Design of Variation Tolerant Near Threshold Processor Using Artificial Ecosystem Optimizer with Hybrid Deep Learning

[1]Raghu Gundaala and [2]Selvakumarasamy K
[1,2]Department of Electronics and Communication Engineering, Saveetha School of Engineering,
Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu, India.
[1]raghu.gundaala44@gmail.com, [2]selvakumarasamyk@gmail.com

Correspondence should be addressed to Selvakumarasamy K : selvakumarasamyk@gmail.com

Article Info
Journal of Machine and Computing (http://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202404078
Received 22 March 2024; Revised from 06 June 2024; Accepted 10 July 2024.
Available online 05 October 2024.

Abstract – Recently, several applications of data mining and pattern recognition require statistical signal processing (SP) to be used and machine learning (ML) techniques for processing massive data volumes in energy-constrained contexts. It is developing interest in executing difficult ML techniques like convolutional neural network (CNN) on lesser power embedding environments to allow on-device learning and inference. Several of these platforms are that utilized as lower power sensor nodes with lower to medium throughput conditions. Near threshold processor (NT) proposals are appropriate for these applications where as affected by a vital enhancement in variants. This research offers an Artificial Ecosystem Optimizer with Hybrid Deep Learning for Variation-Tolerant Near-Threshold Processor (AEOHDL-VTNT). The inference of embedded systems at the network edge serves as the foundation for the AEOHDL-VTNT approach that is being discussed. In the described AEOHDL-VTNT approach involves two primary processes: HDL-based VTNT design and hyper parameter tweaking. Initially, the HDL model is used to develop the VTNT. Next, in the second step, the AEO method is used for hyper parameter tweaking of the HDL model, which improves the HDL's overall performance. A number of simulations were carried out to show how the AEOHDL-VTNT approach improved performance. The simulation results showed that the AEOHDL-VTNT approach outperformed other models.

Keywords – Near-Threshold Processor, Variation-Tolerant, Deep Learning, Parameter Tuning, Artificial Ecosystem Optimizer.

I. INTRODUCTION

Recently, in-memory architectures have been suggested to directly address the cost of memory access in ML systems via embedding analog computation in closer proximity to the memory bit-cell array (BCA) which makes in-memory structures massively and intrinsically parallel and well-suited to the data flow of ML algorithm [1]. On the other hand, its inherent analogue nature makes an in-memory architecture vulnerable to process, voltage, and temperature (PVT) variation. The Delay and energy costs of modern ML algorithm inhibit their deployment for real-time inference on sensor-rich platforms namely Internet of Things (IoT) [2], wearable's, UAVs, personal biomedical devices, etc. This application involves devices that process and acquire data to derive action and interpretation for monitoring or automating different tasks without human interference [3]. Such approaches need to be understood as computationally intensive ML algorithms under rigorous constraints on form-factor, energy, and latency. As is generally known that the latency and energy costs of realizing ML algorithms are under the control of memory access [4]. In recent times, various IC implementations and energy efficient digital architectures were introduced that make an attempt to decrease the number of memory accesses through approaches like efficient data-flow, data reuse, and minimizing computations [5]. Dropping supply voltage to the near threshold voltage (NTV) area is the potential technique for accomplishing high energy effectiveness in energy-constraint circuits [6]. However, NTV operation causes additional difficulties as a result of the rising delay caused by PVT variation under the scaling voltage [7]. This challenge was precisely demonstrated as follows: (1) a five-fold increase in the rate at which memory and logic circuits malfunction, (2) over 10× loss in performance, and (3) 5× increase in performance variation. Furthermore, the PVT-induced variation affects data paths and clock signals, such the critical path fails to supply the output dataset within the provided clock period [8]. Moreover, timing error in dataset path could not be endured by concealing since delay of bit flipping would be accumulated recurrently in circuits namely the multiply-accumulate (MAC) unit in the

neural network (NN) processors. Therefore, the transmission of timing errors incurs a substantial accuracy loss, particularly in the DNN accelerator encompassing a wide variety of MACs [9]. Classical integrated circuit design avoids PVT-induced timing error by preserving timing margins and voltage as a timing guard band. But conservative guard bands lead to a decline in excessive throughput cost of energy wastage since circuit doesn't work in worst case. Timing error-tolerant technique related to error detection and correction (EDAC) circuit has developed as a potential solution [10]. This research presents an Artificial Ecosystem Optimizer with Hybrid Deep Learning for Variation-Tolerant Near-Threshold Processor (AEOHDL-VTNT). The inference of embedded systems at the network's interface serves as the foundation for the AEOHDL-VTNT approach that is being discussed. The two main procedures of the AEOHDL-VTNT approach that is being discussed are hyper parameter tweaking and HDL based VTNT design. At the initial stage, the HDL model is employed for the design of VTNT. At the first stage, the HDL model is used for the design of VTNT. Next, in the second step, the AEO method is used for hyper parameter tweaking of the HDL model, which improves the HDL's overall performance. To show the improved performance of the AEOHDL-VTNT approach, a series of simulations were conducted.

II. RELATED WORKS

Lin and Cavallaro [11] present a variation-tolerant architecture for CNN capable of operating in NTC systems for energy efficacy. Especially, the author constructs strong CNN from two lower costs untrustworthy designs that have dissimilar error information: a K-means approximated design where weight vectors in the CNN were clustered to lessen complication and NTC paradigm with full precision. Wang et al. [12] propose a 3D optimization algorithm that could efficiently recognize the system configuration to balance amongst energy, reliability, and performance. The author uses a dynamic programming system for determining the approximate level and proper voltage based on three predictors: output quality, system performance, and energy consumption. The author suggests an output quality forecast that uses collaborative design fault injection platforms in hardware or software to assess the effect of defects on output reliability under NTC.

Fan et al. [13] propose a light-weight timing resilient system to allow the NTC effectual ICs. The system that is being shown uses nine extra transistors in the node transition signal detector (NTSD) approach. A circuit that integrated using edge-triggered flip-flops was injected into the targeted ICs' monitored location. The author develops a combination selection strategy that takes failure awareness into account in order to further reduce overheads. Fan et al. [14] propose a lightweight timing error-tolerant flip-flop (ETFF) scheme. The proposal uses just nine transistors to create node transition signal detectors, which can identify timing faults and rectify them in a comparable clock cycle. Furthermore, transistor dimensions was discovered to augment the tradeoff amongst area overhead and performance. The original flip-flops are replaced at timing-monitored points in monitored circuits with the anticipated ETFF. A variation-tolerant architecture for CNN is introduced by Lin et al. [15]; it can perform robust operations in the NTV zone for conservation of energy. The main goal was to derive low-cost estimators for error identification and compensation by taking use of intrinsic redundancy in matrix-vector multiplication, often known as dot product ensemble, a power-hungry operation in CNN.

