Automatic disease diagnosis using optimised weightless neural networks for low-power wearable devices

Ramalingaswamy Cheruku[1] ✉, Damodar Reddy Edla[1], Venkatanareshbabu Kuppili[1], Ramesh Dharavath[2], Nareshkumar Reddy Beechu[3]

[1]Department of Computer Science and Engineering, National Institute of Technology Goa, Ponda, Goa 403401, India
[2]Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, Jharkhand 826004, India
[3]Department of Electronics and Communication Engineering, National Institute of Technology Goa, Ponda, Goa 403401, India
✉ E-mail: rmlswamygoud@nitgoa.ac.in

Low-power wearable devices for disease diagnosis are used at anytime and anywhere. These are non-invasive and pain-free for the better quality of life. However, these devices are resource constrained in terms of memory and processing capability. Memory constraint allows these devices to store a limited number of patterns and processing constraint provides delayed response. It is a challenging task to design a robust classification system under above constraints with high accuracy. In this Letter, to resolve this problem, a novel architecture for weightless neural networks (WNNs) has been proposed. It uses variable sized random access memories to optimise the memory usage and a modified binary TRIE data structure for reducing the test time. In addition, a bio-inspired-based genetic algorithm has been employed to improve the accuracy. The proposed architecture is experimented on various disease datasets using its software and hardware realisations. The experimental results prove that the proposed architecture achieves better performance in terms of accuracy, memory saving and test time as compared to standard WNNs. It also outperforms in terms of accuracy as compared to conventional neural network-based classifiers. The proposed architecture is a powerful part of most of the low-power wearable devices for the solution of memory, accuracy and time issues.

1. Introduction: In the present era, machine learning techniques have been used by several researchers in the medical domain for diagnosis of various diseases. In the literature, there exists so many wearable diagnostic devices, such as Gluco Track [1], Dexcom G5 [2], QuantuMDx [3], Gluco Beam [4], GeneXpert [5] and so on. These devices are non-invasive and pain-free for the better quality of life [4]. However, it is hard to embed these techniques in low-power hardware devices because of its memory, power and speed constraints. In this direction, Bledsoe and Browning in 1959 [6] invented weightless neural network (WNN). It is a breakthrough among neural network techniques suitable for hardware implementation [7]. The major advantages of this kind of network are the ease of implementation and the ability to learn in single iteration. WNNs are also called *n*-tuple classifiers. These classifiers offer fast training and testing performance. Standard multi-layer feed-forward neural network (MLFFNN) stores knowledge in the form of network weights whereas WNNs store knowledge in random access memory (RAM). The first version of WNN was designed by Bledsoe and Browning in 1959 and various improved WNNs [8] have been developed by many researchers. All these implementations of WNNs have the following common properties: (i) interconnections in WNN do not carry weights; (ii) the WNN can only take binary inputs; and (iii) the knowledge of network is stored in the form of binary look-up tables (LUTs).

In simple WNNs implementation, RAM is used as a LUT. Each neuron (RAM) synapses (address lines) is supplied with a binary bit string from the input pattern. This binary string is used as an address to access the RAM. In the training phase, this binary string is used as an address to store desired output in the RAM. In the testing phase, a binary unseen test pattern is provided as an address to access the previously learned contents from the RAM. It is observed from the above training and testing procedures that training and testing in WNNs are made in a single iteration. It is also observed that the neuron (RAM) size grows exponentially with the size of the input vector because of RAM size is always in the power of 2.

As the WNN proposed by Bledsoe and Browning suffers from a memory problem, to resolve this Wilkes and co-workers [9] have proposed WiSARD WNNs. WiSARD is identical to the simplest WNN, except that in WiSARD the input pattern is partitioned into multiple segments. Every WiSARD WNN is made up of several RAM-discriminators and each RAM-discriminator made up of Y one-bit word RAMs. Each one-bit RAM receives a portion of binary input pattern (n-bits). Along with Y one-bit word RAMs, each RAM-discriminator also consists of a summing device (Σ). The number of such RAM-discriminators is equal to the number of distinct classes in the dataset. In the training phase, a binary pattern of $Y*n$ bits partitioned randomly into Y equal sized segments. These segmented binary patterns are used as the address of RAMs to store desired output. In the testing phase, binary input pattern is supplied to each discriminator and every discriminator provides a response in terms of number of matches. These responses are evaluated according to the majority voting principle. Schematic representation of both WiSARD and RAM-discriminator is shown in Fig. 1 [7].

