among the most of the ANN designs in terms of resource utilization.

5.3 Edge Gateway Performance Analysis

After the extensive experimentation and analysis, the adaptive ANN IP core with 8-bit data precision is finalized as a real-time heterogeneous sensor input analyzer IP in edge gateway. The low latency, throughput reduction and processor offloading are achieved by the incorporation of adaptive ANN IP core on the FPGA or SoC Based edge gateway. This IP core can classify four types of sensor features in real time. **Table 5** shows the percentage saving of FPGA resources achieved by proposed IP core. 61% to 97% reduction in respective resource utilization has been observed compared with the independent FPGA implementation of all four ANN models. Similarly, 44% reduction in estimated power consumption is also achieved due to the selection of adaptive ANN model, which significantly improves the performance of battery enabled edge gateway.

Table 5. FPGA resource reduction and power consumption saving achieved by adaptive ANN IP core.

Performance Parameters		Independent Implementation of all ANN models	Adaptive ANN	Resource Utilization Reduction	Percent Saving
Estimated Power Consumption (milliwatt)		25 mW	14 mW	11 mW	44%
FPGA Resource Utilization	**LUTs Slice**	10236	3984	6252	61%
	Registers Slice	2313	795	1518	66%
	F7 Muxes	98	3	95	97%
	LUT FF Pairs	1602	618	984	61%
	DSP48	216	79	137	63%

96.87% to 98.75% throughput reduction has been achieved by the edge gateway due to the proposed IP core. The classification accuracy of human activity recognition and abnormal ECG detection are almost equal to the simulation classification accuracy. The response time of adaptive ANN IP core is 31 microseconds at 10 MHz operating frequency, due to which edge gateway gives the real-time performance. The IP core executes on Programmable Logic (PL) of SoC, which offloads the processor from the computation burden of data analysis. In total power consumption of edge gateway, the adaptive ANN IP core contributes only 12 milliwatts of power at 10 MHz operating frequency.

6. Conclusion

The real-time and power-efficient heterogeneous sensor data classification on an edge gateway using FPGA based adaptive ANN is implemented in this paper. At first, four independent ANN designs using an optimum number of hidden neurons is designed and later it is integrated into a single adaptive ANN. All previously presented ANN implementations use higher bit precision, which unnecessarily increases the FPGA resource utilization. After extensive experimental analysis, we found that the design with the lowest precision of 8-bit is an efficient hardware design. The proposed adaptive ANN performs classification in 31-microseconds while consuming only 12-milliwatts of power. This performance significantly improves the gateway response time by eliminating network delay required for the data analysis. The selection of fog computing approach reduces the data throughput around 96.87% to 98.75%, which saves communication bandwidth requirements and indirectly reduces the power consumption involved in data transmission. The adaptive ANN hardware design reduces 44% power consumption compared with independent implementations, which help to increase the battery life of edge gateway. It also saves 61% to 97% FPGA resources utilization, which reduces the total system cost. In time-sensitive and battery operated IoT applications, the developed IP core can be used in any Xilinx FPGA/SoC based edge gateway by only changing the weights and biases of the adaptive ANN.

References

[1] J. Yick, B. Mukherjee and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292-2330, 2008. Article (CrossRef Link)

[2] Wei, Wei, and Yong Qi, "Information potential fields navigation in wireless Ad-Hoc sensor networks," *Sensors, MDPI*, vol. 11(5), pp. 4794-4807, 2011.

[3] Wei, Wei, Xiao-Lin Yang, Pei-Yi Shen, and Bin Zhou, "Holes detection in anisotropic sensornets: Topological methods," *International Journal of Distributed Sensor Networks*, vol. 8 (10), p. 135054, 2012. Article (CrossRef Link)

[4] "Fog computing and the Internet of Things: Extend the cloud to where the things are," *cisco*, pp. 1-6, 2015. Article (CrossRef Link)

[5] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang and P. Liljeberg, "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641-658, 2018. Article (CrossRef Link)

[6] J. Ni, K. Zhang, X. Lin and X. S. Shen, "Securing fog computing for internet of things applications: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, pp. 601-628, 2017. Article (CrossRef Link)

[7] P. Hu, S. Dhelim, H. Ning and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," *Journal of Network and Computer Applications*, vol. 98, pp. 27-42, 2017. Article (CrossRef Link)

[8] G. Premsankar, M. D. Francesco and T. Taleb, "Edge Computing for the Internet of Things: A Case Study," *IEEE Internet of Things Journal*, vol. 5, pp. 1275-1284, 2018. Article (CrossRef Link)

[9] M. Suárez-Albela, T. M. Fernández-Caramés, P. Fraga-Lamas and L. Castedo, "A practical evaluation of a high-security energy-efficient gateway for IoT fog computing applications," *Sensors*, vol. 17, p. 1978, 2017. Article (CrossRef Link)

[10] R. Morabito, R. Petrolo, V. Loscri and N. Mitton, "LEGIoT: a Lightweight Edge Gateway for the Internet of Things," *Future Generation Computer Systems*, vol. 81, pp. 1-15, 2018. Article (CrossRef Link)