A meta stability condition detection and correction (MEDAC) approach is developed by Lin et al. [16] and is included in near-threshold voltage (NTV) network-on-chip (NoC) circuitry. As the projected method presents double-sampling-based circuitry for detecting either input data reaches closely enough to receiver clock edge that is determined as meta stability condition. Also, MEDAC has a meta stability condition mitigation system for monitoring whether the existence of meta stability condition is recurrent and tuning the receiver clock stage adaptively to rise the mean time amongst meta stability conditions. Pandey et al. [17] develop Low-power NTC TPU variant called Green TPU. Green TPU detects a pattern in the error-causing activation sequence in the systolic array and periodically increases the operating voltages associated with the multiplier-and-accumulator unit in the TPU to prevent additional timing error from the related pattern. This ensures higher inference accuracy at a lower voltage operation.

III. THE PROPOSED MODEL

During this research, we have developed a novel AEOHDL-VTNT approach via the hybridization of DL models. The presented AEOHDL-VTNT technique is primarily grounded in integrated system inference on the network's edge. The two main procedures of the AEOHDL-VTNT approach that is being discussed are hyper parameter tweaking and HDL based VTNT design.

Process involved in HDL Model

Primarily, the HDL model is employed for the design of VTNT. The HDL model is derived by the use of CNN with LSTM models. LSTM was utilized to resolve learning model for the Recurrent Neural Network (RNN) to generate potential outcomes on a different task involving construction of language and prediction model [18]. Tasks with complex time delays that the RNN algorithm was unable to solve are addressed by the LSTM. The LSTM learning model can learn long-term relationships by substituting a cell with memory in place of the RNN hidden layer.

The technique includes or excludes data to cell state named Get. The input gate (i_t), forget gate (f) and output gate (O) and it is determined by:

$$f_t=\sigma(W_f.[h_{t-1}.x_t]+b_f\cdot) \quad (1)$$

$$i_t=\sigma(W_i.[h_{t-1,x_t}]+b_i), \quad (2)$$

$$C_i=\tanh(W_c.[h_{c-1.x_t}]+b_c), \quad (3)$$

$$C_{t=}f_t*C_{t-1}+i_t*C_t, \quad (4)$$

$$O_i=\sigma(W_o.[h_{t-1.x_t}]+b_o), \quad (5)$$

$$h_{t=}O_t*\tanh*(C_t), \quad (6)$$

Combining CNN and LSTM in an efficient way was introduced for limiting the employment of CNN to get learning ability and invaluable knowledge with great efficacy of LSTM in a time-series-internal-representation-dependent way to drastically model and discover the long -and short -term temporal relationship cornified in the data. **Fig 1** shows the structure of CNN-LSTM.

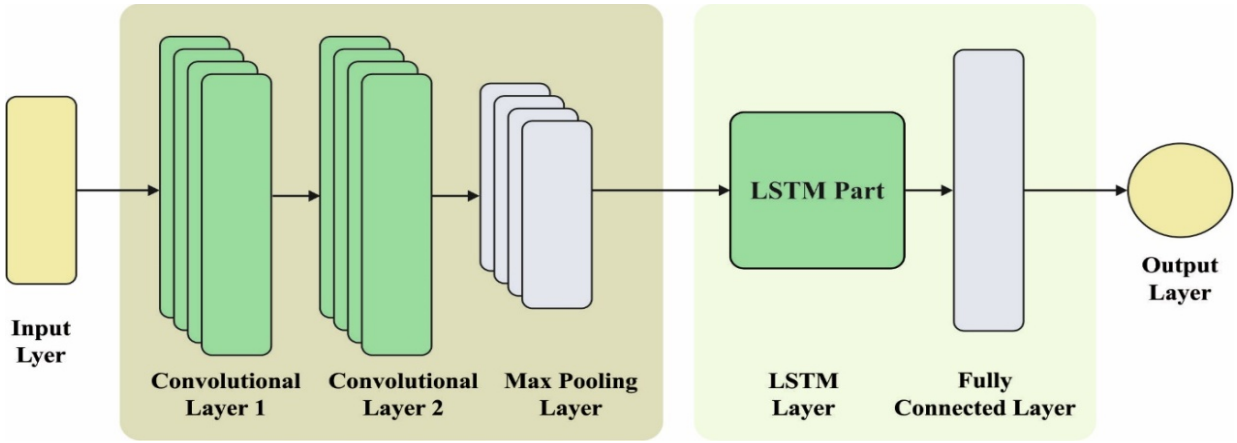

Fig 1. Structure of CNN-LSTM.

To accomplish the aforementioned objective, the presented CNN-LSTM is composed of two fundamental modules:

1D-CNN, which is composed of pooling and convolutional layers to carry out mathematical process on input dataset and thereby generate feature.

To exploit the feature, by applying dense layers and LSTM. The presented CNN-LSTM technique makes use of CNN as encoder, whereas LSTM acts as a decoder correspondingly. An encoder learned the attributes provided to the decoder (LSTM) via the source dataset. Then, decoder models and identifies each of them, could be long and short term temporal relations from data. The series of an event of all the steps are concisely discussed in the following: Source layer: receive the source dataset;1st Convolution layer: scan through input dataset of (1) above prior to subjecting the outcome on the feature map; 2nd Convolution layers: find feature map again, to improve notable features with 32 feature maps for each kernel size and convolution layer of 3 times steps to read input series would be applied; Max pooling layers: remove particular feature from (3) above so that over simplifying feature map and producing matric which has a smaller dimension; Dropout layers: improves learning network to secure the model from being over fitting; Flatten layers: fatten the distilled feature map into single long vector that is harnessed as input for decoding purposes; Repeat Vector layers: the internal demonstration of input sequence was reiterated, once for every time step in output series; LSTM decoders: has a 100-unit hidden layer can able to output the entire arrangement all of them containing 100 units which provide value everyday, which serves as foundation to forecast what would have happened in the following days in the output order; Fully connected layers: comprehend every step in output sequence to end up having same layer to predict a single sequence output indicates that LSTM decoder can able to operate at provided given time similar to fully connected and output layers; Output layer: Prediction of class label.

AEO based Hyper Parameter Tuning Process

In this study, the AEO algorithm is applied for the hyper parameter tuning of the HDL model. AEO is a kind of population-related optimized technique [19]. It imitates the processes of creation, breakdown, and utilization that occur in living things. The flow of energy through an ecosystem is described by this model. Throughout the environment, producer uses carbon dioxide, water, and sunlight for making food energy. Likewise, the production process will improve the balance between exploitation and exploration. Then, the consumer was unable to prepare their food. They acquire nutrients and energy from

other producers or consumers. The next phase will enhance the investigative process. Finally. Decomposer feeds on both consumers and producers. The phase will improve the exploitation.