As shown in Fig. 1, WiSARD overcomes limitation of simple WNNs by dividing the $Y*n$-sized input vector in Y segments. As a result the total memory requirement is reduced from 2^{Y*n} to $Y*2^n$. On the other side, it lowers the generalisation capability of WiSARD. This caused applications using WiSARD, performing lower than that of virtual generalising RAM (VG-RAM) WNN [10, 11]. However, WiSARD has used to solve the problems in automatic video surveillance [12], robotics [13], 3D video animation [14] and text categorisation [15].

The VG-RAM WNN is a different type of WNN, in which each neuron memory size is proportional to the training set [10, 16]. During training, VG-RAM neurons store input binary pattern along with its associated class information. During the testing, each VG-RAM neuron searched for closest learned pattern using
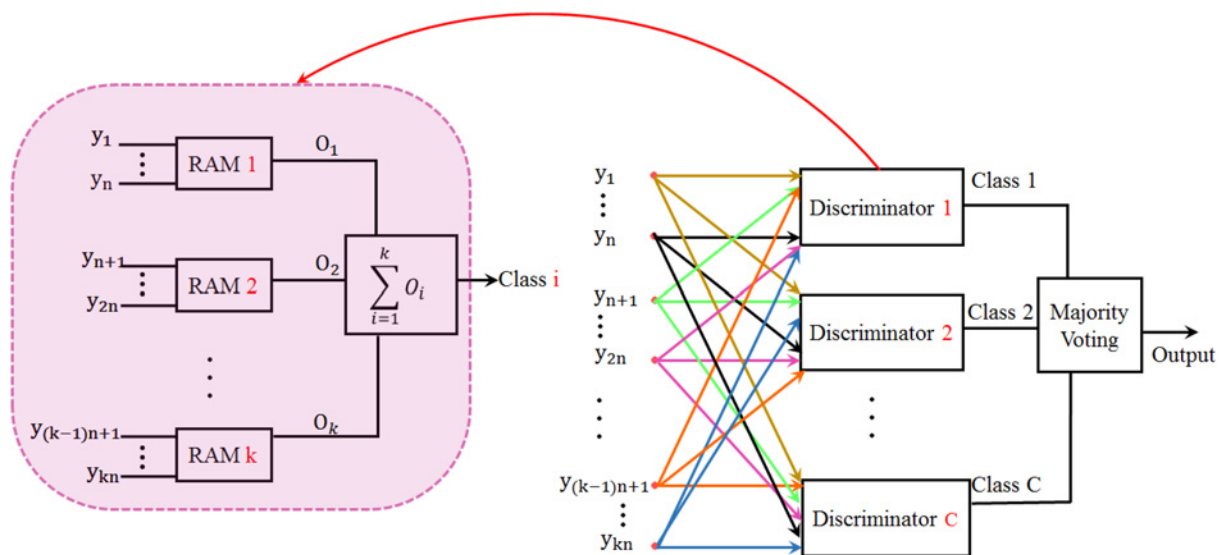
Fig. 1 *Illustration of both RAM-discriminator and WiSARD WNN*

any distance measure, such as hamming distance and so on. The class information of the closest input pattern is the neurons output. The searching procedure for the closest pattern is sequential and requires scanning of each neuron whole memory, which is costly in terms of time, if there are many training patterns. Moreover, the memory size of each VG-RAM is increasing on par with training patterns. Even though VG-RAM WNNs have these limitations, these are used in many applications, such as face recognition [17], text categorisation [18] and traffic sign detection [18].

To deal with such problems, a novel architecture for WNNs is proposed in this Letter. It uses variable sized RAMs (neurons) to optimise the memory usage, and a modified binary TRIE data structure for reducing the test time [19, 20]. In addition, a genetic algorithm (GA) is used to improve the accuracy of WNNs by optimising the mapping function [21].

2. Proposed architecture: The proposed architecture emphasises on improving the classification accuracy, reducing the memory

usage and finally reducing the test time. To accomplish all these tasks, different techniques have been employed and all of them are explained in Sections 2.1–2.3. The proposed architecture is shown in Fig. 2. According to this figure, the input binary pattern is mapped to set of VG-RAM neurons using the mapping function for improving the accuracy. During the training, in each VG-RAM neuron, patterns are managed using TRIE data structure (for simplicity of diagram only prefix, access counts of each class are shown, class information is not shown in TRIE node). This helps us in memory saving and faster access to contents. During testing, each neuron outputs class label, final class is determined on the basis of the majority voting.

