IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

Multivariate SoftMax Regressed Deep Spiking Neural Network Classifier for Energy Efficient VLSI Circuit Design

Journal:	*Transactions on Computer-Aided Design of Integrated Circuits and Systems*
Manuscript ID	TCAD-2023-0089
Manuscript Type:	New Research Article
Date Submitted by the Author:	22-Feb-2023
Complete List of Authors:	Raveendran, Arun Prasath; Siddhartha Institute of Technology & Sciences Sekaran , Ramesh ; Jain University Parasuraman, Manikandan; Jain University Ramachandran, Manikandan; SASTRA Deemed University, Gandomi, Amir ; University of Technology Sydney
Keywords:	3D floorplanning, floorplanning, timing verification, timing optimization
Manuscript Subject:	Machine Learning/AI Automation

Multivariate SoftMax Regressed Deep Spiking Neural Network Classifier for Energy Efficient VLSI Circuit Design

Arun Prasath Raveendran, Ramesh Sekaran, Manikandan Parasuraman, Manikandan Ramachandran, Amir H Gandomi, *Senior Member, IEEE*

***Abstract*— The design of Very Large-Scale Integration (VLSI) architecture is essential to reduce power utilization, area, and speedup operating frequency to achieve a higher integrated circuit performance. However, modern VLSI circuits face challenges with higher energy consumption with higher cost and time. To address such problems, an Energy-Efficient Multivariate SoftMax Regressed Deep Spiking Neural Network (EEMSRDSNN) model is introduced in this work for VLSI architecture design with higher accuracy and less energy consumption. The input circuits are collected from the data base in the input layer, and then functional units are sent from each circuit to hidden layer1. The suitable functional unit is selected in hidden layer1 for performing the multivariate SoftMax regression analysis. Subsequently, the selected functional units are transmitted to hidden layer2 a long with the weight. By using a time-driven swap binding process, the selected functional unit gets bonded in hidden layer2 to form an efficient VLSI architecture that is ultimately transmitted to the output layer. The performance of the EEMSRDSNN model was examined using the ISCAS'89 Benchmark Dataset. The simulation results show that the EEMSRDSNN model increased the performance accuracy of VLSI design with minimal energy consumption.**

***Index Terms*—Functional unit, Integrated circuits, Multivariate SoftMax Regression, Time-driven swap binding,VLSI Design**

I. INTRODUCTION

VERY Large-Scale Integration (VLSI) technology plays an essential role in expanding the high-tech electronic circuits. VLSI design present small size, less cost, less power consumption, high reliability, and good functionality. VLSI is aimed at the physical planning of integrated circuits, for which low-power designs are critical for the technology development of modern VLSI circuits in portable devices and medical instruments several works have implemented spiking neural networks (SNNs) in VLSI designs for information processing, whereas parse binary signals are processed in a parallel fashion. A novel MRF-based method was recently proposed in [1] for designing efficient and reliable low-power VLSI circuits with considerably low cost, delay, and power consumption. Monte Carlo analysis was performed to measure the voltage and temperature variations in the proposed MRF-based method, but the results showed that the VLSI design accuracy was not enhanced. In addition, to reduce the delay time, a new supervised learning algorithm was proposed introduced in [2] for SNNs, in which a spiking neuron was introduced for VLSI execution with analog resistive memory for attaining higher energy efficiency. The designed SNN was used to perform the time-series data for improving the learning performance. Though the energy consumption was minimized, the computational cost was not reduced by the supervised learning algorithm. To solve the above-mentioned issues, the presented work employed multivariate SoftMax regression analysis and a deep spike neural network to prepare an Energy-Efficient Multivariate SoftMax Regressed Deep Spiking Neural Network (EEMSRDSNN) model for analyzing energy consumption. In the proposed model, functional units are gathered from the database, sent to the input layer, and then to hidden layer1. The multivariate SoftMax regression analysis is used in hidden layer1 to select a suitable functional unit, which subsequently gets bonded in hidden layer2 with the aid of a time-driven swap binding process to prepare an efficient VLSI architecture. The final VLSI architecture design is sent to the output layer. This process helps to increase the VLSI design accuracy and reduce the processing time.

A. Novel Contributions

The novel contributions of the EEMSRDSNN model are described as follows:

• The EEMSRDSNN model is introduced to improve the performance of VLSI circuit design with less energy consumption and processing time.

• In the EEMSRDSNN model, the functional units are collected from the input database in the input layer. The novelty of multivariate SoftMax regression analysis is that it can select the suitable functional unit in hidden layer1 for efficient VLSI circuit design with numerous functional units. The multivariate SoftMax regression analysis is performed on dependent and independent data, which yields either '0' or '1'.

Arun Prasath Raveendran is with the Department of Electronics and Communication Engineering, Siddhartha Institute of Technology and Sciences, Narapally, Ghatkesar, Hyderabad, Telangana - 500088, India.(e-mail: prasath2k6@gmail.com).

Ramesh Sekaran is with the Department of Computer Science and Engineering, JAIN (Deemed to be University), Bengaluru - 562112, Karnataka, India. (e-mail: sramsaran1989@gmail.com).

Manikandan Parasuraman is with the Department of Computer Science and Engineering, JAIN (Deemed to be University), Bengaluru - 562112, Karnataka, India. (e-mail: mani.p.mk@gmail.com).

Manikandan Ramachandran is with the school of Computing, SASTRA Deemed University, Thanjavur – 613401, Tamil Nadu, India.(e-mail: srmanimt75@gmail.com).

Amir H Gandomi is with the Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. (e-mail:gandomi@uts.edu.au). *Corresponding author (Amir H Gandomi)*

If the value is '1', the tree creates the decision with the functional unit. Otherwise, the tree does not consider the functional unit. The selected functional units are transmitted to hidden layer2 along with the weight.

