International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Workload Characterization in Embedded Systems Utilizing Hybrid Intelligent Gated Recurrent Unit and Extreme Learning Machines

R. Sivaramakrishnan *[1], G. SenthilKumar[2]

Submitted: 29/06/2023 **Revised**: 09/08/2023 **Accepted**: 29/08/2023

Abstract: As the demand for embedded systems continues to rise exponentially, accurately estimating and predicting the necessary resources for these systems remains a significant challenge due to their dynamic workloads. Numerous intelligent algorithms have been developed using machine learning and deep learning techniques, leveraging their computational power and ability to capture workload patterns. However, these algorithms still require further refinement to effectively handle the increasingly diverse and rapidly evolving workloads. This framework aims to design Gated Extreme learning machines for Embedded Workload Characterization (GEEWC) to address the challenges of accurately characterizing and predicting resource requirements in embedded systems, which often operate under dynamic and unpredictable workloads. By combining the Gated Recurrent Unit (GRU) and Extreme Learning Machines (ELM), GEEWC leverages the strengths of both models to improve the accuracy and efficiency of workload characterization and resource prediction. The results of the experimentation show that the suggested framework performs consistently well across all three workload benchmarks such as the Internet of Medical Things (IoMT), EEMBC, and SPARK workloads. The F1-Score, recall, specificity, accuracy, and recall metrics consistently reflect high levels of performance, indicating that the framework is able to effectively handle dynamic workloads. This robustness makes it a reliable solution for real-world scenarios where workloads can vary significantly. Further analysis of the results reveals that the framework is particularly effective in handling the complex IoMT workload, suggesting its suitability for healthcare applications. Moreover, the framework exhibits robustness and scalability, allowing it to handle large datasets and accommodate future growth in the healthcare industry. The results also highlight the framework's ability to accurately predict and diagnose medical conditions, making it a valuable tool for healthcare professionals. Overall, these findings solidify the framework's potential for revolutionizing healthcare applications and improving patient outcomes.

Keywords: *Embedded System, Gated Recurrent Unit, Extreme Learning Machines, Workload Characterization*.

1. Introduction

The embedded systems have revolutionized the way we interact with technology and have enabled seamless connectivity between devices [1]. The integration of embedded systems in the IoT has allowed for smart homes, where devices can communicate and automate tasks [2]. Advanced health care applications have benefited from embedded systems by improving patient monitoring and enabling remote diagnosis. Additionally, embedded systems have played a crucial role in the development of 5G communications, ensuring faster and more reliable network connections [3]. This new dimension involves the integration of artificial intelligence and machine learning algorithms into embedded systems. By incorporating these technologies, embedded systems can adapt and learn from the ever-changing workload demands, optimising performance and efficiency. Furthermore, the incorporation of AI and machine learning can enable embedded systems to make intelligent decisions in real-time, improving the overall user experience and satisfaction [4].

As technology continues to advance, the evolution of embedded systems with AI capabilities will be crucial to meeting the growing demands of various industries.

These techniques involve analysing the workload patterns and behaviour of embedded systems in order to optimise their performance and resource allocation [5]. By understanding the specific needs and requirements of the workload, AI and machine learning algorithms can be leveraged to dynamically adjust the system's settings and configurations, ensuring efficient utilisation of resources and improved overall performance.

This integration of workload characterization techniques with AI capabilities will further enhance the adaptability and responsiveness of embedded systems, making them well-equipped to handle the ever-changing demands of different industries [6]. This constant evolution has allowed embedded systems to seamlessly integrate into a wide range of industries, including automotive, healthcare, telecommunications, and manufacturing.

[1]*Research Scholar, Department of Electronics and Communication Engineering, Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Kanchipuram- 631561, TamilNadu, India*
ORCID ID: 0009-0000-0807-1207
[2]*Associate Professor, Department of Electronics and Communication Engineering, Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Kanchipuram- 631561, TamilNadu, India*
ORCID ID: 0009-0002-2796-569X
** Corresponding Author Email: sivaram6685@gmail.com*

From controlling complex machinery to monitoring patient health, embedded systems have proven their ability to adapt and respond to the unique requirements of each industry. With their versatility and reliability, they have become an indispensable component of modern technology-driven sectors [7].

By identifying the bottlenecks and areas for optimization, hardware developers can make necessary adjustments to improve the overall performance of the system. For example, if the processing speed is found to be a bottleneck, developers can explore options to increase clock speed or improve the efficiency of the processing units. Similarly, if cache memory utilization is identified as a limitation, developers can optimize data caching techniques to reduce memory access times and improve overall system performance. Ultimately, this understanding of internal hardware characteristics is crucial for achieving maximum efficiency and performance from the hardware system. Furthermore, studying the internal characteristics of hardware systems helps in making informed decisions regarding hardware upgrades or replacements. By considering factors such as processing speed and cache memory utilization, system performance can be enhanced, leading to improved efficiency and user satisfaction. Overall, the analysis and validation of hardware architectures are crucial for achieving optimal performance in computing systems [8].

As the complexity of embedded systems continues to increase, it becomes crucial to employ effective techniques that can accurately capture the system's workload. This includes gathering data on various metrics such as CPU utilization, memory usage, and I/O operations. By using efficient and accurate techniques, developers can gain valuable insights into the system's behavior and make informed decisions regarding resource allocation and optimization. Ultimately, this leads to improved system performance, reduced energy consumption, and enhanced overall reliability of embedded systems. In addition to monitoring these metrics, developers can also implement anomaly detection algorithms to identify any abnormal patterns or deviations in the system's workload. Furthermore, by analyzing historical data and trends, developers can proactively anticipate potential bottlenecks or performance issues and take preventive measures. These techniques enable developers to create more efficient and reliable embedded systems that meet the demands of today's complex and resource-intensive applications [9].