[11] "The Internet of Things for Defense," *WIND*, 2016. Article (CrossRef Link)

[12] P. Fraga-Lamas, T. Fernández-Caramés, M. Suárez-Albela, L. Castedo and M. González-López, "A review on internet of things for defense and public safety," *Sensors*, vol. 16, p. 1644, 2016. Article (CrossRef Link)

[13] G. J. García, C. A. Jara, J. Pomares, A. Alabdo, L. M. Poggi and F. Torres, "A survey on FPGA-based sensor systems: towards intelligent and reconfigurable low-power sensors for computer vision, control and signal processing," *Sensors*, vol. 14, pp. 6247-6278, 2014. Article (CrossRef Link)

[14] A. De La Piedra, A. Braeken and A. Touhafi, "Sensor systems based on FPGAs and their applications: A survey," *Sensors*, vol. 12, pp. 12235-12264, 2012. Article (CrossRef Link)

[15] E. Koromilas, I. Stamelos, C. Kachris and D. Soudris, "Spark acceleration on FPGAs: A use case on machine learning in Pynq," in *Proc. of Modern Circuits and Systems Technologies (MOCAST), 2017 6th International Conference on*, 2017. Article (CrossRef Link)

[16] D. Beserra, M. K. Pinheiro, C. Souveyet, L. A. Steffenel and E. D. Moreno, "Performance evaluation of os-level virtualization solutions for hpc purposes on soc-based systems," in *Proc. of Advanced Information Networking and Applications (AINA), 2017 IEEE 31st International Conference on*, 2017. Article (CrossRef Link)

[17] "Zynq-7000 All Programmable SoC Family Product Tables and Product selection Guide," *XILINX*. Article (CrossRef Link)

[18] C.-H. Chen, M.-Y. Lin and C.-C. Liu, "Edge Computing Gateway of the Industrial Internet of Things Using Multiple Collaborative Microcontrollers," *IEEE Network*, vol. 32, pp. 24-32, 2018. Article (CrossRef Link)

[19] S. A. Karthikeya, R. Narayanan and others, "Power-aware gateway connectivity in battery-powered dynamic IoT networks," *Computer Networks*, vol. 130, pp. 81-93, 2018. Article (CrossRef Link)

[20] N. Verba, K.-M. Chao, A. James, D. Goldsmith, X. Fei and S.-D. Stan, "Platform as a service gateway for the Fog of Things," *Advanced Engineering Informatics*, vol. 33, pp. 243-257, 2017. Article (CrossRef Link)

[21] B. Varghese, N. Wang, D. S. Nikolopoulos and R. Buyya, "Feasibility of fog computing," *arXiv preprint arXiv:1701.05451*, 2017. Article (CrossRef Link)

[22] S. K. Sharma and X. Wang, "Live Data Analytics With Collaborative Edge and Cloud Processing in Wireless IoT Networks," *IEEE Access*, vol. 5, pp. 4621-4635, 2017. Article (CrossRef Link)

[23] K. Basterretxea, J. Echanobe and I. Campo, "A wearable human activity recognition system on a chip," in *Proc. of Design and Architectures for Signal and Image Processing (DASIP), 2014 Conference on*, 2014. Article (CrossRef Link)

[24] H. M. Rai, A. Trivedi and S. Shukla, "ECG signal processing for abnormalities detection using multi-resolution wavelet transform and Artificial Neural Network classifier," *Measurement*, vol.

46, pp. 3238-3246, 2013. Article (CrossRef Link)

[25] X. Zhai, A. A. S. Ali, A. Amira and F. Bensaali, "MLP neural network based gas classification system on Zynq SoC," *IEEE Access*, vol. 4, pp. 8138-8146, 2016. Article (CrossRef Link)

[26] 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, pp. 1124-1133, 2016. Article (CrossRef Link)

[27] G. Hinton, "Lecture 4.3 — The softmax output function [Neural Networks for Machine Learning]". Article (CrossRef Link)

[28] N. I. Chervyakov, P. A. Lyakhov, M. G. Babenko, I. N. Lavrinenko, A. V. Lavrinenko and A. S. Nazarov, "The architecture of a fault-tolerant modular neurocomputer based on modular number projections," *Neurocomputing*, vol. 272, pp. 96-107, 2018. Article (CrossRef Link)

[29] 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. Article (CrossRef Link)

[30] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, pp. e215-e220, 2000. Article (CrossRef Link)

[31] 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, pp. 859-869, 2017. Article (CrossRef Link)

[32] A. Vergara, S. Vembu, T. Ayhan, M. A. Ryan, M. L. Homer and R. Huerta, "Chemical gas sensor drift compensation using classifier ensembles," *Sensors and Actuators B: Chemical*, vol. 166-167, pp. 320-329, 2012. Article (CrossRef Link)

[33] N. Pannurat, S. Thiemjarus, E. Nantajeewarawat and I. Anantavrasilp, "Analysis of Optimal Sensor Positions for Activity Classification and Application on a Different Data Collection Scenario," *Sensors*, vol. 17, p. 774, 2017. Article (CrossRef Link)

[34] A. Bayat, M. Pomplun and D. A. Tran, "A study on human activity recognition using accelerometer data from smartphones," *Procedia Computer Science*, vol. 34, pp. 450-457, 2014. Article (CrossRef Link)

[35] J. Guo, X. Zhou, Y. Sun, G. Ping, G. Zhao and Z. Li, "Smartphone-Based Patients' activity recognition by using a self-learning scheme for medical monitoring," *Journal of medical systems*, vol. 40, p. 140, 2016. Article (CrossRef Link)

[36] L. Gao, A. K. Bourke and J. Nelson, "Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems," *Medical engineering & physics*, vol. 36, pp. 779-785, 2014.