• The functional unit gets bonded in hidden layer2 based on the weight by using a time-driven swap binding process to form an efficient VLSI architecture. Finally, the VLSI architecture design is transmitted to the output layer with higher performance results compared to conventional methods. This helps to improve the VLSI design accuracy and reduce the processing time.

The rest of the paper is structured in the following manner. Section2 reviews the related works. Section3 gives a brief description of the EEMSRDSNN model with a diagram. Simulation settings and analysis results of different parameters are demonstrated in tables and graphical representations in Section4. Finally, the conclusion of the research work is given in Section5.

II. RELATED WORKS

An Adaptive Rood Pattern Search (ARPS) algorithm was introduced in [3] with limited hardware resources for sending HD videos. The design eliminates systolic arrays and constructs pattern generation. A near Sum of Absolute Difference (SAD) technique is implemented to minimize the clock cycles for discovering an MV for a macro-block, and interleaved memory organization is applied to improve throughput. However, the power consumption was not reduced by the ARPS algorithm. Satin bowerbird optimization (SBO) algorithm was partitioned in [4] randomly with fitness value assessment. The group migration was used for enhancing the cut cost, while different bio-inspired heuristic algorithms, including ant colony optimization, particle swarm optimization, genetic algorithm, and firefly algorithm, were utilized for achieving the partitioning. Despite the advancement, energy consumption was not minimized.

A hardware-efficient SVM learning unit was designed in [5] with multiplications by approximate computing techniques and employed for building dual-core, quad-core, and octa-core cascade SVM systems. The runtime and energy efficiency of cascade SVM systems was enhanced due to the numerous cores, yet the computational cost was not reduced. To attain optimum power, speed, and area, an efficient VLSI architecture was proposed in [6] for fast motion estimation in video codec via a pattern search process. The designed architecture employs an interleaved memory arrangement and a check technique to determine the sum of absolute differences. Unfortunately, the VLSI architecture could not minimize the computational cost in the hardware area.

Advanced neural network (NN) and machine learning algorithm introduced in [7], which incorporates hardware design techniques to develop the modification properties. However, the delay time was not minimized by the designed neuromorphic learning algorithm. Subsequently, a new static fault-tolerant method proposed in [8] offers redundancy at the circuit level with a low failure rateincluding transistor method to improve reliability. Overall, the designed approach reduced the area, delay, and power overhead.

An operand reduction technique was developed in [9] to minimize the radix-r butterfly units. The merged-bank memory structure was constructed with a number-theoretic transform for area reduction, and a memory addressing scheme was introduced to resolve the computation problems. However, the processing time was not reduced by the designed technique. Later, a functionality-sharing technique was introduced in [10] with resources to prepare the Arithmetic Logic Unit (ALU). The functionality was predictable through the resources with different inputs and control circuits, yet energy consumption was not taken into consideration.

Layout-aware binding algorithm was introduced in [11] for design space examination in high-level synthesis, which combines resource binding tasks and module allotment through silicon following a minimization process. Elmore delay models were introduced to achieve the net topologies in 3-D floorplans for calculating exact individual net delays. However, the computational complexity was not reduced by the designed algorithm. Moreover, an asynchronous technique was introduced in [12], which minimized the power consumption of the Viterbi encoder for different constraint lengths but failed to reduce computational costs.

An advanced encryption standard (AES) algorithm was proposed in [13] for cryptographic applications to achieve low power and high throughput. The power utilization and delay were minimized through the designed algorithm. While the designed AES algorithm showed encryption and decryption support and minimized power utilization and delay, the VLSI design accuracy was not improved. The VLSI architecture was first constructed in [14] for discrete wavelet transforms (DWT). The architecture used an area-efficient lifting technique to operate the processing elements, including one floating-point adder and one fused multiply-add design. Although both low and high pass filters were used, the computational cost was not minimized.

To address the issue, a distributed arithmetic architecture was subsequently designed in [15] for reconfigurable block-based FIR filters with large block sizes and filter lengths. However, the number of registers was not increased proportionately with the block size, and the processing time consumption was not minimized by the distributed arithmetic architecture. VLSI design architecture was presented in [16] with itsconfigurable parameters, where in the high throughput was achieved by an expandable data busline. Yet, the design complexity was not minimized by VLSI design architecture, and the optimization algorithm was not used to select the best design parameters.

Deep convolutional neural network (CNN) inference processor was developed in [17] with Enhanced Output Stationary (EOS) data flow to improve accuracy, using register files (RFs) to reprocess activation data and removing memory access. Processing Elements (PEs) were divided into several groups for enhancing the Activation RFs (ARFs). However, the energy consumption was not minimized by using EOS dataflow. A machine learning (ML)-based motion estimation (ME) controller algorithm was developed in [18] with minimal cost for intelligent mobile systems. The hardware-friendly and power-efficient VLSI architecture was

able to achieve a high-performance and low-power ME controller design, but the processing time was not minimized.

Another machine learning method to address the issue, namely a convolutive blind source separation (CBSS) network, was constructed in [19] using an information maximization method. The designed CBSS includes both Info max filtering and scaling factor computation modules. However, the VLSI design accuracy was not improved by the CBSS network. To manage the design complexity in VLSI, a placement approach with many highest standard cells was proposed in [20] but failed to consider energy consumption. In addition, a novel personalized diagnostic system called 2WKNN was presented in [21] for the optimal selection of neighboring samples for an individual. However, the performance of time consumption was not minimized.