Nowadays, the utilisation of machine learning and deep learning techniques in embedded workload characterization can lead to more efficient and intelligent computing systems that can keep up with the growing demands of modern applications. But existing methods such DNN, CNN methods still suffer from non-handling of heterogeneous workloads, non-adaptability to dynamic workloads, and time independent analysis [10].Motivated by the aforementioned problem, this article suggests a deep learning model with a hybrid approach referred as GEEWC that consists of the Gated recurrent Unit and Extreme Learning Machines to achieve the better workload prediction model for the dynamic and heterogeneous workloads. By integrating these techniques into the development process, developers can stay ahead of potential issues and deliver embedded systems that excel in performance and reliability.

Here is the paper's primary contribution:

- To combine GRU and ELM to develop a powerful tool for ensuring efficient and reliable system performance in dynamic and unpredictable workload environments.
- Exploring the challenges faced by resource allocation in unpredictable workload environments.
- Real-world examples where accurate workload characterization and performance prediction have led to improved system efficiency.
- Investigating the trade-offs between accuracy and computational complexity when implementing GRU and ELM in resource-constrained systems.

The rest of the paper is organised as follows: The related works by various writers are shown in Section 2. The intended proposed method and workings of the intelligent deep learning algorithms are presented in Section 3. The implementation, outcomes of the implementation, and comparative analysis are presented in Section 4. The article concludes with a consideration of the future in the final section.

2. Literature Survey

A four-month HPC workload analysis carried out on the NERSC Cori supercomputer was the topic of Jiwoo Bang et al.'s (2020) research. This framework extracted many features from the log and then utilised Mutual information regression, F-regression, Decision tree, Extra tree, and Min-max mutual information to choose the features that best represented the data. After that, data clustering methods like KMeans, GMM, and Ward linkage are used to group the characteristics that have been chosen. This system chooses the appropriate clustering method and feature selection approach by measuring cluster validity metrics against clustering performance. The best clustering result, according to the results, may be achieved utilising the KMeans clustering approach with features chosen from the Min-max mutual information. Last but not least, this framework can give system designers a better understanding of the HPC applications operating on the system, simplifying the configuration options made available to users for improved performance. The

framework's great computational complexity, however, is a key downside. [11].

The performance of DGX-2, AWS P3, and IBM-P9, three cutting-edge systems built for DL workload performance, was examined by Ren et al. in 2019. An affordable consumer-grade machine called an RTX-2080 Ti server was also taken into account by this architecture. The inclusion of Amazon P3, a system that is essentially a DGX-1 system, was made in order to investigate performance along with the rising use of cloud computing situations for DL workloads. The DL models that were put to the test included both computer vision and natural language processing, are accurate, and are really applied in practical DL applications. The systems were examined utilizing various realistic computing and communication scenarios by altering the types of neural network models and batch sizes per GPU. As the communication performance is slower, this framework provides higher throughput but less scalability [12].

Jayanthi E. et al. (2021) developed the BAT LSTM neural network predictor and evaluated it against support vector machines, decision trees, naive Bayes, random forest methods, and naive Bayes. Cost functions are designed and created for these algorithms in order to identify the optimal processor for each job execution at runtime. To analyze workload metrics including memory use, I/O, CPU utilization, instruction type, cache miss ratios, and other factors, core mark workloads are initially performed on quad core multicore systems. These characteristics are sent into a machine learning system, which determines the ideal processor. Testing workloads are used to assess the performance of proposed algorithms in terms of processor prediction accuracy and execution time parameters. The suggested predictors achieve an average energy usage decrease of 10% and accuracy of 96.8%. The main disadvantage of this framework is its increased latency [13].

Because comprehensive application settings on customers' VMs are not available to the cloud providers, A. Khan et al., (2021), developed a novel way of assessing and estimating workload in a cloud context. With the help of this technology, a cloud customer's groups of VMs can be found and their repeated workload patterns may be used. To find such VM groupings and typical workload patterns, a co-clustering approach is created. An HMM-based approach is created to capture the temporal correlations and forecast changes in the workload pattern based on the co-clusters that were detected. For estimating workload fluctuations at the individual sever level, our technique demonstrated much greater prediction accuracy. Yet, the training is more difficult [14].

Sebastian Stefan et al. presented a workload prediction model in 2022 that makes use of MultiLayer Perceptron (MLP) and takes micro service concerns into account. The study highlights the excellent results obtained with the MLP (Multi-Layer Perceptron) model, which are superior to those obtained with conventional statistical techniques, with a reduction of the average error in estimation to 49%, while utilizing two Wikipedia traces over 12 days and with two distinct time windows: 10 and 15 minutes. The results of the tests and comparison analyses suggest that in addition to accuracy, computational complexity and prediction time should also be considered. It might be tough to scale up to bigger datasets [15].

Krishan Kumar et al. (2021) examined several machine learning techniques to anticipate the workload for next forecasts. NASA datasets and ClarkNet are utilised for it. According to the experimental findings, KNN and ARIMA models significantly improve for ClarkNet and NASA, respectively, whereas linear regression models significantly improve for NASA. “MAE, MSE, RMSE, and MAPE” are the QoS metrics that have seen the most improvement. The experimental findings reveal that the QoS metrics and cloud data center availability in a cloud environment are both significantly improved by the ARIMA model, as is forecasting. Nonetheless challenging to train due to the memory-bandwidth-bound computation they demand [16].