Producer

In this phase, the best individual (x) and a random individual (x_{randi}) are produced at random. The decomposer (best individual) and lower and upper limitations update the producer (worse individual). It facilitates other individuals to search for separate regions and in the following expression, it is mathematically expressed.

$$x_1(ti+1)=(1-a)x_q(ti)+ax_{randi}(ti) \tag{7}$$

Whereas

$$a=\left(1-\frac{ti}{It}\right)*r_1 \tag{8}$$

$$x_{randi}=r*(Up-Lw)+Lw \tag{9}$$

$$x_2(ti+1)=x_2(ti)+K[x_2(ti)-x_1(ti)] \tag{10}$$

The equation denotes a linear weight coefficient, q indicates the population size, r_1 and r indicate a random integer that lies in [0,1], it demonstrates maximal iteration, and the Lw and Up denote the lesser and higher bounds, respectively. The purpose of the flight tax, K, is to enhance the research phase:

$$K=0.5*\frac{v_1}{|v_2|} \tag{11}$$

Where $v_2 = N(0,1)$ and $v_1 =\sim N(0,1)$. $N(0,1)$ represents uniform distribution.

Consumption

This phase improves exploration by allowing the algorithm to update individual solutions. The consumer is categorized into herbivore, omnivore and carnivore. The herbivore feed on consumer and the producer. The second feed on consumers having high energy levels. Lastly, feed on the producer and/or consumer with high energy levels. Herbivore as a consumer is mathematically expressed below:

$$x_i(ti+1)=x_i(ti)+K[x_i(ti)-x_i(ti)],\ i\in[3,\ \ldots,\ n] \tag{12}$$

Consumer as a carnivore is mathematically expressed below:

$$\begin{cases} x_i(ti+1)=x_i(ti)+K[x_i(ti)-\chi_{j(ti)}], \\ i\in[3,\ \ldots,n],\ j=randi([2i-1]) \end{cases} \tag{13}$$

Omnivore consumer can is mathematically expressed as:

$$\begin{cases} x_i(ti+1)=x_i(ti)+K[r_2(x_i(ti)-x_1(ti))+(1-r_2)(x_i(ti)-x_j(ti))], \\ i\in[3,\ \ldots,n],\ j=randi([2i-1]) \end{cases} \tag{14}$$

Where r_2 lies within [0,1] interval.

Dissection

The dissection stage is essential because it completes the food chain and feeds the producer. After the consumer in the ecosystem dies, the decomposer consumes its residues. The component was the weight coefficient (w_e and h_e) and the decomposition factor (D_e). By updating each site in accordance with a better solution, the dissection facilitates utilization in the manner described below.

$$x_i(ti+1)=x_n(ti)+D_e\left(w_ex_n(ti)-h_ex_i(tt)\right),\ i=1,\ \ldots,n \tag{15}$$

The newly introduced factor was defined by:

$$\begin{cases} D_e=3u, & u\sim N(0,1) \\ w_e=r_3*randi([1,2])-1 & x \\ h_e=2*r_3-1 & x \end{cases} \tag{16}$$

r3 is located inside the interval [0,1]. **Fig 2** depicts the steps involved in AEO technique.

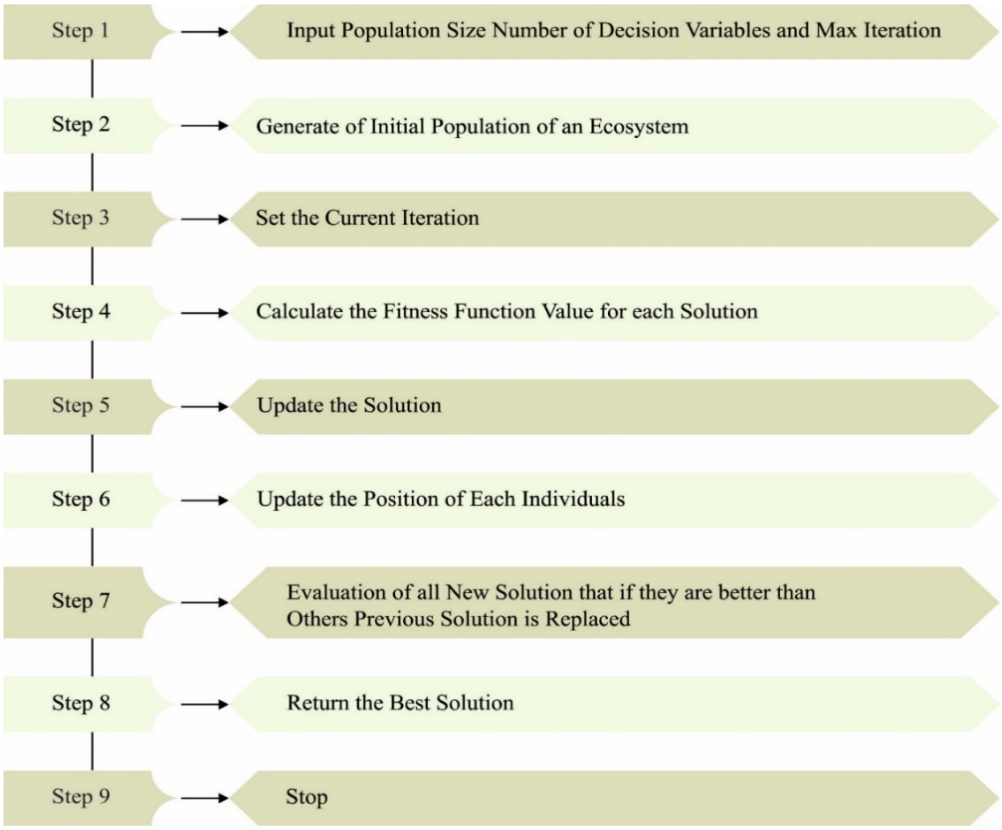

Fig 2. Steps involved in AEO.

Elimination

During this stage, x_q is adjusted once everyone has reached the desired level of physical well-being. As a result, the ending criteria are assessed; if it is satisfied, the implementation is ended and x_q is returned, otherwise, the first phase is reiterated.

Algorithm 1: Pseudocode of AEO algorithm

Initializes: arbitrary initialization of the AEO ecosystem, x_1, and assessment of fitness, f_{fi}, x_q = better solution obtained.
While ending criteria are not attained, implement:
First Phase: Production
Individualx_1, upgrade its location using (7).
Second Phase: Consumption
Individual $x_1 (i = 2, \dots, n)$
Herbivorous act takes place
If rand $< \frac{1}{3}$, individual update was performed based on (12)
Omnivorous act takes place
Else if 1/3 and rand $< 2/3$ update to individual is performed based on (14) Carnivorous act takes place
Else individual updates can be performed by (13)
End if
End if
Third phase: Decomposition
Individual updates can be performed by (15)
Individual fitness c is evaluated
Better location attained so far is upgraded, x_q
End while
Fourth Phase: Termination
Return

IV. EXPERIMENTAL VALIDATION

This section evaluates the scientific feasibility of the AEODHL-VTNT approach using eight datasets. This section uses these eight datasets to assess the scientific feasibility of the AEODHL-VTNT technique. **Table 1** and **Fig 3** show the NUTE analysis of the AEODHL-VTNT model with revised models on eight datasets. These data indicated that the AEODHL-VTNT approach produced the lowest NUTE values across all normalized frequencies. Simultaneously, it is seen that when normalized frequencies increase, the AEODHL-VTNT model produces rising NUTE values. NUTE's value is minimal at 1.0x normalized frequency and reaches its peak at 4.0x normalized frequency.