In this section, many application-specific VLSI systems are mentioned. These literatures include several issues such as poor VLSI design accuracy and high complexity, energy consumption, processing time, power consumption, and computational costs. To address these issues, new Deep SNNs has developed to enhance the energy efficiency of deep neural networks during data-driven event-based computation. An elaborate description is provided in the next section.

III. METHODOLOGY

The Energy-Efficient Multivariate Soft Max Regressed Deep Spiking Neural Network (EEMSRDSNN) model is introduced for VLSI design architecture with minimal cost consumption and space complexity. The architecture diagram of the EEMSRDSNN model is illustrated in Figure1.

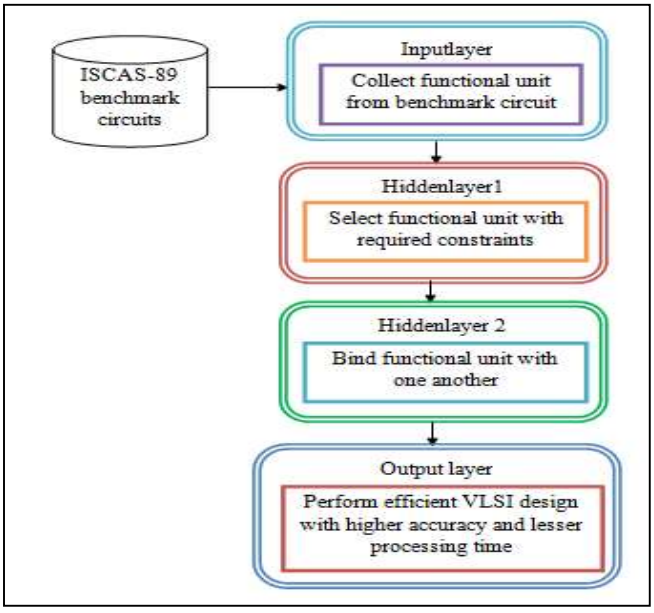

Fig. 1. Architecture Diagram of EEMSRDSNN Model

In Figure 1, initially, ISCAS-89 benchmark circuits are considered as an input. After that, the functional units are collected from the input database in the input layer. Then, the suitable functional unit is selected through multivariate SoftMax regression analysis for efficient VLSI circuit design. Later, the selected functional unit gets bonded to form an efficient VLSI architecture. Finally, the VLSI architecture design is transmitted to the output layer. A brief description of the EEMSRDSNN is described below section.

A. Spiking Neural Network

Spiking neural networks (SNNs) are more energy efficient than CNN. CNN in interference time because they utilize matrix addition instead of multiplication. SNNs are supported by new computing paradigms and hardware. SNN is used to edge computing, robotics, and other fields for building low power intelligent systems. SNNs are artificial neural networks that more closely mimic natural neural networks. In addition to neuronal and synaptic state, SNNs incorporate the time concept into their operating model. The neurons in SNN do not transmit information at each propagation cycle but transmit information when a membrane potential reaches specific threshold value. When membrane potential reaches the threshold, the neuron fires and generates the signal that travels to other neurons to increase or decrease their potentials in response to this signal. A neuron model that fires at the moment of threshold crossing is called as a spiking neuron model.SNN is a two-layered feed-forward network with lateral connections in second hidden layer. SNN is heterogeneous in nature.

SNNs were assumed to have other possible than traditional ANNs in some aspects. Initially, SNN can outperform that of an ANN based on the technique used to train the input data into spikes and the number of neurons that require producing spikes. Motivated by, propose a novel SNNs is presented to accomplish maximum algorithmic performance and energy efficiency. It is also possible to implement our algorithm in VLSI circuits with analog resistive memory. SNNs are includes of spiking neuron models that help VLSI implementations for enabling SNNs to execute tasks by less spikes than SNNs. We also propose novel deep learning techniques to develop the learning performance.

B. Multivariate SoftMax Regression

Softmax regression is a generalization of logistic regression to handle multiple classes. Softmax Regression algorithm applies binary logistic regression to multiple classes at once. The weights (θ) were omitted from the diagram for clarity, but they are computed for each attribute-to-class mapping. Softmax Regression varies from multiple binary logistic regression classifiers. Those classifiers are used with multiple classes when objects belong to more than one class. It computes membership in a given class independently from membership in other classes. Softmax Regression computes the activation values for each output class. It normalizes the values to attain the set of probabilities that sum to 1. The class with highest probability wins, and the data item is assigned to that class. An ideal scenario is if one class has a probability close to 1 and the others are close to 0.

C. Energy-Efficient Multivariate SoftMax Regressed Deep Spiking Neural Network

SNNs are employed for implementing VLSI circuit design to achieve minimum energy efficiency. Compared to NN,

SNNs are advantages such as the ability to model dynamics modes of network operations and computing in continuous real-time considered.The main advantages of SNN are ease of use in neural interface circuits, lower power consumption, and minimized computation complexity. SNNs have shown to be promising for generating machine learning algorithms with maximum energy efficiency in VLSI circuits. As the third generation of ANNs, SNNs have the advantage of attaining better energy efficiency by incorporating brain-inspired techniques to process information in spikes [23]. Temporal coding allows SNNs to perform tasks with fewer spikes. Spike Propagation [22] is the pioneer method that defines the computational error based on the distance between the actual and target firing time to reduce error. SpikeProp allows the multilayer SNNs to learn the temporal coding and differentiates the spike timing regarding the weights. However, the design process consumes a large amount of time and energy with higher risk, thus energy efficiency is a critical design parameter in VLSI systems, ranging from lowto high-performance servers. In this paper, we propose a novel EEMSRDSNN Model for SNNs based on temporal coding. The contribution of deep spiking neurons has been employed to make easy VLSI implementations by resistive memory for obtaining energy efficiency. The proposed EEMSRDSNN model is designed with the novelty of Multivariate softmax regression analysis, Time driven swap binding process. As mentioned, EEMSRDSNN model, which is proposed for constructing an efficient VLSI design architecture, is comprised of one input layer, more than one hidden layer, and one output layer. In the input layer, every neuron spike at a time is proportional to the brightness of the individual functional unit. In design tasks, the output of the network is signified through the first neuron to a spike in the output layer. The input given for SNN at time 't' is denoted as 'X(t)'; the hidden layer is symbolized as 'H(t)'; and the output layer is denoted as 'Y(t)'. Figure 2 describes the different layers of the deep spike neural network.