Sivaramakrishnan et al. (2021) provide a unique method for workload characterisation that improves application performance in terms of both hardware-software resource mapping. To anticipate the optimal computing resource for each task at runtime, a long short term memory (LSTM) based RNN classifier is presented for HSoC platforms. In order to comprehend how real-time workloads operate at runtime, the suggested classifier looked at the implementation of numerous HSoC platforms. For the purpose of training and testing the LSTM classifier, the observed attributes are presented as a real-time database. To evaluate how well the suggested algorithms work, accuracy, throughput, sensitivity, and selectivity measures are found. For real-time embedded benchmark workloads like MiBench and IoMT, LICHA achieved accuracy up to 96% while using up to 30% less energy during execution. Nevertheless, this approach has a disadvantage in that it takes longer [17].

Ahamed, Zaakki, and colleagues (2023) provided a technical analysis of using DL models like Recurrent Neural Networks (RNN), MLP, LSTM, and Convolutional Neural Networks (CNN) to benefit from the time series characteristics of real-world workloads from the Parallel Workloads Archive of the Standard Workload Format (SWF) in order to conduct an objective analysis. We evaluate the robustness of these models using the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) error metrics. The results of these

investigations show that the LSTM model performs better than the other models. It takes a long time to learn complicated connections, though [18].

A hybrid technique that combines an enhanced LSTM network with a multilayer perceptron network is the basis of the predictive cloud workload management framework proposed by KumarK. Dinesh et al. in 2020. With the addition of an opposition-based differential evolution method, dropout technique on recurrent connections, and dropout on memory-preserving connections, the proposed approach may perform a better prediction process than the existing LSTM architecture. Improved prediction performance for the cloud workload is the goal of a revolutionary hybrid predictive technique. Check the efficacy of the suggested strategy using benchmark data from Saskatchewan servers and NASA. High scalability is offered by this framework, however it uses more energy [19].

Om, Khandu et al. (2020) considered email traffic as a time series function when estimating the workloads associated with email traffic. RNN and LSTM models have been used to model the email traffic from four different institutions employing this system. It has been proven that by employing the best initialization of the training weights, the right activation functions, and hyper-parameters, the RNN model's ability to simulate email traffic may be greatly improved. The performance of LSTM is greater for the majority of email traffic types, whereas RNN's maximum accuracy is less impressive. Although not assured, the resilience [20].

From the literature survey it is clear that the existing systems were lack of handling dynamic workloads, consumes more power, possess less accuracy, required more MBT and not suitable for real time environment. To consider the all the above mentioned issues, this research work introduced a novel GEEWC to achieve better performance in embedded systems.

3. Proposed Methodology

The proposed deep learning model-based workload characterisation and prediction process is shown generally in Figure 1. The three different workloads are collected and recorded in a workload repository within the workload prediction and management unit. The repository is mined for the raw data, such as register use, instruction mix, cache usages, branch prediction units, and arithmetic operations, which are then extracted and transported to the intended deep learning unit.Data pre-processing involves extracting, combining, and normalizing noteworthy features from the raw content sample. The proposed model is used to generate and develop across a number of phases, including training, assessment, and testing, for the forecasting of real-time workload. The final prediction structure reviews and evaluates data on resource utilisation, energy consumption, and memory usage, among other things, to give efficient resource management recommendations. The following is a full explanation of the GEEWC.

3.1Workload Dataset Description

The “Internet of Medical Things (IoMT) [21], MiBench [22], Spark [23], and EEMBC” [24] workload datasets, which are suited for embedded systems, are three separate workload benchmarks that are included in this study.

3.2 Data Pre-processing

The input datasets are initially processed and fed through a normalization procedure that helps to convert the majority of characteristics with numeric values to a finite numeric range. The notion of linear transformation is used to implement one hot encoding in order to do this. From the initial raw datasets, new pre-processed datasets are created after the pre-processing procedure. For the feature extraction module, these pre-processed data are provided.

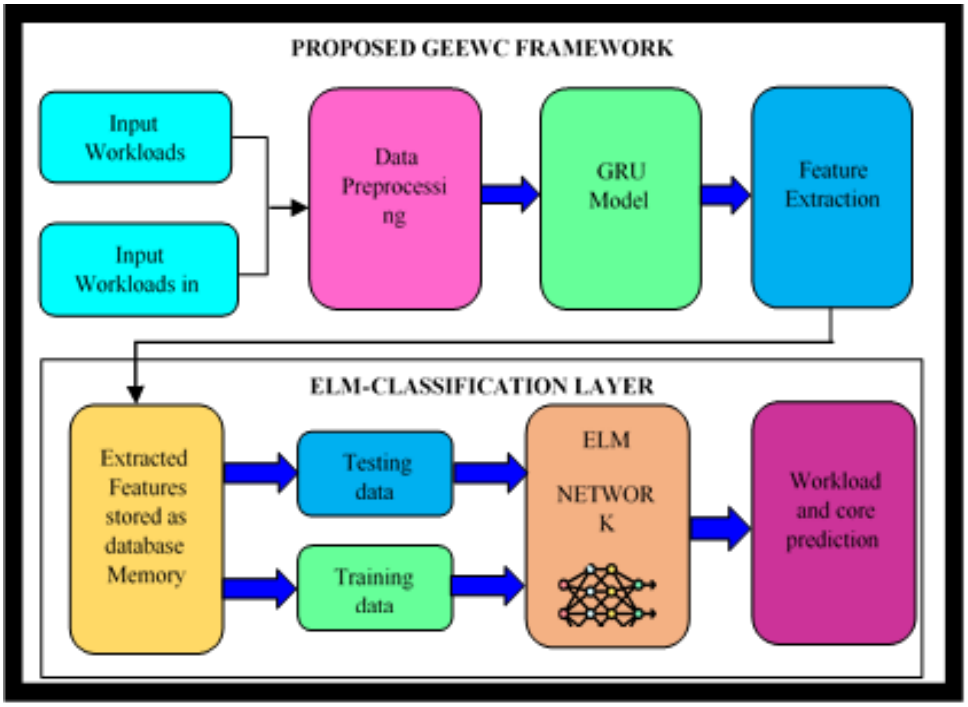

Fig.1. Overall architecture for the proposed deep learning framework deployed for workload characterization.