Table 1. Comparison between the AEODHL-VTNT System using Eight Different Datasets using NUTE Analysis

Total Undiscovered Timing Errors (log)								
Frequency (Normalized)	SVHN	CIFAR-10	IMDB	GTSRB	REUTERS	MNIST	FMNIST	AMNIST
4.0x	117,774,037	105,632,799	94,596,883	96,255,244	89,096,054	79,707,896	71,434,930	68,677,402
3.0x	117,216,815	102,331,440	89,642,963	97,908,853	84,130,343	76,964,337	71,984,628	65,917,946
2.4x	110,603,881	94,595,220	83,019,001	95,699,848	82,471,779	73,640,639	71,432,728	54,333,885
2.0x	97,918,998	81,360,118	75,854,855	81,912,316	71,986,905	59,851,821	68,682,035	46,074,941
1.71x	77,507,579	71,997,442	62,067,496	62,621,720	54,344,854	44,405,948	34,490,169	27,867,194
1.5x	64,819,275	51,035,649	46,616,955	48,817,148	42,206,897	30,077,901	23,462,672	20,135,554
1.33x	47,166,003	37,233,491	38,352,437	38,344,046	32,271,934	23,448,384	19,587,381	14,071,363
1.2x	35,596,783	27,857,538	22,900,250	18,483,197	21,253,810	12,424,625	18,481,688	835,718
1.09x	29,528,326	14,629,984	1,397,314	2,486,140	7,459,427	8,052	8,230	5,003
1.0x	9,658,840	9,130	6,284	6,308	8,703	8,447	7,360	8,159

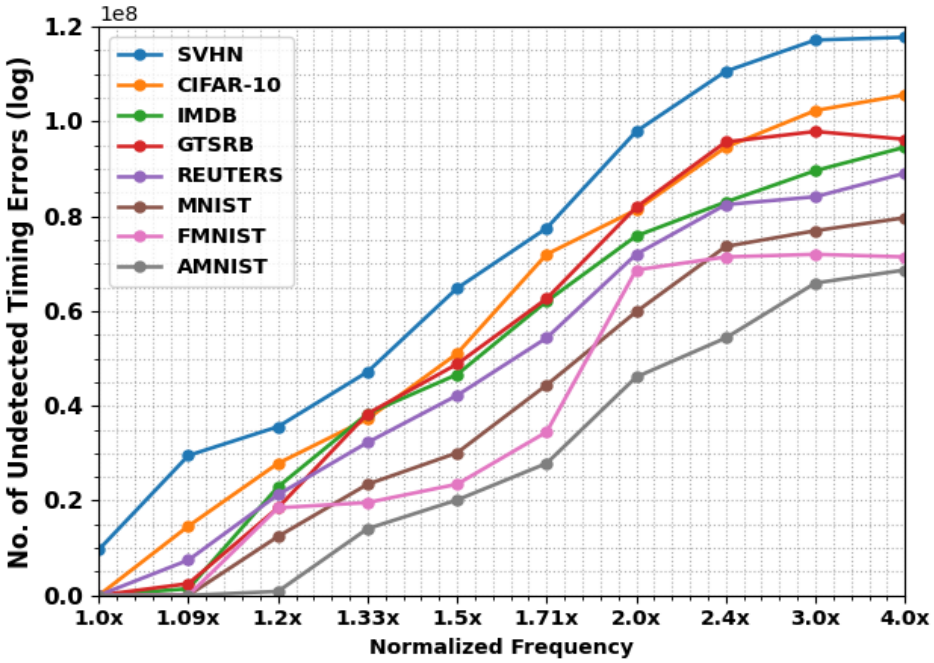

Fig 3. NUTE Examines AEODHL-VTNT System with Distinct Eight Datasets.

Table 2 and **Fig 4** deliver NACC inspection of the AEODHL-VTNT technique with recent approaches on eight datasets. These results denoted the AEODHL-VTNT algorithm has least values of NACC in all normalized frequencies. Meanwhile, it is noted that the AEODHL-VTNT method has gained reduced NACC values with an increase in normalized frequencies. i.e., the value of NACC was maximum at 1.0x normalized frequency and will reach minimum at 4.0x normalized frequency.

Table 2. NACC Comparison between the AEODHL-VTNT System using Eight Independent Datasets

Accuracy (Normalized)										
Frequency (Normalized)	1.0x	1.09x	1.2x	1.33x	1.5x	1.71x	2.0x	2.4x	3.0x	4.0x
SVHN	0.9993	0.9964	0.9939	0.9873	0.9787	0.9476	0.8926	0.7414	0.6517	0.6419
CIFAR-10	0.9864	0.9793	0.985	0.9669	0.9614	0.9177	0.719	0.4735	0.2928	0.3005
IMDB	0.9822	0.9682	0.961	0.9522	0.917	0.8094	0.5502	0.2834	0.168	0.1679
GTSRB	0.9966	0.9738	0.9578	0.9101	0.8105	0.6437	0.406	0.2165	0.1203	0.1141
REUTERS	0.9694	0.9692	0.9358	0.8721	0.7549	0.5682	0.3127	0.2064	0.2115	0.1929
MNIST	0.9523	0.9462	0.921	0.7906	0.5767	0.3764	0.2998	0.2603	0.2437	0.2241
FMNIST	0.9756	0.9758	0.9185	0.7774	0.3757	0.1854	0.1541	0.1389	0.1382	0.1316
AMNIST	0.98	0.9619	0.9528	0.8256	0.4739	0.2304	0.1443	0.1184	0.1356	0.1333