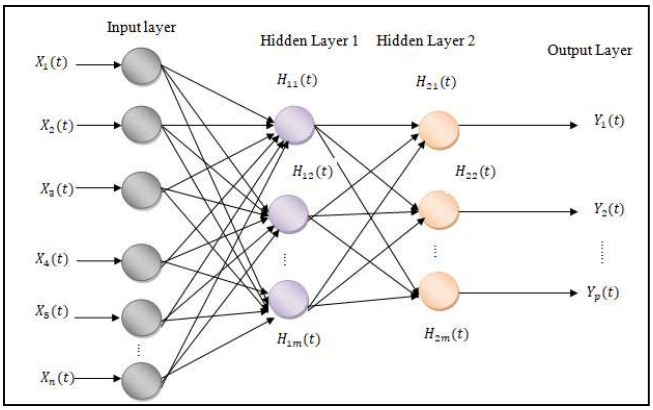

Fig. 2. Different Layers of Deep Spike Neural Network

In Figure 2, the network functions by an alpha synaptic transfer function. In SNNs, spikes are stereo typical events, and information processing is minimized by two factors: timing of spikes and identity of synapses. Spiking neurons communicate through all the discrete spikes. In the EEMSRDSNN model, the spike generates the trace of synaptic current in the target neuron. The target neuron joins the synaptic current overtime until the threshold is attained and emitted through the spike.Feedforward process is used to train SNNs directly to attain higher performance. SNN removed the complexities of trainable parameters through multiple layers to minimize the overfitting risks. Backpropagation method is used for fine-tuning the weights of a neural net based on the error rate obtained in the previous epoch (i.e., iteration). Proper tuning of weights reduces error rate and increases the generalization.In EEMSRDSNN model, two hidden layer network architecture is used for efficient SNN design. Two hidden layers are used for performing function unit selection and for binding the functional unit respectively.

In the EEMSRDSNN model, the number of functional blocks in the input layer is the input. After that, numerous functional units are sent to hidden layer1, where multivariate SoftMax regression is used to prepare an efficient VLSI processor design by selecting a suitable functional unit from the benchmark circuit. The key advantage of using Multivariate Softmax Regression is the output probabilities range of many classes. The range varies from 0 to 1. The sum of all probabilities is equal to one. When the softmax function used for multi-classification model, it returns the probabilities of each class and the target class has high probability. In EEMSRDSNN model, functional unit selection is carried out based on the probability. Multivariate SoftMax regression analysis is performed between the dependent and independent data, yielding two outcomes: '0'and '1'. Multivariate SoftMax regression uses multiple independent objectives (i.e., size, energy, delay, and complexity) of functional units in VLSI circuits for the processor design. In this regression, the decision tree is constructed with a root node, internal node, and leaf node, where the root node executes the test depending on the connection measure. The internal node denotes the test results, and the leaf node presents the output class label. Here, consider the benchmark circuit as an input. Multivariate Soft Max regression analysis with benchmark circuit is presented in Figure3.

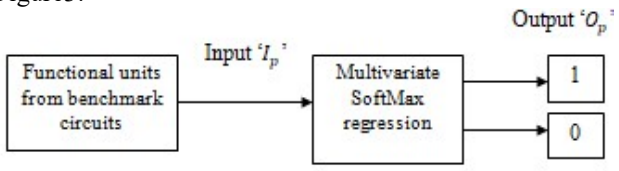

Fig. 3. Multivariate SoftMax Regression

Figure 3 describes the multivariate Soft Max regression analysis with a number of functional units, which is executed using the following equation:

$$MSR(B_f) = \frac{1}{1+exp\,(-\alpha_0+\alpha_1\theta)} \quad (1)$$

Where 'MSR(B_f)' represents the multivariate SoftMax regression output; 'α' is the regression coefficient; and 'α_0'is

considered zero; and 'θ' represents the explanatory variable. When the output of the regression tree is 0, then it is computed as $1 - \mathrm{MSR}(\mathrm{B_i})$. Depending on the regression analysis, the tree makes the decision to take the functional unit when the output is '1'under the power, voltage, and temperature variations. Otherwise, the tree does not consider the functional unit for efficient VLSI design. After that, the selected functional units are sent to hidden layer2 through spikes. The membrane potential time progression of the spiking neuron in the EEMSRDSNN model is formulated as:

$$\frac{dv_{m(i)}{}^{j}(t)}{dx} = \sum_i (w_{ji}\, MSR(B_f)) \sum_r (\varphi(t - t_i{}^r)) \qquad (2)$$

Where '$\mathrm{v_{m(i)}{}^{j}}$' denotes the membrane potential of neuron 'i' in $\mathrm{l^{th}}$ layer; '$\sum_i \mathrm{w_{ji}} \sum_r \varphi(\mathrm{t} - \mathrm{t_i^r})$' is the synaptic current; '$\mathrm{w_{ji}}$' symbolizes the weight of the synaptic connection from neuron 'i' to neuron 'j'; '$\mathrm{t_i^r}$ ' represents the time of the $\mathrm{r^{th}}$ spike from neuron 'i'; and 'φ' indicates the synaptic current kernel function, which is given by:

$$\varphi(x) = \vartheta(x) exp\left(-\frac{x}{\tau_s}\right) \qquad (3)$$

$$\vartheta(x) = \begin{cases} 1 & if x \geq 0 \\ 0 & Otherwise \end{cases} \qquad (4)$$