3.3 Feature Extraction Phases

The GRU model is utilized for feature extraction technique are discussed in this section.

3.3.1Gated Recurrent Units (GRU)

LSTM is thought to have a particularly fascinating variety, called GRU. This concept, which tries to merge the forget gate and input vector as a single vector [25]. In addition to supporting extended memories, this network also supports long term sequences. Comparing the complexity to the LSTM network, it is drastically decreased. The Figure 2 illustrates the GRU architecture.

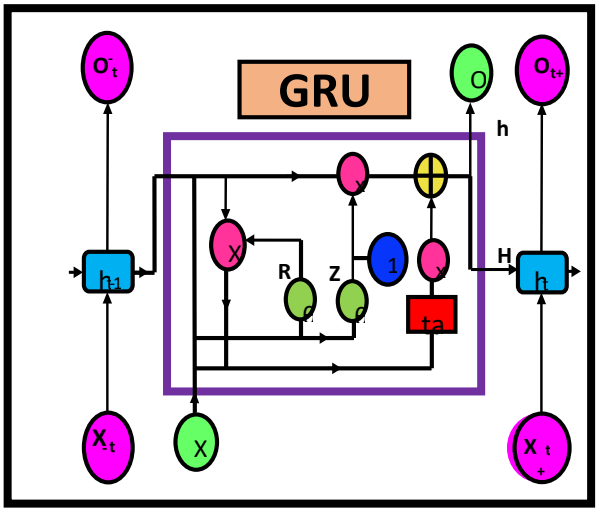

Fig. 2. GRU -network Architecture

Following equations are coined to represent the characteristics of GRU

$$h_t = (1 - x_t) \odot h_{t-1} + x_t \odot h_t \quad (1)$$

$$\widetilde{h_t} = g(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h \quad (2)$$

$$z_t = (W_h x_t + U_z h_{t-1} + b_z) \quad (3)$$

$$r_t = (W_h x_t + U_r h_{t-1} + b_r) \quad (4)$$

The overall GRU characteristic equation is represented by

$$P = GRU(\sum_{t=1}^{n}[x_t, h_t, z_t, r_t(W(t), B(t), \eta(tannh))] \quad (5)$$

where "xt→ input feature at the current state , yt→ output state , ht→ output of the module at the current instant, Zt and rt is update and reset gates, W(t) is weights,B(t) is bias weights at the current instant". The extracted feature are fed into ELM model for the prediction of workloads in embedded systems.

3.4 Classification layers

Once the fully linked feed forward network has received these features, the final classification is performed. Extreme Learning Machines (ELM) theory is used to create the completely linked layers.A specific type of neural network called an ELM uses one hidden layer and operates on the idea of auto-tuning. When compared to alternative learning models such as "SVM, Bayesian Classifier (BC), KNN, and even Random Forest (RF)", ELM displays superior performance, high speed, and little computing cost.

A single hidden layer is used in this type of neural network, and the hidden layer tuning is not strictly

necessary. ELM has superior performance, fast speed, and little computational cost when compared to other learning algorithms like SVM and RF. To produce excellent accuracy for greater speed, ELM employs the kernel function. The ELM's key advantages are improved estimation and less training error. ELM uses automated weight bias adjustment and non-zero activation functions. The ref [26] outlines the technical mechanism for operation of the ELM. The following is a representation of the ELM's input features maps following attention maps:

$$X = (Y) \quad (6)$$

Where Y is the features from DSA network ,

The ELM output function given as follows

$$Y(n) = X(n)\beta = X(n)X^T(\frac{1}{c}XX^T)^{-1}O \quad (7)$$

The ELM's overall training is given by

$$S = \alpha(\sum_{n=1}^{N}\big(Y(n), B(n), W(n)\big) \quad (8)$$

For the feedforward layers discussed above, softmax activation layers are then used to get the greatest accuracy. The meta-heuristic BAT method is used to adjust the feed forward layers' hyperparameters in order to improve performance and decrease training networks' complexity.

3.5. Bat optimized Feed Forward Layers

The echolocation or bio-sonar traits of microbats were utilized by conventional bats to calculate certain things. Ref [27] developed the bat computation with a further 3 extravagant suggestions considering the results of the cancellation of echo computations.

1. All bats use echoes to discriminate among things, but they also seem to "know" how to differentiate between food/prey and environmental obstructions.

2. Bats search for prey by flying erratically at a speed of vi, in a position of xi, with a repetition of fmin, a variable length of wavelength, and a loudness of A0. In light of this, they can modify the signal that is sent pulse's length (or recurrence) and pulse rate of emission (r 2 [0, 1]) based on how near they are to their target.

Although it could vary based on a lot of circumstances, we estimate that the loudness will decrease from a big (positive) A0 to a small constant Amin.

Each bat Motion is associated with the "velocity v_i^t and initial distance x_i^t with the 'n' number of iterations" in a dimensional space or search space. The three aforementioned rules will determine which bat, out of all the bats, is the greatest. The updated $v_i^t$and initial distancexit using the three rules are given below

$$fi = fmin + (fmax - fmin)\,\beta \quad (9)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (10)$$

Where $\beta \in (0,1)$ fmin represented the minimum frequency of 0 and fmax represents the maximum frequency. Initial frequency assignment for each bat is a range between fmin and fmax. So the measurement of bats might be seen as a frequency tuning computation that offers a good balance between study and utilisation. In essence, the levels of emissions and volume serve as a system for controlled monitoring and automatic expanding into surrounding areas with useful information.