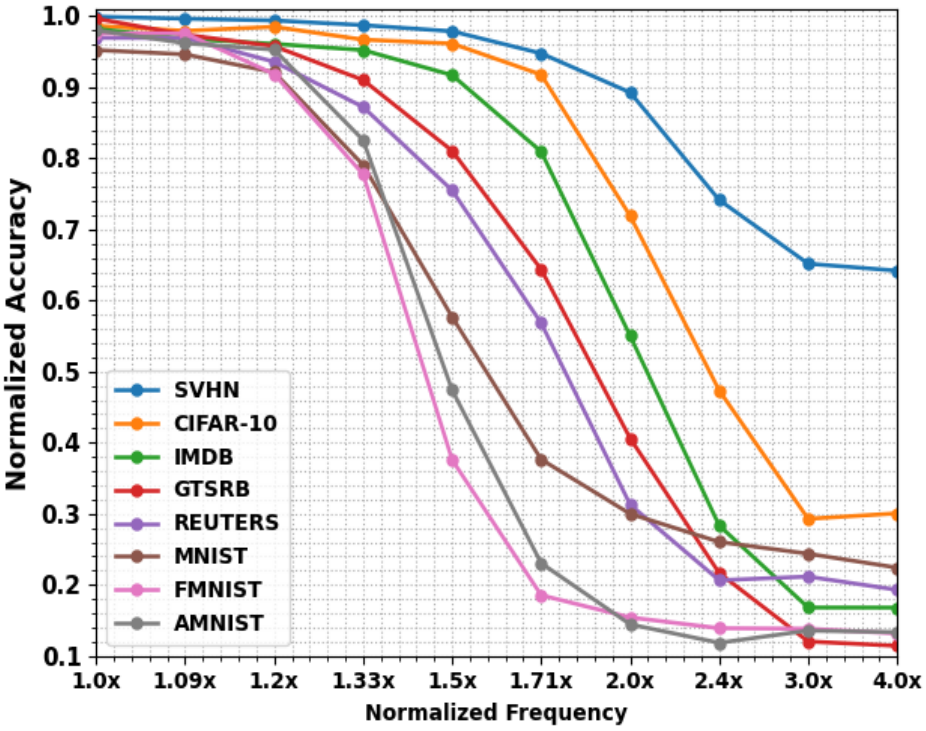

Fig 4. NACC examines AEODHL-VTNT System with Distinct Eight Datasets.

Table 3 offers comparative NACC outcomes of the AEOHDL-VTNT model with SVHN dataset [4]. The results of the experiment showed that, across all NFs, the IA(MRFF) model obtained the lowest values of NACC. Next, the BE (PRED) and IA(TED) techniques have tried to report closer NACC values. On the other hand, the IA(PRED) model exhibits certainly improved NACC. Although the EEHPT-DNN model has provided near optimal values of NACC, the AEOHDL-VTNT model has shown increasing values of NACC under all NFs.

Table 3. NACC Comparison of the AEOHDL-VTNT Method with Alternative Systems using the SVHN Dataset

Accuracy (Normalized)– SVHN													
Frequency (Normalized)	1.0x	1.075x	1.15x	1.255x	1.36x	1.515x	1.67x	1.875x	2.14x	2.57x	3.0x	4.0x	5.0x
AEOHDL-VTNT	0.9992	0.9979	0.9984	0.9988	0.9979	0.9966	0.9954	0.993	0.9935	0.9935	0.9927	0.9929	0.992
EEHPT-DNN	0.998	0.997	0.997	0.997	0.996	0.995	0.994	0.992	0.992	0.992	0.991	0.991	0.99
IA(PRED)	0.974	0.98	0.981	0.973	0.981	0.98	0.978	0.97	0.974	0.97	0.97	0.972	0.97
IA(MRFF)	0.973	0.973	0.976	0.971	0.976	0.968	0.977	0.971	0.966	0.852	0.584	0.394	0.312
BE(PRED)	0	0	0	0	0	0	0.888	0.908	0.974	0.972	0.963	0.961	0.965
IA(TED)	0.979	0.978	0.974	0.978	0.973	0.979	0.972	0.973	0.968	0.969	0.97	0.97	0.965

Table 4 offers Comparing the AEOHDL-VTNT model's NACC results to the CIFAR-10 dataset.

The results demonstrated that the IA(MRFF) method has gotten least values of NACC under all NFs. Next, the BE (PRED) and IA(TED) techniques have tried to report closer NACC values. On the other hand, IA(PRED) model exhibits certainly improved NACC. Although the EEHPT-DNN model has provided near optimal values of NACC, the AEOHDL-VTNT model has shown increasing values of NACC under all NFs.

Table 4. NACC Examine AEOHDL-VTNT Method with Alternative Systems using CIFAR-10 Dataset

Accuracy (Normalized)- CIFAR-10													
Frequency (Normalized)	1.0x	1.075x	1.15x	1.255x	1.36x	1.515x	1.67x	1.875x	2.14x	2.57x	3.0x	4.0x	5.0x
AEOHDL-VTNT	0.9969	0.9929	0.9976	0.9779	0.9978	0.9873	0.9763	0.9849	0.997	0.9896	0.9842	0.979	0.9775
EEHPT-DNN	0.995	0.992	0.996	0.977	0.997	0.986	0.975	0.984	0.996	0.988	0.983	0.978	0.976
IA(PRED)	0.959	0.97	0.969	0.962	0.962	0.964	0.965	0.959	0.958	0.955	0.959	0.955	0.959
IA(MRFF)	0.956	0.954	0.966	0.958	0.959	0.951	0.961	0.957	0.953	0.856	0.573	0.48	0.292
BE(PRED)	0	0	0	0	0	0	0.876	0.892	0.958	0.953	0.943	0.949	0.91
IA(TED)	0.962	0.968	0.957	0.961	0.96	0.965	0.962	0.959	0.949	0.956	0.957	0.952	0.928

Table 5 offers comparative NACC outcomes of the AEOHDL-VTNT model with IMDB dataset. The outcomes demonstrated that the IA(MRFF) technique has gotten least values of NACC under all NFs. Next, the BE (PRED) and IA(TED) techniques have tried to report closer NACC values. On the other hand, the IA(PRED) model exhibits certainly improved NACC. Although the EEHPT-DNN model has provided near optimal values of NACC, the AEOHDL-VTNT model has shown increasing values of NACC under all NFs.

Table 5. NACC Evolution of AEOHDL-VTNT Method with Alternative Systems using IMDB Dataset

Accuracy (Normalized)– IMDB													
Frequency (Normalized)	1.0x	1.075x	1.15x	1.255x	1.36x	1.515x	1.67x	1.875x	2.14x	2.57x	3.0x	4.0x	5.0x
AEOHDL-VTNT	0.9823	0.9829	0.9873	0.9985	0.998	0.9968	0.9992	0.9798	0.9919	0.9902	0.9783	0.9805	0.9785
EEHPT-DNN	0.981	0.982	0.986	0.997	0.996	0.995	0.998	0.979	0.991	0.989	0.977	0.979	0.977
IA(PRED)	0.975	0.979	0.983	0.979	0.982	0.954	0.953	0.97	0.953	0.957	0.954	0.954	0.968
IA(MRFF)	0.965	0.955	0.955	0.951	0.975	0.962	0.978	0.971	0.963	0.893	0.675	0.412	0.309
BE(PRED)	0	0	0	0	0	0	0.88	0.911	0.951	0.967	0.956	0.95	0.905
IA(TED)	0.977	0.971	0.969	0.97	0.972	0.954	0.958	0.968	0.956	0.967	0.954	0.969	0.857

Table 6 offers comparative NACC outcomes of the AEOHDL-VTNT model with GTSRB dataset. The results demonstrated that the IA(MRFF) method has gotten the least values of NACC under all NFs. Next, the BE (PRED) and IA(TED) techniques have tried to report closer NACC values. On the other hand, the IA(PRED) model exhibits certainly improved NACC. Although the EEHPT-DNN model has provided near optimal values of NACC, the AEOHDL-VTNT model has shown increasing values of NACC under all NFs.