In Eqs. (3) and (4), the synaptic current jumps immediately on the arrival of the input spike and decays with a time constant, where 'τ_s' represent the time constant and is set as '1'. The neuron spikes one time when the membrane potential crosses the firing threshold. The membrane potential changes to zero after the spike occurrence and then drops below zero when the integral of the synaptic current is negative. Let us consider a neuron that collects 'n' spikes at different times $\{t_1, t_2, t_3 \ldots t_N\}$ with varying weights $\{w_1, w_2, w_3 \ldots w_N\}$. Each weight is considered positive or negative, and the neuron gets a spike at the time 't_o'. The membrane potential for $t < t_o$ is then formulated as:

$$v_{m(i)}(t) = \sum_{i=1}^{N} \quad \vartheta(t - t_i) w_i (1 - exp(-(t - t_i))) \qquad (5)$$

An input spike has indices in $ID \subseteq \{1, \ldots, N\}$ before t_o. The sum of weights of the input spike must be larger than 1, then t_o is defined by:

$$F_T = \sum_{i \in ID} w_i (1 - exp(-(t_o - t_i))) \qquad (6)$$

$$\exp(t_o) = \frac{\sum_{i \in ID} \; w_i exp(t_i)}{\sum_{i \in ID} w_i - 1} \qquad (7)$$

where spike time is in exponential form. A transformation of the variable '$\exp(\mathrm{t_i}) \rightarrow \mathrm{Z_i}$' connects the input spike time to the output neuron in the post-transformation domain, which is formulated as:

$$Z_o = \frac{\sum_{i \in ID} \; w_i Z_i}{\sum_{i \in ID} \; w_i - 1} \qquad (8)$$

Where '$\sum_{i \in ID} w_i$' has to be greater than '1' to attain 'Z_o' as a positive result. The output spike time is larger than the input spike time in the causal set from the causal set. A similar expression relating the time of the Q^{th} spike of output neuron '$Z_o{}^{Q}$' to input spike time in the z-domain is obtained by:

$$Z_o{}^{Q} = \frac{\sum_{i \in ID^Q} \; w_i Z_i}{\sum_{i \in ID^Q} w_i - Q} \qquad (9)$$

Where, 'ID^Q' represent set of indices of the input spike that arrives before the 'Q^{th}'output spike. Every neuron gets a spike for every input presentation to formulate the spiking activity sparsely and to make optimal use of each spike in hidden layer1.Then, the selected functional unit is transmitted to hidden layer2, in which a time-driven swap binding process is carried out for optimal module binding. The time-driven swap binding process performs the search with the sequence of moves that perturb the resource bindings. The design process conducts the neighborhood search forbinding space for the VLSI design. A local search is carried out with a combination of binding variations that improve the design performance. After that, the bonded circuits are sent to the output layer, where efficient LSI circuits are designed. The algorithmic description of the Energy-Efficient Multivariate Soft Max Regressed Deep Spiking Neural Network is given below.

Algorithm1 Energy-Efficient Multivariate SoftMax Regressed Deep Spiking Neural Network Algorithm

Input: Benchmark circuits

Output: Improved VLSI design accuracy and reduced processing time

\\Energy-Efficient Multivariate SoftMax Regressed Deep Spiking Neural Network

Step1:Begin

Step2: Benchmark circuit with functional units

Step3:While (termination criteria reached) **do**

Step4: For each benchmark circuit at the input layer

Step5: The input layer transmits the functional units to hiddenlayer1

Step6: Hiddenlayer1 analyses the functional unit by multivariate SoftMax regression and transmit selected functional unit to hiddenlayer2

Step7: Hiddenlayer2 computes the weight of each function block and performs time-driven swap binding process

Step8: The output layer generates result '$y(t)$'

Step9:end for

Step10:end

Algorithm1 describes the proposed EEMSRDSNN to attain higher VLSI design accuracy, which is used to design energy-efficient VLSI circuits with minimum cost.

IV. Simulation Setting and Result Analysis

The EEMSRDSNN model was evaluated in MATLAB Simulink with 3.4 GHz Intel Corei3 processor, 4 GB RAM, and Windows 7 platform for efficient VLSI circuit design. Experiments were conductedusing ISCAS-89 benchmark circuits, which include four inputs, one output, three D-type

flip-flops, two inverters, and eight gates (one AND + one NAND + two OR + four NOR). The benchmark circuits are four inputs depends hardware circuit's integrated circuits such as AND + NAND + OR + NOR for efficient VLSI circuit design.The dataset is divided into two sets such as training and testing. Most of the data is used for training (70%) and less data is used for testing (30%).Then, the EEMSRDSNN model was compared with the existing Markov Random Field (MRF) method [1] and supervised learning algorithm [2]. The efficiency of the EEMSRDSNN model was computed in terms of metrics, such as energy consumption, design efficiency, and processing time.

A. Impact on Energy Consumption

Energy consumption, or the amount of energy consumed by the designed algorithm for efficient VLSI design with numerous functional units, is measured in joules (J) and formulated as:

$$Energy\ Consumption = \quad N * \quad Energy\ consumed\ by\ one\ functional\ unit \qquad (10)$$

From (10), 'N' denotes the number of functional units. When the energy consumption is lesser, the method is said to be more efficient.

