energies

Article

Optimized Hierarchical Tree Deep Convolutional Neural Network of a Tree-Based Workload Prediction Scheme for Enhancing Power Efficiency in Cloud Computing

Thirumalai Selvan Chenni Chetty [1,*], Vadim Bolshev [2,*], Siva Shankar Subramanian [3], Tulika Chakrabarti [4], Prasun Chakrabarti [5], Vladimir Panchenko [6], Igor Yudaev [7] and Yuliia Daus [7]

[1] Department of Computer Science and Engineering, Gitam School of Technology, Gitam University, Bengaluru 561203, Karnataka, India
[2] Laboratory of Power Supply and Heat Supply, Federal Scientific Agroengineering Center VIM, 109428 Moscow, Russia
[3] Department of Computer Science and Engineering, KG Reddy College of Engineering and Technology, Chilkur Village, Moinabad Mandal, RR District, Hyderabad 501504, Telangana, India
[4] Department of Chemistry, Sir Padampat Singhania University, Udaipur 313601, Rajasthan, India
[5] Department of Computer Science and Engineering, ITM SLS Baroda University, Vadodara 391510, Gujarat, India
[6] Department of Theoretical and Applied Mechanics, Russian University of Transport, 127994 Moscow, Russia
[7] Energy Department, Kuban State Agrarian University, 350044 Krasnodar, Russia
* Correspondence: thirumalaiselvan027@outlook.com (T.S.C.C.); vadimbolshev@gmail.com (V.B.)

check for updates

Citation: Selvan Chenni Chetty, T.; Bolshev, V.; Shankar Subramanian, S.; Chakrabarti, T.; Chakrabarti, P.; Panchenko, V.; Yudaev, I.; Daus, Y. Optimized Hierarchical Tree Deep Convolutional Neural Network of a Tree-Based Workload Prediction Scheme for Enhancing Power Efficiency in Cloud Computing. *Energies* **2023**, *16*, 2900. https://doi.org/10.3390/en16062900

Academic Editor: José Antonio Domínguez-Navarro

Received: 17 February 2023
Revised: 10 March 2023
Accepted: 16 March 2023
Published: 21 March 2023

Abstract: Workload prediction is essential in cloud data centers (CDCs) for establishing scalability and resource elasticity. However, the workload prediction accuracy in the cloud data center could be better due to noise, redundancy, and low performance for workload prediction. This paper designs a hierarchical tree-based deep convolutional neural network (T-CNN) model with sheep flock optimization (SFO) to enhance CDCs' power efficiency and workload prediction. The kernel method is used to preprocess historical information from the CDCs. Additionally, T-CNN model weight parameters are optimized using SFO. The suggested TCNN-SFO technology has successfully reduced excessive power consumption while correctly forecasting the incoming demand. Further, the proposed model is assessed using two benchmark datasets: Saskatchewan HTTP traces and NASA. The developed model is executed in a Java tool. Therefore, associated with existing methods, the developed technique has achieved higher accuracy of 20.75%, 19.06%, 29.09%, 23.8%, and 20.5%, as well as lower energy consumption of 20.84%, 18.03%, 28.64%, 30.72%, and 33.74% when validating the Saskatchewan HTTP traces dataset. It has also achieved higher accuracy of 32.95%, 12.05%, 32.65%, and 26.54%.

Keywords: cloud data center; cloud computing; convolutional neural network; sheep flock optimization; workload prediction; kernel correlation

1. Introduction

A modern computing model is used for business, marketing, academia, research, and online transactions. It offers all customers flexible and independent pay-per-use IT resources and pandemic computing services [1]. To sustain constant service availability and maintain service level agreement (SLA) standards, the scalability of IT resources is unquestionably a crucial characteristic of the cloud [2]. Therefore, effective resource management procedures are essential to the continued operation of all data centers [3]. Additionally, fine-grained resource allocation is necessary for the dynamic scheduling process of jobs such as the sub-units of user applications. Therefore, an effective workload detection system is crucial for CDCs to achieve these aims [4,5].

Furthermore, the precise workload forecast approach can efficiently address CDC resource management decisions [6]. However, the workload prediction method typically concentrates on various difficulties, such as unpredictable changes in workload prediction that are openly linked to the redistribution and distribution of resources [7]. It can result in both overestimating and underestimating resources [8]. Additionally, unneeded and unreliable information causes inaccurate forecasts with substantial computing costs [9]. In recent years, many scholars from around the world have become interested in the workload forecasting issue in CDCs. A time series modeling approach accompanied by a hidden Markov model (HMM) [10], autoregressive moving average (ARMA) technique [11], dynamic resource provisioning (DRP) [12], or autoregressive integrated moving average (ARIMA) technique [11] are some of the standard workload prediction models proposed in recent times. However, an effective prediction model is still required to minimize the prediction error [13]. An essential component of effective cloud resource management is workload estimation. The precise workload forecast method can efficiently address the cloud data center's resource management decisions [14,15]. However, the workload is poorly projected by the current approaches, which are computationally expensive [16]. It improves the CDC's energy efficiency. The present method cannot estimate the prediction of workload and increasing energy efficiency of the CDC with any degree of accuracy [17]. The most common problems in workload prediction models are as follows:

- Low energy efficiency;
- Poor efficiency;
- Poor forecasting;
- High energy cost;
- Large prediction error; and
- High energy consumption.