Table 6. NACC Comparison of AEOHDL-VTNT Method with Alternative Systems using GTSRB Dataset

Accuracy (Normalized)– GTSRB													
Frequency (Normalized)	1.0x	1.075x	1.15x	1.255x	1.36x	1.515x	1.67x	1.875x	2.14x	2.57x	3.0x	4.0x	5.0x
AEOHDL-VTNT	0.987	0.9914	0.9865	0.9991	0.9987	0.986	0.983	0.9965	0.9778	0.9852	0.987	0.9878	0.9806
EEHPT-DNN	0.978	0.984	0.98	0.995	0.992	0.982	0.975	0.991	0.973	0.979	0.981	0.981	0.976
IA(PRED)	0.971	0.971	0.977	0.953	0.973	0.973	0.964	0.973	0.963	0.967	0.968	0.963	0.956
IA(MRFF)	0.98	0.959	0.958	0.979	0.965	0.966	0.965	0.959	0.961	0.892	0.573	0.328	0.287
BE(PRED)	0	0	0	0	0	0	0.871	0.906	0.951	0.962	0.953	0.955	0.846
IA(TED)	0.973	0.97	0.975	0.98	0.965	0.968	0.969	0.961	0.95	0.96	0.965	0.963	0.89

Table 7 offers comparative NACC outcomes of the AEOHDL-VTNT model with REUTERS dataset. The results demonstrated that the IA(MRFF) method has gotten least values of NACC under all NFs. Next, the BE (PRED) and IA(TED) techniques have tried to report closer NACC values. On the other hand, the IA(PRED) model exhibits certainly improved NACC. Although the EEHPT-DNN model has provided near optimal values of NACC, the AEOHDL-VTNT model has shown increasing values of NACC under all NFs.

Table 7. NACC Study of AEOHDL-VTNT Method with Alternative Systems using REUTERS Dataset

Accuracy (Normalized)– REUTERS												
Frequency (Normalized)	1.0x	1.075x	1.15x	1.255x	1.36x	1.515x	1.67x	1.875x	2.14x	3.0x	4.0x	5.0x
AEOHDL-VTNT	0.9848	0.9941	0.9861	0.9885	0.9965	0.9825	0.9886	0.9855	0.9854	0.9848	0.9758	0.9957
EEHPT-DNN	0.978	0.987	0.981	0.983	0.991	0.975	0.981	0.981	0.981	0.979	0.971	0.994
IA(PRED)	0.972	0.951	0.964	0.964	0.952	0.951	0.946	0.962	0.945	0.966	0.956	0.945
IA(MRFF)	0.975	0.957	0.959	0.961	0.963	0.961	0.95	0.967	0.952	0.573	0.374	0.298
BE(PRED)	0	0	0	0	0	0	0.867	0.904	0.957	0.945	0.959	0.945
IA(TED)	0.962	0.958	0.97	0.963	0.966	0.966	0.955	0.966	0.97	0.966	0.944	0.94

Table 8 offers comparative NACC outcomes of the AEOHDL-VTNT model with MNIST dataset. The results demonstrated that the IA(MRFF) method has gotten least values of NACC under all NFs. Next, the BE (PRED) and IA(TED) techniques have tried to report closer NACC values. On the other hand, the IA(PRED) model exhibits certainly improved NACC. Although the EEHPT-DNN model has provided near optimal values of NACC, the AEOHDL-VTNT model has shown increasing values of NACC under all NFs.

Table 8. AEOHDL-VTNT Method with Alternative Systems using MNIST Dataset of NACC Study

Accuracy (Normalized)– MNIST													
Frequency (Normalized)	5.0x	4.0x	3.0x	2.57x	2.14x	1.875x	1.67x	1.515x	1.36x	1.255x	1.15x	1.075x	1.0x
AEOHDL-VTNT	0.9832	0.9796	0.9925	0.9898	0.989	0.9959	0.993	0.98	0.9938	0.9989	0.9974	0.9929	0.9992
EEHPT-DNN	0.979	0.974	0.987	0.985	0.984	0.989	0.985	0.976	0.989	0.998	0.996	0.985	0.999
IA(PRED)	0.953	0.943	0.944	0.968	0.957	0.943	0.95	0.963	0.964	0.955	0.968	0.972	0.962
IA(MRFF)	0.294	0.481	0.581	0.837	0.959	0.952	0.964	0.967	0.963	0.953	0.968	0.969	0.969
BE(PRED)	0.942	0.957	0.958	0.961	0.956	0.909	0.883	0	0	0	0	0	0
IA(TED)	0.968	0.948	0.962	0.961	0.97	0.952	0.956	0.969	0.952	0.958	0.957	0.973	0.96

Table 9 offers comparative NACC outcomes of the AEOHDL-VTNT model with FMNIST dataset. The outcomes demonstrated that the IA(MRFF) technique has gotten least values of NACC under all NFs. Next, the BE (PRED) and IA(TED) techniques have tried to report closer NACC values. On the other hand, the IA(PRED) model exhibits certainly improved NACC. Although the EEHPT-DNN model has provided near optimal values of NACC, the AEOHDL-VTNT model has shown increasing values of NACC under all NFs

Table 9. NACC Study of AEOHDL-VTNT Method with Other Systems in the FMNIST Dataset

Accuracy (Normalized)– FMNIST													
Frequency (Normalized)	5.0x	4.0x	3.0x	2.57x	2.14x	1.875x	1.67x	1.515x	1.36x	1.255x	1.15x	1.075x	1.0x
AEOHDL-VTNT	0.9984	0.9754	0.9864	0.9981	0.9963	0.9825	0.9861	0.989	0.983	0.9978	0.99	0.999	0.9975
EEHPT-DNN	0.992	0.971	0.979	0.996	0.995	0.976	0.982	0.983	0.978	0.992	0.983	0.991	0.997
IA(PRED)	0.953	0.949	0.959	0.958	0.949	0.952	0.953	0.951	0.964	0.95	0.968	0.976	0.972
IA(MRFF)	0.292	0.382	0.564	0.937	0.95	0.967	0.955	0.976	0.954	0.963	0.952	0.966	0.976
BE(PRED)	0.952	0.955	0.957	0.956	0.953	0.902	0.861	0	0	0	0	0	0
IA(TED)	0.949	0.957	0.959	0.968	0.962	0.962	0.951	0.966	0.969	0.967	0.951	0.971	0.969

Table 10 offers comparative NACC outcomes of the AEOHDL-VTNT model with AMNIST dataset. The figure demonstrated that the IA(MRFF) approach has gotten least values of NACC under all NFs. Next, the BE (PRED) and IA(TED) techniques have tried to report closer NACC values. On the other hand, the IA(PRED) model exhibits certainly improved NACC. Although the EEHPT-DNN model has provided near optimal values of NACC, the AEOHDL-VTNT model has shown increasing values of NACC under all NFs.