TABLE I
TABULATION FOR ENERGY CONSUMPTION

Benchmark Dataset Name	Energy Consumption (J)		
	MRF Method	***Supervised Learning Algorithm***	*EEMSRDSNN Model*
s953	34	52	23
s832	32	49	20
s1196	36	53	25
s510	39	57	28
s1238	30	46	18
s1423	33	50	21
s1488	31	48	19
s5378	35	51	27
s9234	34	52	26
s13207	32	48	20

Table 1 describes the impact of energy consumption for the efficient VLSI design with respect to different benchmark datasets by three methods, namely the MRF method and the EEMSRDSNN model. As shown in the table, the proposed EEMSRDSNN model consumed 21J of energy for s1423 Benchmark Circuit, whereas the MRF method [1] and supervised learning algorithm [2] consumed 33J and 50J, respectively. The energy consumption also varied for different benchmark circuits, as listed in the table.

As mentioned, the deep spike neural network includes three layers, namely the input, hidden, and output layers. The input layer takes the input of the functional unit, and every input functional unit is transmitted to hidden layer1. Then, the functional unit is analyzed in hidden layer1 by using multivariate SoftMax regression analysis under power, voltage, and temperature variations. Subsequently, hidden layer1 transmits the analyzed results with their weights to hidden layer2. In the latter layer, functional units get bonded with each other, and then VLSI circuits get designed in the output layer with higher performance results. This in turn helps to reduce energy consumption.

Let us consider ten different benchmark circuits for achieving the energy-efficient VLSI circuit design, where energy consumption changes in all three techniques in each circuit. Multivariate SoftMax regression analysis selects the suitable functional unit for preparing the efficient VLSI design. Results show that the energy consumption of the EEMSRDSNN model was reduced by 33% and 55% compared to the existing MRF method [1] and Supervised learning algorithm [2] respectively.

B. Impact on Processing Time

Processing time is the amount of time (ms) consumed for efficient VLSI design with the numerous functional units, which is calculated by:

$$Processing\ Time = \quad Ending\ time - Starting\ time\ of\ VLSI\ design \qquad (11)$$

When the processing time is less, the method is said to be more efficient.

The results of the 10 benchmark circuits reveal that the processing time varies each circuit, for which the graphical representation is illustrated in Figure 4.

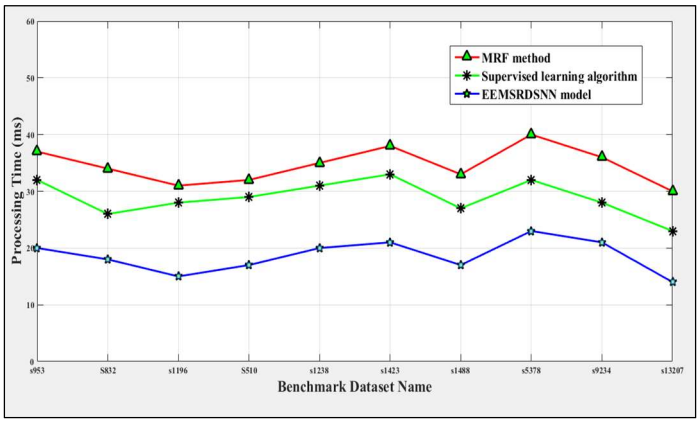

Fig. 4. Comparison of Processing Time using the Proposed Model, MRF Method [1] and Supervised learning algorithm [2]

Figure 4 compares the processing time performance for diverse benchmark circuits using the three methods, where different benchmark circuits are given on in the x-axis and the output of processing time is on, they-axis. In the above figure, the blue line denotes the processing time consumption performance of the EEMSRDSNN model, while the red and green lines represent the MRF method [1] and Supervised learning algorithm [2] respectively. Specifically, for the s510 Benchmark Circuit, the processing time of the EEMSRDSNN model was found to be 17 ms, while those of the MRF method

[1] and Supervised learning algorithm [2] were 32 ms and 29 ms, respectively. It is observed that processing time consumption using the EEMSRDSNN model is comparatively less than the times of the existing MRF method [1] and Supervised learning algorithm [2]. This is attributed to the application of multivariate SoftMax regression analysis in the EEMSRDSNN model.

In the EEMSRDSNN model, the benchmark circuit is considered as an input with numerous functional units, which are sent to hidden layer1. In this layer, the functional units are analyzed by multivariate SoftMax regression analysis under power, voltage, and temperature variations. Then, results with their weights are transmitted to hidden layer2. The functional units get bonded with one another, and VLSI circuits are designed in the output layer, which in turn helps to minimize the time consumption. Considering the ten different benchmark circuits as inputs for performing energy-efficient VLSI circuit design, the processing time was different in all three methods. Specifically, the processing time consumption of the EEMSRDSNN model was 47% and 36% less than that ofthe existing MRF method [1] and Supervised learning algorithm [2] respectively.

C. VLSI Design Accuracy

VLSI design accuracy is defined as the ratio of the number of functional units that are correctly considered to the total number of functional units. It is measured in terms of percentage (%) and calculated as:

$$VLSI\ Design\ Accuracy = \frac{Number\ of\ functional\ units\ correctly\ taken}{Total\ number\ of\ functional\ units} * 100 \qquad (12)$$

When the VLSI design accuracy is higher, the method is said to be more efficient.