A tree-based workload prediction model is developed to address these problems, accurately predicting workload and improving energy efficiency in CDCs while also improving the forecasting results and enhancing efficiency.

The following is a summary of this manuscript's primary contributions:

- This publication suggests using a T-CNN optimized with an SFO-based workload prediction model to improve the power effectiveness of CDCs.
- Initially, redundant data and noise in the historical data obtained from the CDC are filtered using the Kernel Correlation (KCR) approach [18].
- These preprocessed historical data are given to the T-CNN model to forecast workload in a dynamic cloud environment, producing the forecasted workload as an output [19].
- To improve the weight parameters of the T-CNN model in this work, an SFO was devised [20].
- Finally, the suggested TCNN-SFO approach decreases excessive power consumption in CDCs while precisely predicting the incoming workload.
- The proposed approach is implemented as a Java tool, and the effectiveness of the suggested TCNN-SFO-based workload prediction model is assessed using assessment metrics.
- This publication undertakes a series of comparative tests by experimenting with two benchmark datasets, namely Saskatchewan HTTP traces and NASA [21], to show the efficacy of the suggested method. The performance of the presented model is then evaluated associated with prevailing technologies.

The sections of this document are as follows. Section 2 of the document describes the literature survey. Then, the proposed model is described in Section 3, and Section 4 shows the results. Finally, the manuscript is concluded in Section 5.

2. Related Works

Recent studies based on workload prediction in the cloud are evaluated below.

The auto adaptive differential evolution (AADE) approach for predicting workload in the dynamic cloud surroundings was presented by Saxena et. al. in 2020 [22]. The information came from the benchmark dataset, which was used for additional processing. Using the AADE approach, the prediction model familiarizes and absorbs traces of the workload for particular prediction periods from past data. The model's results also showed that the AADE model outperformed techniques in accuracy by 98%, 97%, and 94%, respectively. However, the AADE model's energy efficiency is improved to anticipate cloud workload.

A supervised learning (SL) with neural network (NN) model for estimating workload in CDCs was presented by Kumar J. et al. in 2020 [23]. The adaptive differential evolution (ADE) model is created to improve the effectiveness of the prediction model. The most excellent crossover result and mutation operators are attained with the help of this model. As a result, the NN-SL model outperformed the ADE approach by 91%. However, the NN-SL's workload forecasting accuracy was inferior.

Banerjee et al. [24] introduced the concept in 2021 of multi-step-ahead workload prediction using machine learning techniques. This model's resource allocation was done using a prediction model, allowing for more effective and lower energy usage. Additionally, the suggested framework outperforms alternative strategies for long-term workload prediction, greatly enhances resource use, and enables significant energy savings. However, compared to other models, multi-step-ahead could have improved efficiency.

A usage prediction of aware-based virtual machine (VM) methodology for achieving energy-efficient CDCs was presented by Hsieh et al. in 2021 [25]. The dynamic VM method is frequently used in large CDCs to minimize energy consumption. Here, the technique for VM provided took into account resource consumption by detecting the host underload and host overload. Additionally, the given model was used to simulate real-world workloads in Cloudsim and validated using benchmark techniques. Unfortunately, although this approach improved energy consumption and service quality, the prediction accuracy fell short of expectations.

An online multi-resource neural network (OM-NN) proactive and energy-efficient VM allocation model for CDCs was presented by Saxena et al. in 2021 [26]. The feed-forward neural network (FFNN) is used to anticipate numerous resources for future applications. This approach contains two other principles: first, VMs are automatically scaled, and second, scaled VMs are owed based on the anticipated model's energy efficiency. That said, while estimating the resources in the CDC, the given integrated model had a significant prediction error.

To improve prediction accuracy, Kumar et al. [27] presented the error preventive score (EPS) based on time series prediction techniques in 2020. The proposed EPS model was used to reduce prediction errors and produce more accurate forecasts. Additionally, computations such as magnitude of predictions (MoP) and predictions in error range (PER) are used to determine the accuracy value. Further, these are verified using three approaches for workload estimation and five data traces. As a result, the EPS model could predict workload more accurately by utilizing time series forecasting methods, albeit at a significant energy cost.

Lin et al. proposed the workload-aware power consumption measuring (WAPCM) technique for CDCs in 2021 [28]. The WAPCM approach used workload grouping, prediction, and classification to select the best power model for the incoming workload proactively. The results of the WAPCM model attained better results for reducing the lag in power estimation. Moreover, the accuracy value of the designed model on server requests traces and power consumption using WAPCM was high. However, the WAPCM model had a significant prediction error and high energy consumption.