Table 10. NACC Comparison of the AEOHDL-VTNT Method with Other Systems using the AMNIST Dataset

Accuracy (Normalized)– AMNIST													
Frequency (Normalized)	5.0x	4.0x	3.0x	2.57x	2.14x	1.875x	1.67x	1.515x	1.36x	1.255x	1.15x	1.075x	1.0x
AEOHDL-VTNT	0.9958	0.9863	0.985	0.9977	0.9999	0.983	0.9835	0.9858	0.996	0.9991	0.9904	0.9985	0.9854
EEHPT-DNN	0.991	0.981	0.977	0.996	0.993	0.976	0.976	0.98	0.989	0.995	0.984	0.998	0.981
IA(PRED)	0.96	0.949	0.946	0.957	0.952	0.958	0.969	0.956	0.962	0.959	0.966	0.969	0.961
IA(MRFF)	0.284	0.373	0.577	0.821	0.968	0.953	0.961	0.953	0.968	0.957	0.952	0.965	0.951
BE(PRED)	0.94	0.949	0.946	0.966	0.953	0.89	0.874	0	0	0	0	0	0
IA(TED)	0.944	0.96	0.95	0.953	0.961	0.969	0.951	0.97	0.945	0.954	0.969	0.973	0.952

Fig 5 Depicts the NACC outcomes of the AEOHDL-VTNT approach with other existing methods under distinct databases and varying NF. These results depicted the supreme performance of the AEOHDL-VTNT model.

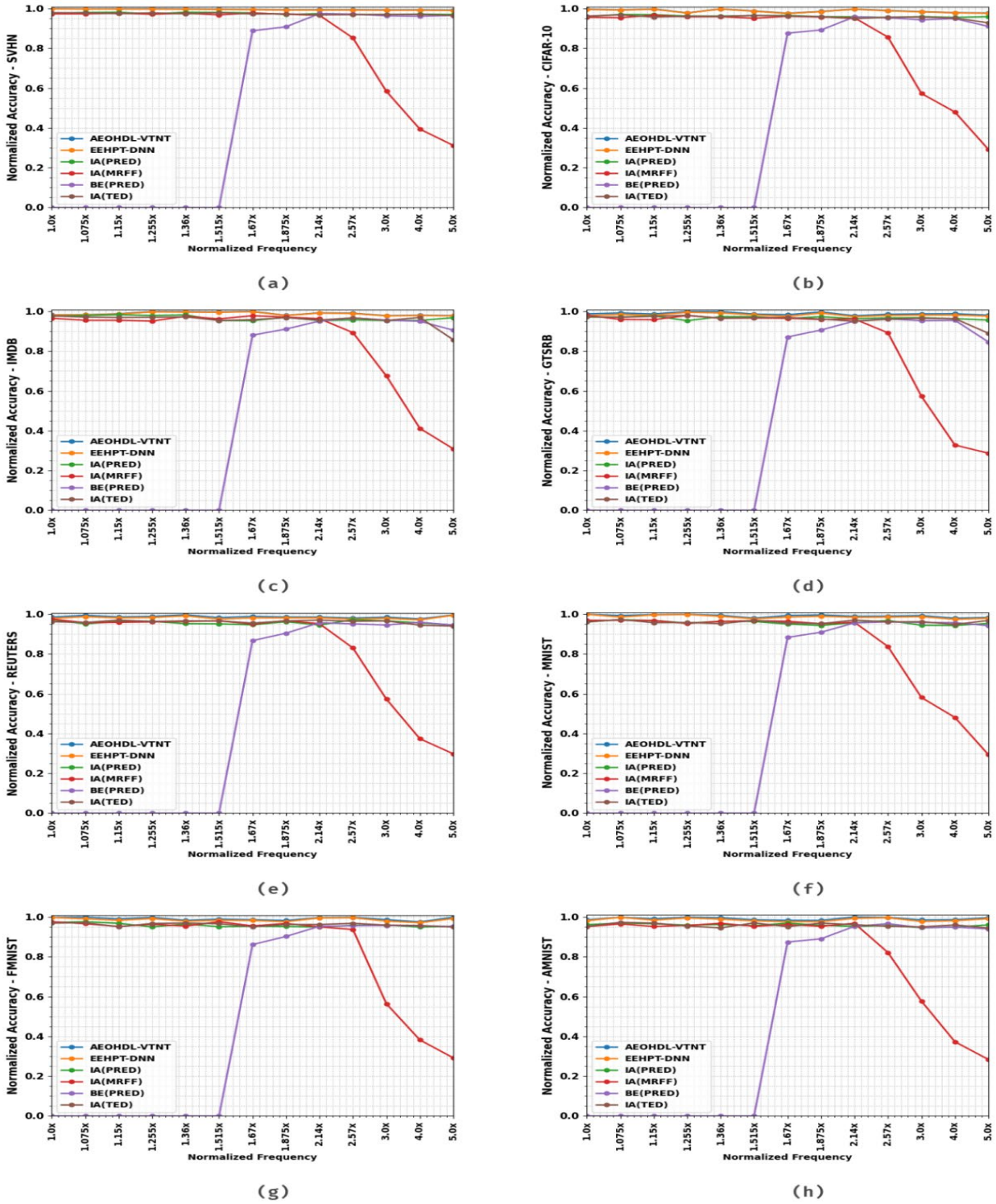

Fig 5. NACC examine AEOHDL-VTNT strategy (a) SVHN, (b) CIFAR-10, (c) IMDB, (d) GTSRB, (e) REUTERS, (f) MNIST, (g) FMNIST, and (h) AMNIST.

V. CONCLUSION

DL models have been hybridized in this work to create a novel AEOHDL-VTNT approach. The inference of systems embedded at the network's edge serves as the foundation for the AEOHDL-VTNT approach that is being discussed. the AEOHDL-VTNT approach that is being demonstrated, two major processes are involved namely HDL based VTNT design

and hyperparameter tuning. At the initial stage, the HDL model is employed for the design of VTNT. Next, in the second stage, the AEO algorithm is applied for the hyperparameter tuning of the HDL model and thereby enhances the overall efficacy of the HDL. TO demonstrate the enhanced performance of the AEOHDL-VTNT technique, a series of simulations were taken place. The simulation outcomes highlighted that the AEOHDL-VTNT technique reaches superior performance over other models.