2.1. Memory reduction using variable sized VG-RAM (VVG-RAM) neuron: VG-RAM neuron stores both patterns and its class information. Hence, its size is proportional to the training set. Usually count of such VG-RAMs neurons needed is undeterministic. In this Letter, VVG-RAMs have been proposed to optimise the memory usage. The size of each VVG-RAM
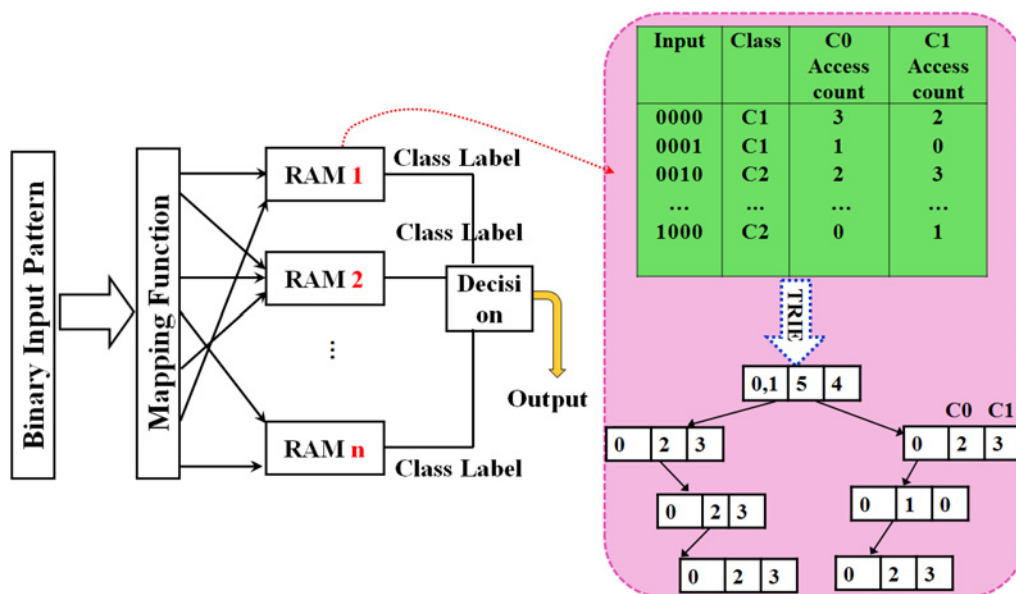

Fig. 2 *Proposed architecture for the classification task*

neuron is determined by a range of values that features can take. However, also count of such neurons is determined by dimensionality of the data. The motivation behind this approach is that every dataset consists of some repetitive subpatterns. By eliminating these repetitive subpatterns, it is possible to constitute VVG-RAM. Proposed VVG-RAM neuron maintains extra information about these repetitive subpatterns in the form of access count (see Fig. 3). This extra information plays a vital role in decision process, especially in case of ties. Architectures of VG-RAM neuron and VVG-RAM neuron have been shown in Figs. 3a and b. Further, the proposed architecture employs the binary TRIE data structure [19, 20] to manage the patterns inside each VVG-RAM neuron.

2.2. Performance improvement using GA: Mapping function maps input pattern to synapses of VVG-RAM neuron, such that one synapse is mapped to exactly one neuron. To improve the VG-RAM network performance, optimal mapping function between input pattern and neuron synapses has to be defined. If there is an N-bit binary pattern there exist $N!$ possible combinations. To select optimal or near optimal mapping by doing an exhaustive search of $N!$ combinations is NP-hard problem. Hence, this combinatorial optimisation problem is solved by one of the most familiar GA with the objective of maximising the objective function. GA parameters have to be properly tuned to obtain the optimal or near optimal solution and these parameter values are dataset specific. The objective function based on sensitivity and specificity is defined in (1) [22]. It is clear from (1) that the objective function is a geometric mean of sensitivity and specificity. Sensitivity focuses only on the positive class case predictions and does not capture any information about how well the WNN handles negative class cases. Similarly, specificity focuses only on the negative case predictions and does not capture any information about how well the WNN handles positive cases. To balance the both positive and negative class predictions, a new measure has been proposed as a geometric mean of sensitivity and specificity

$$\text{Objective function} = \left(\sqrt{\text{sensitivity} * \text{specificity}} \right) \quad (1)$$

$$\text{Sensitivity} = \left(\frac{\text{TP}}{\text{TP} + \text{FN}} \right) \quad (2)$$

$$\text{Specificity} = \left(\frac{\text{TN}}{\text{TN} + \text{FP}} \right) \quad (3)$$

$$\text{Accuracy} = \left(\frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \right) \quad (4)$$

where TP represents the true positive count, which is calculated as the number of positive class records that the WNN predicts as positive, TN represents the true negative count, which is calculated as the number of negative class records that the WNN predicts as negative, FP represents the false positive count, which is calculated as the number of negative class records that the WNN incorrectly classifies as positive and FN represents the false negative count, which is calculated as the number of positive class records that the WNN incorrectly classifies as negative [23].