[37] J.-Y. Yang, J.-S. Wang and Y.-P. Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers," *Pattern recognition letters*, vol. 29, pp. 2213-2220, 2008. Article (CrossRef Link)

[38] F. Benrekia, M. Attari and M. Bouhedda, "Gas sensors characterization and multilayer perceptron (MLP) hardware implementation for gas identification using a field programmable gate array (FPGA)," *Sensors*, vol. 13, pp. 2967-2985, 2013. Article (CrossRef Link)

[39] J. Adnan, N. G. N. Daud, A. S. N. Mokhtar, F. R. Hashim, S. Ahmad, A. F. Rashidi and Z. I. Rizman, "Multilayer perceptron based activation function on heart abnormality activity," *Journal of Fundamental and Applied Sciences*, vol. 9, pp. 417-432, 2017. Article (CrossRef Link)

[40] M. Kachuee, M. M. Kiani, H. Mohammadzade and M. Shabany, "Cuff-less high-accuracy calibration-free blood pressure estimation using pulse transit time," in *Proc. of Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, 2015. Article (CrossRef Link)

[41] Ben Levinsky, "ARTY Power Demo," *Xilinx Wiki*, 2018. Article (CrossRef Link)

[42] K. A. Sidek, I. Khalil and H. F. Jelinek, "ECG biometric with abnormal cardiac conditions in remote monitoring system," *IEEE Transactions on systems, man, and cybernetics: systems*, vol. 44, pp. 1498-1509, 2014. Article (CrossRef Link)

[43] A. Gomperts, A. Ukil and F. Zurfluh, "Development and implementation of parameterized FPGA-based general purpose neural networks for online applications," *IEEE Transactions on Industrial Informatics*, vol. 7, pp. 78-89, 2011. Article (CrossRef Link)

[44] M. Bahoura, "Fpga implementation of blue whale calls classifier using high-level programming tool," *Electronics*, vol. 5, p. 8, 2016. Article (CrossRef Link)

[45] Z. Szadkowski, D. Głas, K. Pytel and M. Wiedeński, "Optimization of an FPGA trigger based on an artificial neural network for the detection of neutrino-induced air showers," *IEEE Trans. Nucl. Sci.*, vol. 64, pp. 1271-1281, 2017. Article (CrossRef Link)

Nikhil B. Gaikwad was born in 1989. He completed his B.E. in Electronics and communications engineering from Nagpur University in 2012. He completed his M. Tech from VJTI, Mumbai, India in 2015. He is currently pursuing Ph.D from Visvesvaraya National Institute of Technology (VNIT), Nagpur, India, under the supervision of Dr. Avinash Keskar. He also served as a research scientist at Society for Applied Microwave Electronics Engineering & Research (SAMEER), Mumbai during 2015-20016. His area of research includes FPGA based artificial intelligence, reconfigurable systems, sensor systems, fog computing and Internet of Things.

Varun Tiwari was born in 1988. He completed his B.E. in Electronics engineering from Nagpur University in 2011 and M.Tech from YCCE, Nagpur, India in 2013. He is currently pursuing Ph.D. from Visvesvaraya National Institute of Technology (VNIT), Nagpur, India, under supervision of Dr. Avinash Keskar and Dr. NC Shivaprakash. His area of research includes human machine interaction, internet of things, digital system design, image processing and computer vision.

Avinash Keskar was born in 1959, India. He completed his B.E. from VNIT, Nagpur in 1979 and received gold medal for the same. He completed his M.E. from IISc, Bangalore in 1983, receiving the gold medal again. He received his Ph.D in 1997 from Nagpur University. He has 26 years of teaching experience and 7 years of industrial experience. He is currently a professor in Department of Electronics and Communication Engineering, VNIT Nagpur. His current research interests include embedded systems, computer vision, soft computing, fuzzy logic, etc. Dr. Keskar is a senior member of IEEE, FIETE, LMISTE, FIE.

N.C. Shivaprakash was born in 1955. He completed his B Sc., M Sc., and Ph.D. from Mysore University, India in 1976, 1978 and 1982. He is associated with IISc, Bangalore since 1983 and is currently Chief Research Scientist in the department of Instrumentation and applied physics. His research interest includes embedded systems, intelligent instrumentations, structure and physical properties of liquid crystals, high pressure DTA, etc. He is a senior member of IEEE (USA) and fellow institute of electronics and telecommunication.