Data Availability
No data was used to support this study.

Conflicts of Interests
The author(s) declare(s) that they have no conflicts of interest.

Funding
No funding agency is associated with this research.

Competing Interests
There are no competing interests

References

[1]. G. H. Choi and T. Na, "Novel MTJ-Based Sensing Inverter Variation Tolerant Nonvolatile Flip-Flop in the Near-Threshold Voltage Region," IEEE Access, vol. 8, pp. 191057–191066, 2020, doi: 10.1109/access.2020.3032576.
[2]. J. Lee, M. Kim, M. Jeong, G. Shin, and Y. Lee, "A 20F2/Bit Current-Integration-Based Differential nand-Structured PUF for Stable and V/T Variation-Tolerant Low-Cost IoT Security," IEEE Journal of Solid-State Circuits, vol. 57, no. 10, pp. 2957–2968, Oct. 2022, doi: 10.1109/jssc.2022.3192903.
[3]. R. Uytterhoeven and W. Dehaene, "Design Margin Reduction Through Completion Detection in a 28-nm Near-Threshold DSP Processor," IEEE Journal of Solid-State Circuits, vol. 57, no. 2, pp. 651–660, Feb. 2022, doi: 10.1109/jssc.2021.3106245.
[4]. N. D. Gundi, P. Pandey, S. Roy, and K. Chakraborty, "Implementing a Timing Error-Resilient and Energy-Efficient Near-Threshold Hardware Accelerator for Deep Neural Network Inference," Journal of Low Power Electronics and Applications, vol. 12, no. 2, p. 32, Jun. 2022, doi: 10.3390/jlpea12020032.
[5]. S. Vangal et al., "Near-Threshold Voltage Design Techniques for Heterogenous Manycore System-on-Chips," Journal of Low Power Electronics and Applications, vol. 10, no. 2, p. 16, May 2020, doi: 10.3390/jlpea10020016.
[6]. M. Kumar, A. Majumder, A. J. Mondal, A. Raychowdhury, and B. K. Bhattacharyya, "A low power and PVT variation tolerant mux-latch for serializer interface and on-chip serial link," Integration, vol. 87, pp. 364–377, Nov. 2022, doi: 10.1016/j.vlsi.2022.08.008.
[7]. D. Wang, S. J. Kim, M. Yang, A. A. Lazar, and M. Seok, "9.9 A Background-Noise and Process-Variation-Tolerant 109nW Acoustic Feature Extractor Based on Spike-Domain Divisive-Energy Normalization for an Always-On Keyword Spotting Device," 2021 IEEE International Solid- State Circuits Conference (ISSCC), Feb. 2021, doi: 10.1109/isscc42613.2021.9365969.
[8]. S. Ryu, J. Koo, W. Kim, Y. Kim, and J.-J. Kim, "Variation-Tolerant Elastic Clock Scheme for Low-Voltage Operations," IEEE Journal of Solid-State Circuits, vol. 56, no. 7, pp. 2245–2255, Jul. 2021, doi: 10.1109/jssc.2020.3048881.
[9]. M. Manimaran, A. Sasi Kumar, N. V. S. Natteshan, K. Baranitharan, R. Mahaveerakannan, and K. Sudhakar, "Detecting the Human Activities of Aging People using Restricted Boltzmann Machines with Deep Learning Technique in IoT," 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), Feb. 2023, doi: 10.1109/icais56108.2023.10073665.
[10]. M. Ling, Q. Lin, K. Tan, T. Shao, S. Shen, and J. Yang, "A Design of Timing Speculation SRAM-Based L1 Caches With PVT Autotracking Under Near-Threshold Voltages," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 12, pp. 2197–2209, Dec. 2021, doi: 10.1109/tvlsi.2021.3120653.
[11]. Y. Lin and J. R. Cavallaro, "Energy-efficient Convolutional Neural Networks via Statistical Error Compensated Near Threshold Computing," 2018 IEEE International Symposium on Circuits and Systems (ISCAS), 2018, doi: 10.1109/iscas.2018.8351679.
[12]. Mahaveerakannan R, C. Anitha, Aby K Thomas, S. Rajan, T. Muthukumar, and G. Govinda Rajulu, "An IoT based forest fire detection system using integration of cat swarm with LSTM model," Computer Communications, vol. 211, pp. 37–45, Nov. 2023, doi: 10.1016/j.comcom.2023.08.020.
[13]. X. Fan, H. Li, Q. Li, R. Wang, H. Liu, and S. Lu, "A Light-Weight Timing Resilient Scheme for Near-Threshold Efficient Digital ICs," 2020 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Dec. 2020, doi: 10.1109/apccas50809.2020.9301717.
[14]. S. P. Jadhav, A. Srinivas, P. Dipak Raghunath, M. Ramkumar Prabhu, J. Suryawanshi, and A. Haldorai, "Deep learning approaches for multi-modal sensor data analysis and abnormality detection," Measurement: Sensors, vol. 33, p. 101157, Jun. 2024, doi: 10.1016/j.measen.2024.101157.
[15]. Y. Lin, S. Zhang, and N. R. Shanbhag, "A Rank Decomposed Statistical Error Compensation Technique for Robust Convolutional Neural Networks in the Near Threshold Voltage Regime," Journal of Signal Processing Systems, vol. 90, no. 10, pp. 1439–1451, Feb. 2018, doi: 10.1007/s11265-018-1332-4.
[16]. M. Lalithambigai, V. Kalpana, A. Sasi Kumar, J. Uthayakumar, J. Santhosh, and R. Mahaveerakannan, "Dimensionality Reduction with DLMNN Technique for Handling Secure Medical Data in Healthcare-IoT Model," 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), Feb. 2023, doi: 10.1109/icais56108.2023.10073679.
[17]. P. Pandey, P. Basu, K. Chakraborty, and S. Roy, "GreenTPU," Proceedings of the 56th Annual Design Automation Conference 2019, Jun. 2019, doi: 10.1145/3316781.3317835.
[18]. L. J. Muhammad, A. A. Haruna, U. S. Sharif, and M. B. Mohammed, "CNN-LSTM deep learning based forecasting model for COVID-19 infection cases in Nigeria, South Africa and Botswana," Health and Technology, vol. 12, no. 6, pp. 1259–1276, Nov. 2022, doi: 10.1007/s12553-022-00711-5.
[19]. H. O. Omotoso, A. M. Al-Shaalan, H. M. H. Farh, and A. A. Al-Shamma'a, "Techno-Economic Evaluation of Hybrid Energy Systems Using Artificial Ecosystem-Based Optimization with Demand Side Management," Electronics, vol. 11, no. 2, p. 204, Jan. 2022, doi: 10.3390/electronics11020204.