2.3. Faster neuron memory search with modified TRIE data structure: Usually in standard VG-RAM networks, during the testing to measure the closeness of test pattern over patterns stored in neuron, distances need to be calculated sequentially. As a result test time increases. Since hashing technique is more appropriate for search and insertion operations, Forechi et al. [24] have used a hash table for test time reduction. Still hash tables are not an efficient solution as it increases chance of collisions as the number of entries grows. Moreover, it is desirable to design an effective hash function to handle these collisions; this imposes the computational complexity on hash function. Hence, to address these problems in this Letter a modified version of the binary TRIE data structure has been proposed and it is shown in Fig. 4.

According to Fig. 3, every node of the binary TRIE data structure stores pattern prefixes, class information along with access counts of each class during the training. During the test, it finds a longest prefix match. The output of each neuron is the class information along with each class access count associated with the longest prefix match. In case of tie, access counts of each class are useful to determine the class of the test pattern on the majority basis. Unlike standard VG-RAM neuron, the proposed VVG-RAM neuron does not require any distance computations for measuring similarity. As compared to hash-technique-based neuron, proposed neuron is free from hash function calculations. Hence, the proposed VVG-RAM neuron is efficient in terms of computations (calculations) as compared with standard VG-RAM and hash-technique-based neurons.

3. Results

3.1. Experimental setup: All software and hardware experiments are performed on an Intel (R) Core i7 processor with 3.60 GHz speed and 8 GB RAM. Three categories of datasets, such as more data with low dimensionality (MDLD), small data with high dimensionality (SDHD) and more data with high dimensionality (MDHD) are chosen from UCI machine learning repository [25] to validate proposed WNN. All the datasets used for experiments are shown in Table 1 and are partitioned according to 10-fold cross-validation (10-FCV) method [26]. According 10-FCV,

Pattern	Class
0000	C1
0001	C1
0010	C2
...	...
1000	C2

Input	Class	Class 0 Access count	Class 1 Access count
0000	C1	3	2
0001	C1	1	0
0010	C2	2	3
...	...	...	...
1000	C2	0	1

a	b

Fig. 3 Comparison of VG-RAM and VVG-RAM neuron structures
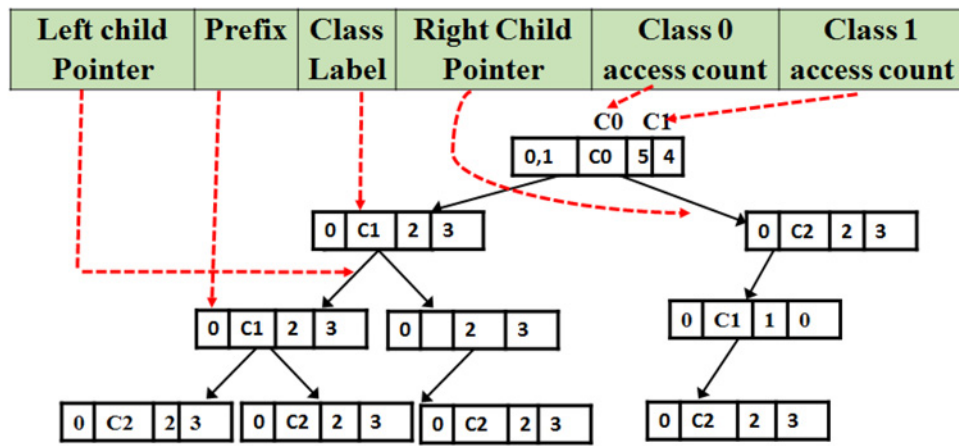a VG-RAM neuron
b Proposed VVG-RAM neuron

Fig. 4 *Modified binary TRIE data structure*

Table 1 Datasets used in this Letter

Dataset	Number of patterns	Dimensionality
PID	768	8
LCD	32	56
DRD	1151	19

every dataset is partitioned into ten equal folds. Every time, nine different folds constitute the training set and remaining fold is treated as the testing set. During the training phase, the training dataset is used to store the contents into RAM nodes (neurons). During the testing phase, the testing dataset is used to evaluate the performance of trained model using performance measures, such as accuracy, sensitivity and so on. This process is repeated for ten folds and at the end of tenth fold all the values are averaged. As the GA parameter values are dataset specific find tuned parameter values for three datasets are shown in Table 2.