In order to provide superior solutions, it is essential that there be a variety of volume and pulse output. Since Amin = 0 indicates that a bat has newly located its target and has temporarily ceased communicating any signals, any calculation of tolerance between Amin and Amax may be used to determine the loud. This is due to the fact that a bat's loudness normally drops after it locates its meal, but its pulse rate of emission often rises.

3.5.1 Advantages of BAT Algorithm:

BAT algorithms have the following main advantages [28]:

1. Greater Efficiency Compared to "PSO, GA, and Other Heuristic Algorithms"

2. More rapid and adaptable space for searching

Motivated by these benefits, optimization of hyperparameters in Feed Forward layers are designed based on BAT algorithm.

3.6 Bat optimized Feed Forward Classification Layers:

The feed forward classifier networks' weights are optimised using the basic bat method, as was covered in Section 3.5. The primary term utilised to optimise the feed forward network's hyperparameters in this instance is the bat's prey hunting process. Considered to be the network's hyperparameters are the "input weights, hidden layers, epochs, and learning rates". The classification network is first given a random selection of these hyperparameters. Equation (11) serves as the basis for the fitness function. Using equations (7) and (8), hyperparameters are determined for each iteration. When the fitness function and equation (11), the iteration comes to an end. Algorithm-1 provides the full operating mechanism.

$$\boldsymbol{FitnessFunction}\ (\mathrm{FF}) \leq \mathbf{1} - ((\mathbf{1} - (\boldsymbol{A})) + (\mathbf{1} - (\boldsymbol{P})) + (\mathbf{1} - \boldsymbol{M}(\boldsymbol{F}))) \quad (\mathbf{1}1)$$

Step	Algorithm-2 //Pseudo Code for the Proposed GEEWC
1	Inputs : Raw datasets from the three workbench

2 Outputs : Prediction of Embedded Resources

3 Three Embedded Workload data collection

4 Construct the GRU Model using Equation (5)

5 Predict the resource management mechanism using Equation(8) and (9)

6 Fitness Function is calculated by using the Equation(11)

7 If fitness function === Equation(11)

8 Go to Step 12

9 Else

10 Go to Step 5

11 If (output value <=1)

12 //Normal Resource usage is Predicted

13 Otherwise check for (output <=2 and output >1)

14 // Medium Resource is Predicted

15 Otherwise check for (output <=3 and output >2)

16 // Heavy Resource is predicted

17 Else

18 Jump to Step 07

19 Stop

20 Stop

21 stop

4. Section-4

For the purpose of evaluating the proposed model, the experiments and ablation studies are presented in this part. Also covered and shown in the part before are results analysis.

4.1 Experimentation

The suggested model was put into practise using an NVIDIA Embedded JETSON nano board with CORTEX architectures constructed using Keras Libraries and Tensorflow v. 2.1 as the backend. The formula for calculating the performance measures is shown in Table 1. To show how much superior the suggested model is, we also computed the AUC and confusion matrix. Early pausing is used to address the problems of overfitting and generalization. This method is used to halt an iteration when the proposed model's validation performance exhibits no improvement over an extended period of time. Training and testing employ an evenly distributed set of data in order to overcome the problem of class imbalance. The mathematical technique for calculating the performance indicators necessary to evaluate the recommended model is shown in Table 3.

Table 1. Performance indicators for evaluation

SL.NO	Performance Metrics	Mathematical Expression
01	Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
02	Recall	$\frac{\text{TP}}{\text{T P+FN}}$ x100
03	Specificity	$\frac{TN}{TN+FP}$
04	Precision	$\frac{TN}{TP+FP}$
05	F1-Score	$2.\frac{Precison * Recall}{Precision + Recall}$

"TP ⇨ True positive values, TN ⇨ True negative values, FP ⇨ False positive & FN ⇨ False negative"

4.2 Results and its Discussion:

In this section, the performance metrics stated in table 3 were used to compare the proposed framework against the remaining sophisticated deep learning models. This study and comparison employed existing deep learning architectures such DNN [25], CNN [26], ELM [27], and GRU [28]. In terms of workload categorization and prediction, these algorithms are thought of as recent models. Since these methods already exist, this study focuses on them, and the models used in this research are trained using the three different workload benchmarks indicated in Section 3.2. That each model is trained under identical experimental settings was important to note. With test data, the trained models are verified and assessed. Datasets were divided in each scenario into two parts: 30% for testing and 70% for training.

Figure 3 provides the ROC Curve for the proposed framework for analyzing the three different data sets. The performance of the recommended model in forecasting the resources using the three distinct datasets is shown in Figure 3 as having been the best. 88% of the suggested model's detection accuracy was found in the 30% of testing data. The region of convergence curves from Figure 3's Figure 3 have been used to demonstrate this. From Figure 3, where the Area Under Curve (AUC) for

workload-based resource prediction is determined to be 0.877, it is evident that the proposed model has successfully exhibited the uniform properties.