TABLE II
TABULATION FOR VLSI DESIGN ACCURACY

Benchmark Dataset Name	VLSI Design Accuracy (%)		
	MRF Method	***Supervised Learning Algorithm***	***EEMSRDSNN Model***
s953	75	79	89
s832	78	85	92
s1196	80	87	94
s510	79	86	93
s1238	77	84	90
s1423	74	81	88
s1488	76	83	91
s5378	78	87	95
s9234	74	84	92
s13207	75	85	94

Table 2 demonstrates the impact of VLSI design accuracy for an efficient VLSI design with respect to different benchmark circuits by three methods, namely the MRF method [1], Supervised learning algorithm [2] and EEMSRDSNN model. As shown in the table, the proposed EEMSRDSNN model attained a higher VLSI design accuracy for efficient VLSI circuits when compared to the existing works. For the s5378 Benchmark Circuit in the first iteration, the EEMSRDSNN model attained 95%VLSI design accuracy, while the MRF method [1] and supervised learning algorithm [2] attained 78% and 87% VLSI design accuracy, respectively. Likewise, the remaining nine iterations were carried out with various ranges of different benchmark circuits. Based on this percentage improvement, it is inferred that the VLSI design accuracy using the EEMSRDSNN model is better than the MRF method [1] and supervised learning algorithm [2]. This is contributed to the multivariate SoftMax regression analysis and spike neural network that are applied in the EEMSRDSNN model.

In the EEMSRDSNN model, the benchmark circuit is taken as the input with numerous functional units and transmitted to hidden layer 1, which analyzes the functional units through multivariate SoftMax regression analysis. The results with their weights are sent to hidden layer 2, then the functional units get bonded with one another, and VLSI circuits are designed in the output layer. In this way, VLSI design accuracy is improved. For each of the 10 input benchmark circuits, the VLSI design accuracy varied for all three methods. Specifically, the VLSI design accuracy of the EEMSRDSNN model was 20% and 9% better than that of the existing MRF method [1] and supervised learning algorithm [2] respectively.

D. Power Consumption

Power consumption is the amount of energy utilized per unit time. It is formulated as,

$$PowerConsumption = N * \text{Power consumed by one functional unit} \qquad (13)$$

From (13), 'N' denotes the number of functional units. It is measured in terms of joules (J). When the power consumption is minimum, the method is said to be more efficient.

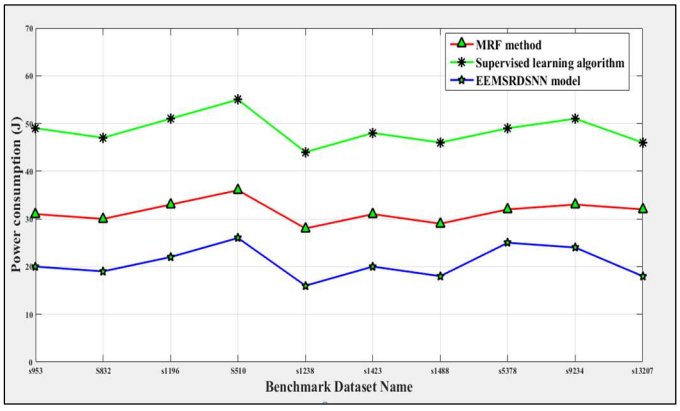

Fig. 5. Comparison of power consumption using the Proposed Model, MRF Method [1], and supervised learning algorithm [2]

Figure 5 describes the graphic representation of the power consumption performance of diverse benchmark circuits. From the above-mentioned graph, different benchmark circuits are taken in the 'X' axis while the output of power consumption is considered in the 'Y' axis. In the figure, the blue color line represents the power consumption performance of the EEMSRDSNN model whereas the red color line and the green color line represent the energy consumption performance of the MRF Method [1] and supervised learning algorithm [2]. The reason for minimizing power consumption is to apply the deep spike neural network in the EEMSRDSNN model. The power consumption of the EEMSRDSNN model is diminished by 34% and 57% compared to the existing MRF method [1] and supervised learning algorithm [2] respectively.

E. Area

Area is defined as whole cell area of VLSI intend after design energy-efficient VLSI circuits.

TABLE III
TABULATION FOR AREA

Benchmark Dataset Name	Area (μm^2)		
	MRF Method	***Supervised Learning Algorithm***	***EEMSRDSNN Model***
s953	940	1045	740
S832	920	990	720
s1196	890	960	700
S510	930	980	730
s1238	870	940	710
s1423	890	960	740
s1488	910	970	750
s5378	880	930	720
s9234	860	910	700
s13207	830	890	680

Table 3 represents the impact of area for efficient VLSI design with respect to different benchmark datasets by three methods, namely MRF Method [1], and supervised learning algorithm [2] and the EEMSRDSNN model. Compared to other existing methods, the proposed EEMSRDSNN model utilizes a minimum amount of area to form an efficient VLSI design. The area of the EEMSRDSNN model is diminished by 19% and 25% compared to the existing MRF method [1] and supervised learning algorithm [2] respectively.

V. CONCLUSION

In this work, the Energy-Efficient Multivariate SoftMax Regressed Deep Spiking Neural Network (EEMSRDSNN) Model is introduced for effective VLSI architecture design with minimal energy consumption and high design accuracy. In the model, functional units are collected from the input database, transmitted to the input layer, and then sent to hidden layer 1. The suitable functional unit is selected by multivariate SoftMax regression analysis in hidden layer 1and transmitted to hidden layer 2. The selected functional unit gets bonded by a time-driven swap binding process in hidden layer 2 to form an efficient VLSI architecture. The designed VLSI architecture is subsequently transmitted to the output layer. The performance of the EEMSRDSNN model was evaluated in terms of energy consumption, VLSI design accuracy, and processing time. Results indicate that the proposed EEMSRDSNN model constructs a more efficient VLSI design than other state-of-the-artworks, namely the MRF method and supervised learning algorithm. Particularly, the overall result of proposed EEMSRDSNN model exhibited better performance with a reduction of energy consumption, processing time, power consumption and area by 45%, 41%, 46%, 22% and improved VLSI design accuracy by 13%.