To validate the performance of proposed WNN further it compared with conventional neural networks, such as multi-layer perceptron network (MLPN) [27], MLFFNN [27], probabilistic neural network (PNN) [28], radial basis function neural network (RBFNN) [27] and time delay network (TDN) [29].

The MLPN and MLFFNN used in these experiments are constructed with one input layer, two hidden layers and one output layer. Hidden layers of MLFFNN consist of 21 and 19 neurons, whereas in MLPN consists of 9 and 5 neurons. The MLPN and MLFFNN are trained using back propagation and scaled conjugate gradient algorithms, respectively. The learning rate and maximum epochs for both the training algorithms are set to 0.01 and 1000, respectively. As the output values of both the networks are between 0 and 1, a transformation function has been applied with cut off of 0.5, to transform into binary values. Similarly, the RBFNN and PNN are configured with spread value of 1.60. Finally, TDN is providing the eight positive vectors as input delays.

3.2. Performance measures: We measure the memory usage of WiSARD, standard VG-RAM and proposed VVG-RAM using the following equations:

$$M_{\text{WiSARD}} = \left(C * N * \left(2^{S/N} * 1 \right) \right) \text{ bits} \qquad (5)$$

where S is the number of synapses, N is the number of neurons chosen such that $N * (2^{S/N}) \geq \text{Tr}$, and C is the number of classes

$$M_{\text{VG-RAM}} = \left(\text{Tr} * N * \left(2^B * (B + 1) \right) \right) \text{ bits} \qquad (6)$$

where Tr is the number of training samples, B is an integer chosen such that $S \leq 2^B$, N is the number of neurons such that $N = S/B$.

$$M_{\text{Proposed}} = \left(\sum_{i=1}^{D} 2^{b_i} * (b_i + 1 + C * a_i) \right) \text{ bits} \qquad (7)$$

Table 2 GA tuning parameter values for three medical datasets

Dataset	Parameter	Value	Explanation
PID	PopSize	150	initial population size
	cross-over rate	0.82	cross-over rate
	MutRate	0.1	mutation rate
	selection rate	0.6	population that survive after every generation
	MaxGen	10,000	maximum number of generations
LCD	PopSize	250	initial population size
	cross-over rate	0.32	cross-over rate
	MutRate	0.01	mutation rate
	selection rate	0.7	population that survive after every generation
	MaxGen	25,000	maximum number of generations
DRD	PopSize	100	initial population size
	cross-over rate	0.8	cross-over rate
	MutRate	0.14	mutation rate
	selection rate	0.6	population that survive after every generation
	MaxGen	10,000	maximum number of generations

where D is the dataset dimensionality, b_i is the number of bits required to represent values of ith feature and a_i is the number of bits to represent each class access count of ith feature, i.e. $a_i = \lambda * b_i$, $0 < \lambda \leq 1$. In all the experiments, λ value chosen as $1/b_i$.

Apart from memory usage of each model, accuracy and test time also considered for evaluation. Accuracy is used to measure the overall predictive performance of model on unseen data. It is calculated according to (4). Test time is used to measure the fastness in the model response, i.e. how fast the model providing the response. It is calculated using *tic* and *toc* functions of Matlab.

3.3. Software realisation: The proposed WNN along with two standard WNNs namely WiSARD and VG-RAM are implemented using Matlab 2015a software. These WNNs are experimented on various datasets to obtain performance measures like memory usage, accuracy and test time. These results are shown in the following subsections.

3.3.1. More data with low dimensionality: In this category, Pima Indians Diabetes (PID) dataset has been selected. The PID dataset consists of 768 records of diabetes patients, of which 500 are negative and 268 are positive classes [25]. It has eight predictive attributes, one decision attribute. The experimental results on this dataset are provided in Table 3 in terms of memory usage, accuracy and test time. From Table 3, it is observed that the proposed WNN performed better than standard WNNs in terms of memory usage, accuracy and test time. Further, proposed method also compared with conventional neural network classifiers in Table 4 and best values are highlighted in bold. It is observed from the table results that proposed method achieved best rank in terms of accuracy. It is due to the GA performance on this dataset. In this Letter, GA parameters are used specifically to dataset (see Table 2).