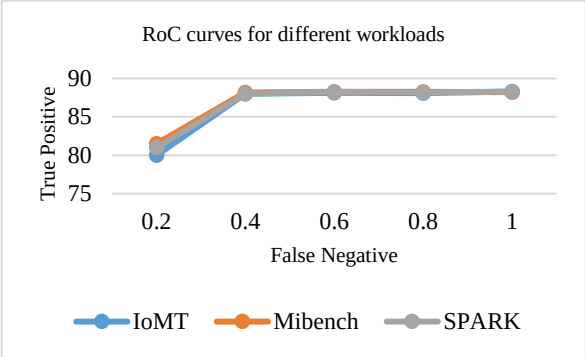

Fig. 3. The suggested model's ROC curves a)IoMT Workbench b) Mi-Bench d) SPARK Workloads

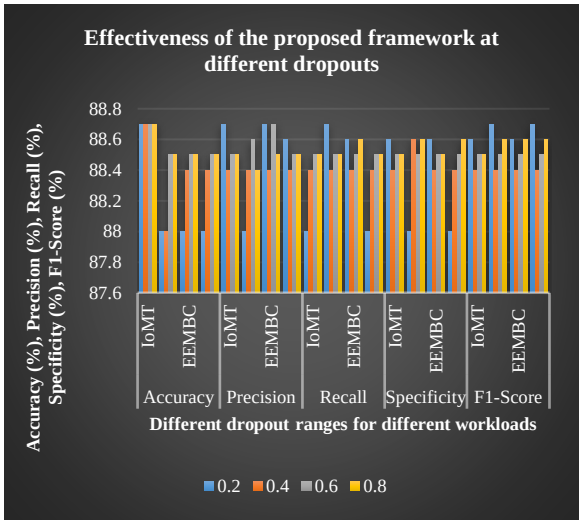

Fig. 4. Using the IoMT, MiBench, EEMBC, and SPARK workloads, the recommended framework's average effectiveness at Dropout was 0.20, 0.40, 0.60, and 0.80 (learning rate = 0.001)

Benchmark workloads were used to inform the implementation of the proposed system, which was done to confirm the effectiveness of the model. The performance measures as depicted in Figure 4 were used to inform all of the research. The datasets are split into training, testing, and validation groups for each iteration of a section. In order to optimize the hyper parameter of the proposed classification network, the bat Optimization technique is used, as was previously mentioned. Dropouts fall between a search range of 0.2 to 0.4. The evaluation is performed on the values 0.6 and 0.8 with a constant learning rate of 0.001. A certain number of epochs, starting at 50, 100, and 150, were also present. Figure 4 makes it abundantly evident from all the data that the recommended technique demonstrated its highest performance of 88.7% at dropouts of 0.2 and 0.4 and same sort of performance at higher drop-outs of 0.6 and 0.8, respectively. This assessment makes it abundantly evident that the proposed network is stable in managing the diverse and dynamic workloads even at higher drop-out rates. The proposed network now has high speed fuel to handle the changing workloads thanks to the implementation of the bat optimization algorithm.

4.3 Comparative analysis:

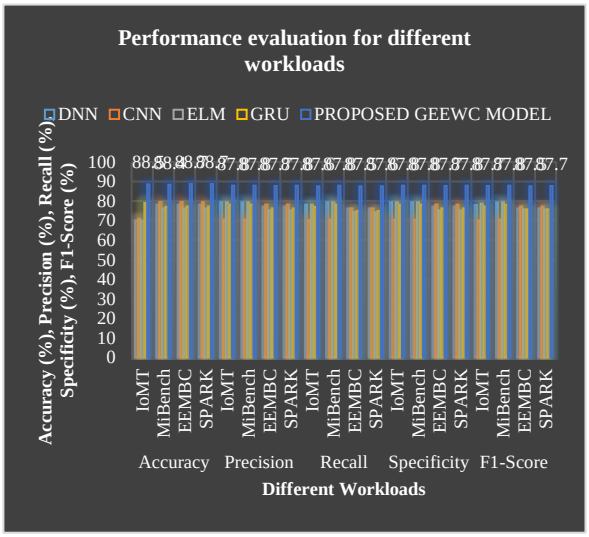

Fig. 5. The typical performance of several deep learning models while handling diverse workloads as IoMT, MiBench, EEMBC, and SPARK.

With the use of 4 different workbenches, Figure 5 shows how well different deep learning frameworks forecast resources. Figure 5 shows that whereas other deep learning models indicate deteriorated performance as the number of datasets increases (Mi-Bench, EEMBC, SPARK), the recommended technique has shown clean and consistent performance. For processing medium-sized (IoMT) datasets, the proposed model performed better than earlier deep learning techniques. Thus, it can be shown that the existing model has produced good outcomes in estimating the resources for managing the dynamic and diverse input workloads.

4.4 Model Building Time Analysis:

Table 2 lists the length of time required to develop a model for various deep learning frameworks. The main driving force for choosing MBT is the conviction that it is essential to consider how long a model must train since this will directly affect how many resources are utilized as well as how well the model performs in predicting the resources using three distinct datasets. MBT helps to achieve a good trade-off between processing complexity and classifier efficacy as a result. The suggested model's average MBT for training the various datasets is 7.3 seconds, according to the above table, compared to 9.3 seconds for DNN, 9.23 seconds for CNN, 12.1 seconds for ELM, and 11.3 seconds for GRU.As a result of the investigation, it is clear that the suggested model uses just

7.03 seconds and excels at generating resource predictions by analyzing workload.

Table 2. Different Deep Learning-based Workload Characterization Techniques: MBT Analysis

Workload Datasets	Model Building Time Analysis(secs)				
	DNN	**CNN**	**ELM**	**GRU**	**Proposed Model**
IoMT Workloads	9.5	9.2	9.5	9.2	6.1
EEMBC Workloads	9.2	9.3	12.5	12.8	7.1
SPARK Workloads	9.2	9.2	14.3	12.0	7.9
Average MBT	**9.3**	**9.23**	**12.1**	**11.3**	**7.03**

5. Conclusion

The strong gated recurrent neural networks and Extreme Learning Networks for prediction were combined in this article's hybrid deep learning framework, known as GEEWC. The feed forward networks' hyper-parameters are modified using the bat optimization to simplify the system and cope with changing embedded workloads. Significant experimentation is carried out using the three benchmarks, and the performance indicators accuracy, recall, precision, specificity, and f1-score have been calculated and assessed. To show how effective the proposed model is, its prediction performance is compared to that of existing deep learning models in use. TensorFlow and Kera's, two Python-based libraries, are used to implement the whole environment, which is installed on an embedded system based on the NVIDIA JETSON NANO architecture. When managing dynamic and heterogeneous workloads, the results show that the recommended model has outperformed alternative deep learning techniques. The XAI (explainable artificial intelligence) paradigm may be used to develop future workload prediction algorithms with traceable processes, which will help to characterize accuracy, transparency, fairness, and anticipate results in AI management of resources. The efficacy of the proposed prediction models can further be improved by offering a more lightweight optimization strategy and reducing its computing cost.