REFERENCES

[1] S.M. Razavi, and S.M. Razavi,"An efficient and reliable MRF-based methodology for designing low-power VLSI circuits",*Integration*, vol. 73, pp.77-88, 2020.

[2] Y. Sakemi, K. Morino, T. Morie, and K. Aihara, "A Supervised Learning Algorithm for Multilayer Spiking Neural Networks Based on Temporal Coding Toward Energy-Efficient VLSI Processor Design", *Neural and Evolutionary Computing*, vol. 1, Pages 1-24, January 2020.

[3] A. Mukherjee, and A.S. Dhar,"Reliable VLSI Architecture Design Using Modulo-Quad-Transistor Redundancy Method",*Circuits, Systems, and Signal Processing*, vol. 37, no. 12, pp.5595-5615, 2018.

[4] R, Pavithra Guru, and V. Vaithianathan,"An efficient VLSI circuit partitioning algorithm based on satin bowerbird optimization (SBO)", *Journal of Computational Electronics*, vol. 19, no. 3, pp.1232-1248. 2020.

[5] M. Loukrakpam, and M. Choudhury,"Hardware-Efficient VLSI Design for Cascade Support Vector Machine with On-Chip Training and Classification Capability",*Circuits, Systems, and Signal Processing*, vol. 39, no. 10, pp.5272-5297, 2020.

[6] B. Biswas, R, Mukherjee, I. Chakrabarti, P.K. Dutta and A.K. Ray,"A high-speed VLSI architecture for motion estimation using modified adaptive rood pattern search algorithm",*Circuits, Systems, and Signal Processing*, vol. 37, no. 10, pp.4548-4567, 2018.

[7] J. Park, J. Lee, and D. Jeon,"A 65-nm neuromorphic image classification processor with energy-efficient training through direct spike-only feedback",*IEEE Journal of Solid-State Circuits*, vol. 55, no. 1, pp.108-119, 2019.

[8] R. Mukherjee, B. Biswas, I. Chakrabarti, P.K Dutta, and A.K. Ray, "Efficient VLSI design of adaptive rood pattern search algorithm for motion estimation of high definition videos",*Microprocessors and Microsystems*, vol. 45, pp.105-114, 2016.

[9] J.H. Ye, and M.D. Shieh,"Low-complexity VLSI design of large integer multipliers for fully homomorphic encryption",*IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 9, pp.1727-1736, 2018.

[10] R. Nangia, and N.K Shukla,"Resource utilization optimization with design alternatives in FPGA based arithmetic logic unit architectures", *Procedia computer science*, vol. 132, pp.843-848, 2018.

[11] V. Krishnan., A Timing-driven Binding Algorithm for High-Level Synthesis of Three-dimensional Integrated Circuits", *International Journal of Computer Applications*, vol. 179, no. 49, pp.1-7, 2018.

[12] T.K. Devi, and S. Palaniappan, "An asynchronous low power and high performance VLSI architecture for Viterbi decoder implemented with quasi delay insensitive templates", *The Scientific World Journal*, pp.1-13, 2015.

[13] K. Kalaiselvi, and H. Mangalam, "Power efficient and high performance VLSI architecture for AES algorithm",*Journal of Electrical Systems and Information Technology*, vol. 2, no. 2, pp.178-183, 2015.

[14] M.M.A. Basiri, and Sk.N. Mahammad, "An efficient VLSI architecture for lifting based 1D/2D discrete wavelet transform", *Microprocessors and Microsystems*, Elsevier, vol.47, Part B, pp.404-418, 2016.

[15] B.K. Mohanty,P.K. Meher,S.K. Singhal and M.N.S. Swamy, "A high-performance VLSI architecture for reconfigurable FIR using distributed arithmetic", Integration, vol. 54, pp.37-46, 2016.

[16] J. Mitra, and T. Nayak,"Reconfigurable very high throughput low latency VLSI (FPGA) design architecture of CRC 32", *Integration*, vol. 56, pp.1-14, 2017.

[17] Y. Sakemi, K. Morino, T. Morie, and K. Aihara, "A Supervised Learning Algorithm for Multilayer Spiking Neural Networks Based on Temporal Coding Toward Energy-Efficient VLSI Processor Design", *Neural and Evolutionary Computing*, vol. 1, Pages 1-24, January 2020.

[18] J. Hsieh. and H. Wang, "VLSI Design of an ML-Based Power-Efficient Motion Estimation Controller for Intelligent Mobile Systems", *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 26, no. 2, pp.262 – 271, 2018.

[19] J.C. Wang, C.Y. Wang, T.C. Tai, M. Shih, S.C. Huang, Y.C. Chen, Y.Y. Lin. and L.X Lian, "VLSI design for convolutive blind source separation",*IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 2, pp.196-200, 2016.

[20] G. Wu, and C. Chu, "Detailed placement algorithm for VLSI design with double-row height standard cells",*IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 9, pp.1569-1573, 2015.

[21] M. Doborjeh, N. Kasabov, Z. Doborjeh, R. Enayatollah., E. Tu. and A.H. Gandomi,"Personalised modelling with spiking neural networks integrating temporal and static information",*Neural Networks*, vol. 119, pp.162-177, 2019.

[22] Xiurui Xie, Hong Qu, Guisong Liu, Malu Zhang, and Jürgen Kurths, "An Efficient Supervised Training Algorithm for Multilayer Spiking Neural Networks", *PLoS ONE*, vol. 11, no. 4, pp. 1-29, 2016.

[23] Seongsik Park, Seijoon Kim, Hyeokjun Choe and Sungroh Yoon, "Fast and Efficient Information Transmission with Burst Spikes in Deep Spiking Neural Networks", *56th ACM/IEEE Design Automation Conference (DAC*), 2019