3.3.2. Small data with high dimensionality: In this category, Lung Cancer Dataset (LCD) has been chosen. This dataset has 32 instances with 57 attributes (1 decision, 56 predictive) [25]. Results against this dataset, in terms of memory, accuracy and time, are furnished in Table 5 and best values are highlighted in bold. These results show that the proposed WNN requires more

memory than WiSARD and standard VG-RAM WNNs, it is due to high dimension of data and features are taking high range values. From (7), it is clear that memory required for the proposed architecture, i.e. M_{Proposed} is proportional to dimensionality of data (D) and range of values (b_i) features can take. However, proposed method performed better in terms of accuracy. It is due to the dataset-specific GA parameter tuning (see Table 2).

Further, the proposed WNN has been compared against five popular neural network-based classifiers, namely MLPN, MLFFNN, PNN, RBFN and TDN. These comparison results are shown in Table 6 and best values are highlighted in bold. It is observed from these results that the proposed WNN outperformed in terms of accuracy as compared to other neural network-based classifiers.

3.3.3. More data with high dimensionality: In this category, Diabetic Retinopathy Debrecen (DRD) dataset has been selected. It has 1151 instances with 20 attributes (1 class attribute, 19 predictive) [25]. Experimental results on this dataset are furnished and best values are highlighted in bold in Table 7. From these results, it is observed that the proposed WNN outperformed in terms of accuracy and test times as compared to standard WNNs.

Further, the proposed WNN has been compared with five popular neural network-based classifiers, such as MLPN, MLFFNN, PNN, RBFN and TDN. These comparison results are shown in Table 8 and best values are highlighted in bold. It is clear from the table results that the proposed WNN outperformed all other classifiers. Best values are highlighted in bold in Table 8.

From Tables 3–8 results, it is observed that the proposed WNN performed better in terms of accuracy for the datasets of kind MDLD, SDHD and BDHD. Datasets like where the number of patterns less than dimensionality, i.e. SDHD, the proposed method suffered from memory problem this in turn created delayed response.

From all the above results, it is also observed that a number of features and range values that each feature can determine the

Table 3 Comparison results on PID dataset

Type of WNN	Memory, KBs	Accuracy, %	Test time, s
WiSARD	35	64.72	79.23
VG-RAM	31.5	70.15	61.95
proposed	**21.25**	**78.23**	**47.27**

Table 4 Accuracy comparison on PID dataset

S. no	Type of neural network classifier	Accuracy, %
1	MLPN	75.20
2	MLFFNN	74.00
3	PNN	67.20
4	RBFN	68.53
5	TDN	66.54
6	proposed WNN	**78.23**

Table 5 Comparison results on LCD

Type of WNN	Memory, KBs	Accuracy, %	Test time, s
WiSARD	512	82.31	148.41
VG-RAM	**4.3125**	86.75	**43.86**
proposed	6.3	**86.73**	44.72

Table 6 Accuracy comparison on LCD

S. no	Type of neural network classifier	Accuracy, %
1	MLPN	70.20
2	MLFFNN	60.30
3	PNN	60.40
4	RBFN	49.10
5	TDN	75.45
6	proposed WNN	**86.73**

Table 7 Comparison results on DRD dataset

Type of WNN	Memory, KBs	Accuracy, %	Test time, s
WiSARD	512	68.72	191.33
VG-RAM	121	70.15	173.61
proposed	**22.93**	**72.86**	**63.73**

Table 8 Accuracy comparison on DRD dataset

S. no	Type of neural network classifier	Accuracy, %
1	MLPN	53.17
2	MLFFNN	66.10
3	PNN	60.69
4	RBFN	45.08
5	TDN	49.70
6	proposed WNN	**72.86**

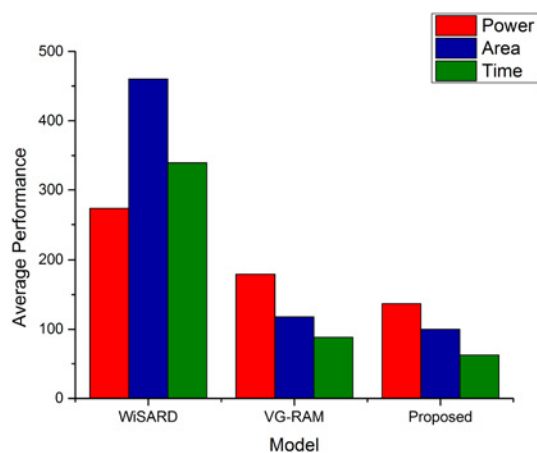**Fig. 5** *FPGA SPARTAN 3E tool kit*

Fig. 6 *Average performance of WiSARD, VG-RAM and proposed model over three medical datasets*

memory usage of proposed model. Also, proper tuning of mapping function using GA affects the model performance. It is also observed that the difference in test times is small. It is due to all the experiments are carried out in high configuration system. In general, low-power devices use 2- or 4-bit processor with low processing capability; there the difference in test times is significant.