References

[1] J. Lee and H. Bahn, "Analyzing Memory Access Traces of Deep Learning Workloads for Efficient Memory Management," 12th International Conference on Information Technology in Medicine and Education (ITME), Xiamen, China, 2022, pp. 389-393, 2022 doi: 10.1109/ITME56794.2022.00090.

[2] S. Hsia, U. Gupta, M. Wilkening, C. -J. Wu, G. -Y. Wei and D. Brooks, "Cross-Stack Workload Characterization of Deep Recommendation Systems," 2020 IEEE International Symposium on Workload Characterization (IISWC), Beijing, China, 2020, pp. 157-168, doi: 10.1109/IISWC50251.2020.00024.

[3] X. Li et al., "LongTail-Bench: A Benchmark Suite for Domain-Specific Operators in Deep Learning," 2022 IEEE International Symposium on Workload Characterization (IISWC), Austin, TX, USA, 2022, pp. 282-295, doi: 10.1109/IISWC55918.2022.00032.

[4] Z. Quan, X. Chen and Y. Han, "AIC-Bench: Workload Selection Methodology for Benchmarking AI Chips," 2022 IEEE 24th IntConf on High Performance Computing & Communications; 8th IntConf on Data Science & Systems; 20th IntConf on Smart City; 8th IntConf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Hainan, China, 2022, pp. 687-694, doi: 10.1109/HPCC-DSS-SmartCity-DependSys57074.2022.00117.

[5] A. A. Awan, A. Jain, C. -H. Chu, H. Subramoni and D. K. Panda, "Communication Profiling and Characterization of Deep Learning Workloads on Clusters with High-Performance Interconnects," 2019 IEEE Symposium on High-Performance Interconnects (HOTI), Santa Clara, CA, USA, 2019, pp. 49-53, doi: 10.1109/HOTI.2019.00025.

[6] A. A. Awan, A. Jain, C. -H. Chu, H. Subramoni and D. K. Panda, "Communication Profiling and Characterization of Deep-Learning Workloads on Clusters With High-Performance Interconnects," in IEEE Micro, vol. 40, no. 1, pp. 35-43, 1 Jan.-Feb. 2020, doi: 10.1109/MM.2019.2949986.

[7] M. Barve, S. Sinha, R. P. Hardikar, A. Gunturu and W. Mallik, "Workload Characterization in HPC Environment for Auto-scaling of Resources – Preliminary Study," 2022 IEEE 19th India Council International Conference (INDICON), Kochi, India, 2022, pp. 1-6, doi: 10.1109/INDICON56171.2022.10040124.

[8] J. Liu, J. Liu, W. Du and D. Li, "Performance Analysis and Characterization of Training Deep

Learning Models on Mobile Device," 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), Tianjin, China, 2019, pp. 506-515, doi: 10.1109/ICPADS47876.2019.00077.

[9] Q. Hu, P. Sun, S. Yan, Y. Wen and T. Zhang, "Characterization and Prediction of Deep Learning Workloads in Large-Scale GPU Datacenters," SC21: International Conference for High Performance Computing, Networking, Storage and Analysis, St. Louis, MO, USA, 2021, pp. 1-15.

[10] A. F. Inci, M. MericIsgenc and D. Marculescu, "DeepNVM: A Framework for Modeling and Analysis of Non-Volatile Memory Technologies for Deep Learning Applications," 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 2020, pp. 1295-1298, doi: 10.23919/DATE48585.2020.9116263.

[11] Jiwoo Bang, Chunyong Kim, Kesheng Wu, Alex Sim, Suren Byna, Sunggon Kim, and HyeonsangEom. 2020. HPC Workload Characterization Using Feature Selection and Clustering. In 3rd International Workshop on Systems and Network Telemetry and Analytics (SNTA '20), June 23, 2020, Stockholm, Sweden. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3391812. 3396270.

[12] Y. Ren, S. Yoo and A. Hoisie, "Performance Analysis of Deep Learning Workloads on Leading-edge Systems," 2019 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS), Denver, CO, USA, 2019, pp. 103-113, doi: 10.1109/PMBS49563.2019.00017.

[13] J. E and V. R, "Application Workload Characterization using BAT_LSTM Learning algorithm for Asymmetric Architectures," 2021 Emerging Trends in Industry 4.0 (ETI 4.0), Raigarh, India, 2021, pp. 1-5, doi: 10.1109/ETI4.051663.2021.9619290.

[14] A. Khan, X. Yan, S. Tao and N. Anerousis, "Workload characterization and prediction in the cloud: A multiple time series approach," 2012 IEEE Network Operations and Management Symposium, Maui, HI, USA, 2012, pp. 1287-1294, doi: 10.1109/NOMS.2012.6212065.

[15] Sebastian Ştefan and Virginia Niculescu, "Microservice-Oriented Workload Prediction Using Deep Learning", In e-Informatica Software Engineering Journal, vol. 16, no. 1, pp. 220107, 2022. DOI: 10.37190/e-Inf220107.