3.4. Hardware realisation: This section introduces hardware implementation of the proposed WNN architecture based on an FPGA SPARTAN 3E tool kit (shown in Fig. 5) for disease diagnosis. Two conventional WNNs architectures namely WiSARD and VG-RAM are also realised with hardware for comparison. Three bench mark disease datasets have been used for testing. For each dataset, we have obtained the performance measure, namely power consumption, area (memory) and test time.

Area or memory is measured as number of LUTs. A LUT consists of a block of SRAM that is indexed by the LUT's inputs. The output of the LUT is whatever value is in the indexed location in its SRAM. The LUT is actually implemented using a combination of the SRAM bits and a MUX. Each LUT size is varied from 8, 16, 32 or 64 words. The power consumption denotes amount of power consumed for operations, it is measured in watts. The test time denotes the delay in the response, it is measured in seconds.

The performance measures over three datasets are averaged and normalised between 0 and 500. These results are shown in Fig. 6.

It is clear from the figure that proposed model performed better in terms of power, area and test times over standard WNNs.

The results obtained by this hardware realisation are used to verify the presented software implementation and to compare software and hardware solutions. The hardware implementations of the WNNs are done using FPGA SPARTAN 3E tool kit. It is observed from the results that memory requirement in hardware is high. It is due to as the hardware area will depend on experiments and FPGA platform. Also, time requirements are slightly higher due to delay in RAM nodes. Overall, the results produced by both software realisation using Matlab and hardware realisation using FPGA platform produced the similar responses.

4. Conclusion: In this Letter, a new architecture has been proposed for WNNs to overcome limitations of low-power wearable devices, such as memory and delay in response. The proposed architecture has been designed using variable sized RAMs (neurons) to optimise the memory usage and a modified binary TRIE data structure for reducing the test time. It also used a bio-inspired GA to improve the accuracy of WNNs by optimising the mapping function. The proposed architecture has been validated using both software and hardware realisations of standard WNNs over various categories of disease datasets. In case of MDLD and MDHD, the proposed architecture reduced the memory, test time and increased the classification accuracy as compared with standard WNNs. In case of SDHD, the proposed architecture achieved the highest accuracy but suffers from memory problem. It is due to the fact that the memory required by proposed model is a function of the number of features in the dataset and the range of values that each feature can take.

The proposed WNN also validated using five popular neural network-based classifiers. As compared to conventional neural network classifiers proposed WNN outperformed for the cases of MDLD, SDHD and MDHD.

Hence, it is concluded that the proposed architecture is a powerful part of various low-power wearable devices, such as Gluco Track [1], Dexcom G5 [2], QuantuMDx [3], Gluco Beam [4], GeneXpert [5] and so on for the solution of memory, accuracy and time issues. This results into applications of low-power wearable diagnostic devices to diagnose diseases, such as diabetes, cancer, HIV, malaria and so on.

6 References

[1] Gluco Track, http://www.glucotrack.com/, accessed: 2017-03-15
[2] Dexcom G5 mobile CGM System, https://www.dexcom.com/g5-mobile-cgm, accessed: 2017-03-15
[3] QuantuMDx-Molecular Diagnostics in Minutes, http://quantumdx.com/, accessed: 2017-03-15
[4] Gluco Beam, http://www.rspsystems.com/, accessed: 2017-03-15
[5] Low-Cost Devices for Diagnosing Diseases in Poor Countries, https://www.bbvaopenmind.com/en/low-cost-devices-for-diagnosing-diseases-in-poor-countries/, accessed: 2016-09-30
[6] Bledsoe W.W., Browning I.: 'Pattern recognition and reading by machine' (PGEC, 1959)
[7] Aleksander I., De Gregorio M., Franca F.M.G., *ET AL.*: 'A brief introduction to weightless neural systems'. ESANN, Citeseer, 2009
[8] Ludermir T.B., de Oliveira W.R.: 'Weightless neural models', *Comput. Stand. Interfaces*, 1994, **16**, (3), pp. 253–263, doi: 10.1016/ 0920-5489(94)90016-7
[9] Aleksander I., Thomas W., Bowden P.: 'WiSARD a radical step forward in image recognition', *Sens. Rev.*, 1984, **4**, (3), pp. 120–124
[10] Berger M., Forechi A., De Souza A.F., *ET AL.*: 'Traffic sign recognition with WiSARD and VG-RAM weightless neural networks', *J. Netw. Innovative Comput.*, 2013, **1**, pp. 87–98
[11] Aleksander I., Morton H.: 'An introduction to neural computing' (Chapman and Hall London, 1990), vol. 240