[16] Krishan Kumar, K. Gangadhara Rao, Suneetha Bulla, D Venkateswarulu, "Forecasting of Cloud Computing Services Workload using Machine Learning," Turkish Journal of Computer and Mathematics Education, Vol.12, No.11, pp. 4841-4846, 2021.

[17] R.Sivaramakrishnan, and Dr.G.Senthilkumar. "LICHA –WORKLOAD CHARACTERIZATION USING LSTM BASED INTELLIGENT CLASSIFICATION FOR HETEROGENEOUS ARCHITECTURES." Turkish Journal of Physiotherapy and Rehabilitation Vol.32,No.3,pp. 250-261(2021).

[18] Ahamed, Zaakki, Maher Khemakhem, FathyEassa, FawazAlsolami, and Abdullah S. Al-Malaise Al-Ghamdi. 2023. "Technical Study of Deep Learning in Cloud Computing for Accurate Workload Prediction" Electronics 12, no. 3: 650. https://doi.org/10.3390/electronics12030650.

[19] KumarK. Dinesh and UmamaheswariE.. "HPCWMF: A Hybrid Predictive Cloud Workload Management Framework Using Improved LSTM Neural Network" Cybernetics and Information Technologies 20, no.4 (2020): 55-73. https://doi.org/10.2478/cait-2020-0047

[20] Om, Khandu, Spyros Boukoros, AnupiyaNugaliyadde, Tanya McGill, Michael Dixon, PolychronisKoutsakis, and KokWai Wong. "Modelling email traffic workloads with RNN and LSTM models." Human-centric Computing and Information Sciences 10, no. 1 (2020). Gale Academic OneFile (accessed March 19, 2023). https://link.gale.com/apps/doc/A634355033/AONE?u=tel_oweb&sid=googleScholar&xid=ebf8a9c2.

[21] C. I. Lee, M. Y. Lin, C. L. Yang and Y. K. Chen: Iotbench: A benchmark suite for intelligent internet of things edge devices. IEEE International Conference on Image Processing. 170-174 (2019).

[22] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge and R. B. Brown: MiBench: a free, commercially representative embedded benchmark suite. Proceedings of the Fourth Annual IEEE International Workshop on Workload Characterization. 3-14 (2001).

[23] A. J. Awan, M. Brorsson, V. Vlassov and E. Ayguade: Micro-architectural characterization of apache spark on batch and stream processing workloads. IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social

Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom). 59-66 (2016).

[24] https://github.com/eembc

[25] J. C. Heck and F. M. Salem, "Simplified minimal gated unit variations for recurrent neural networks," 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 2017, pp. 1593-1596, doi: 10.1109/MWSCAS.2017.8053242.

[26] M. Zhou and W. Ai, "Distributed Reduced Kernel Extreme Learning Machine," 2021 China Automation Congress (CAC), Beijing, China, 2021, pp. 3384-3387, doi: 10.1109/CAC53003.2021.9728238.

[27] D. Singh, R. Salgotra and U. Singh, "A novel modified bat algorithm for global optimization," 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 2017, pp. 1-5, doi: 10.1109/ICIIECS.2017.8275904.

[28] A. H. Ahmadi and S. K. Y. Nikravesh, "A novel instantaneous exploitation based bat algorithm," 2016 24th Iranian Conference on Electrical Engineering (ICEE), Shiraz, Iran, 2016, pp. 1751-1756, doi: 10.1109/IranianCEE.2016.7585804.

[29] J. Bi, S. Li, H. Yuan, Z. Zhao, and H. Liu: Deep neural networks for predicting task time series in cloud computing systems. IEEE International Conference on Networking, Sensing and Control. 86–91 (2019).

[30] U. U. Hafeez and A. Gandhi, "Empirical Analysis and Modeling of Compute Times of CNN Operations on AWS Cloud," 2020 IEEE International Symposium on Workload Characterization (IISWC), Beijing, China, 2020, pp. 181-192, doi: 10.1109/IISWC50251.2020.00026.

[31] Qinghao Hu, Peng Sun, Shengen Yan, Yonggang Wen, and Tianwei Zhang. Characterization and prediction of deep learning workloads in large-scale GPU datacenters. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '21). Association for Computing Machinery, New York, NY, USA, Article 104, 1–15. 2021 https://doi.org/10.1145/3458817.3476223

[32] Milan Jain, Sayan Ghosh, and Sai PushpakNandanoori. Workload characterization of a time-series prediction system for spatio-temporal data. In Proceedings of the 19th ACM International Conference on Computing Frontiers (CF '22). Association for Computing Machinery, New York, NY, USA, 159–168 2022. https://doi.org/10.1145/3528416.3530242.

[33] Chinthamu, N. ., Gooda, S. K. ., Shenbagavalli, P. ., Krishnamoorthy, N. ., & Selvan, S. T. . (2023). Detecting the Anti-Social Activity on Twitter using EGBDT with BCM. International Journal on Recent and Innovation Trends in Computing and Communication, 11(4s), 109–115. https://doi.org/10.17762/ijritcc.v11i4s.6313

[34] Chaudhary, D. S. . (2021). ECG Signal Analysis for Myocardial Disease Prediction by Classification with Feature Extraction Machine Learning Architectures. Research Journal of Computer Systems and Engineering, 2(1), 06:10. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/12

[35] Keerthi, R. S., Dhabliya, D., Elangovan, P., Borodin, K., Parmar, J., & Patel, S. K. (2021). Tunable high-gain and multiband microstrip antenna based on liquid/copper split-ring resonator superstrates for C/X band communication. Physica B: Condensed Matter, 618 doi:10.1016/j.physb.2021.413203