[12] De Gregorio M.: 'An intelligent active video surveillance system based on the integration of virtual neural sensors and BDI agents', *IEICE Trans. Inf. Syst.*, 2008, **91**, (7), pp. 1914–1921

[13] Coraggio P., De Gregorio M.: 'WiSARD and NSP for robot global localization'. Int. Work-Conf. on the Interplay between Natural and Artificial Computation, 2007, pp. 449–458

[14] Do Prado C.B., Franca F.M., Costa E., ET AL.: 'A new intelligent systems approach to 3d animation in television'. Proc. of the 6th ACM Int. Conf. on Image and Video Retrieval, 2007, pp. 117–119, doi: 10.1145/1282280.1282300

[15] Prado C.B., Franca F.M., Diacovo R., ET AL.: 'The influence of order on a large bag of words'. 2008 Eighth Int. Conf. on Intelligent Systems Design and Applications, 2008, vol. **1**, pp. 432–436, doi: 10.1109/ISDA.2008.299

[16] de Aguiar E., Forechi A., Veronese L., ET AL.: 'Compressing VG-RAM WNN memory for lightweight applications'. 2014 Int. Joint Conf. on Neural Networks (IJCNN), 2014, pp. 1063–1070, doi: 10.1109/IJCNN.2014.6889563

[17] De Souza A.F., Pedroni F., Oliveira E., ET AL.: 'Automated multilabel text categorization with vgram weightless neural networks', *Neurocomputing*, 2009, **72**, (10), pp. 2209–2217, doi: 10.1016/j.neucom.2008.06.028

[18] Austin J.: 'RAM-based neural networks' (World Scientific, 1998), vol. 9

[19] Willard D.E.: 'New TRIE data structures which support very fast search operations', *J. Comput. Syst. Sci.*, 1984, **28**, (3), pp. 379–394, doi: 10.1016/0022-0000(84)90020-5

[20] Bodon F., Ronyai L.: 'TRIE: an alternative data structure for data mining algorithms', *Math. Comput. Model.*, 2003, **38**, (7), pp. 739–751, doi: 10.1016/0895-7177(03)90058-6

[21] Genetic algorithm code for combinatorial optimization problem, https://in.mathworks.com/help/optim/ug/travelling-salesman-problem.html, accessed: 2016-09-30

[22] Cheruku R., Edla D.R., Kuppili V.: 'Diabetes classification using radial basis function network by combining cluster validity index and bat optimization with novel fitness function', *Int. J. Comput. Intell. Syst.*, 2017, **10**, (1), pp. 247–265, doi: 10.2991/ijcis.2017.10.1.17.19

[23] Cheruku R., Edla D.R., Kuppili V.: 'SM – Ruleminer: Spider monkey based rule miner using novel fitness function for diabetes classification', *Comput. Biol. Med.*, 2017, **81**, pp. 79–92, doi: 10.1016/j.compbiomed.2016.12.009

[24] Forechi A., De Souza A.F., de Oliveira Neto J., ET AL.: 'Fat-fast VG-RAM WNN: a high performance approach', *Neurocomputing*, 2016, **183**, pp. 56–69, doi: 10.1016/j.neucom.330 2015.06.104

[25] UCI Machine Leaning Repository, https://archive.ics.uci.edu/ml/datasets.html, accessed: 2016-09-30

[26] Fushiki T.: 'Estimation of prediction error by using k-fold cross-validation', *Stat. Comput.*, 2011, **21**, (2), pp. 137–146, doi: 10.1007/335 s11222-009-9153-8

[27] Haykin S.S.: 'Neural networks and learning machines' (Pearson, Upper Saddle River, NJ, USA, 2009), vol. 3

[28] Specht D.F.: 'Probabilistic neural networks', *Neural Netw.*, 1990, **3**, (1), pp. 109–118, doi: 10.1016/0893-6080(90)90049-Q

[29] Waibel A., Hanazawa T., Hinton G., ET AL.: 'Phoneme recognition using time delay neural networks', *IEEE Trans. Acoustics Speech Signal Process.*, 1989, **37**, (3), pp. 328–339, doi:10.1109/29.21701