3. Proposed TCNN-SFO Methodology

The workload prediction model is created to increase cloud efficiency and minimize energy consumption. However, the current models could be achieving a higher performance level. Therefore, this study proposed a T-CNN with SFO for improving power consumption and workload prediction. The suggested TCNN-SFO technique's primary goal is forecasting workload to increase CDCs' power efficiency. The information is gathered from two benchmark datasets, Saskatchewan HTTP traces and NASA HTTP traces, both of which have undergone kernel correlation approach preprocessing. Additionally, the CDC workloads were accurately predicted using the T-CNN model, and the parameters were improved using the SFO technique, increasing the cloud's energy efficiency.

3.1. Data Acquisition

NASA and Saskatchewan HTTP traces are two benchmark datasets used in this study. In addition, Google cluster traces are used to pick the CPU data traces and memory resource requests in this case.

3.2. Preprocessing Using KCR Method

The KCR method is employed in this work to preprocess the dataset to reduce undesired noise. The KCR technique reduces the redundant data and noise present in the dataset. The redundant information and noise in the historical data from the CDC are filtered using the KCR model. Figure 1 depicts the suggested TCNN-SFO model's procedures. The designed model used KCR functions, which enhance the data quality and improve the accuracy of the designed model. Additionally, the developed T-CNN model has several nodes arranged in a tree-like pattern to make up a tree-CNN. Every node (aside from leaf nodes) has a DCNN that is trained to categorize the input into one of the node's children. The first classification takes place at the root node, which is the tree's lowest node. According to the classification label, the image is subsequently sent to its child node. Up until we reach a leaf node, the final stage of classification, this node further categorizes the image. Branch nodes are intermediary nodes that have two or more children and at least one parent. The leaf node is the tree's highest level. No two leaf nodes have the same class, which is uniquely associated with each leaf node. In this layer update, the SFO fitness optimizes the T-CNN parameters and improves the weight limits using a fitness function to enhance the performance of workload prediction in the cloud.

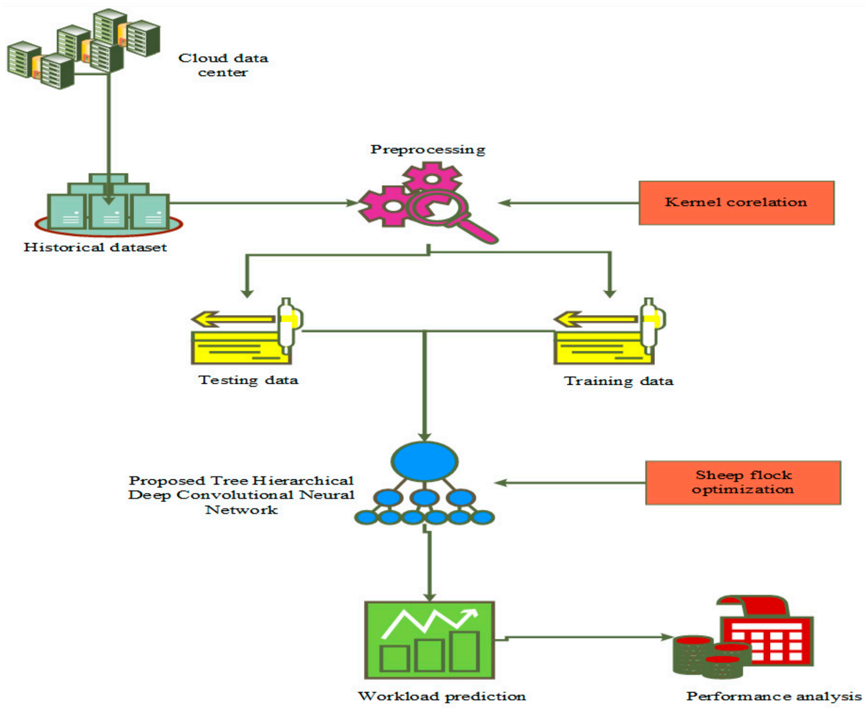

Figure 1. Process of the proposed methodology.

In this instance, the KCR model successfully tracks the dataset by substituting features for pixels. Let the overall samples present in the dataset be denoted as $O = \{1, 2, 3......., n\}$, with the $k_e \times k_e$ kernel matrix (K) declared in Equation (1),

$$K_{ij} = k_e(s_i, s_j) \tag{1}$$

where k_e is the factor of kernel correlation and (s_i, s_j) is the row and column of particular samples. The redundant data in the samples is eliminated using Equation (2) in the suggested kernel correlation model.

$$\alpha = (K + \phi I)^{-1} y \tag{2}$$

where K is the kernel matrix, α is the vector coefficient of samples that denotes the solution of redundant data, y denotes kernel correlation parameters, and ϕ represents the kernel correlation factor that can diminish the redundant data in the samples. As a result, the KCR procedure removes unnecessary information from the data.

3.3. Workload Forecasting Using T-CNN

The NN model is utilized to predict the treasured data in the training input data samples. In the meantime, the CDC's workload appearance is dynamic. As a result, the T-CNN model, which draws its inspiration from hierarchical classifiers, is proposed. In this instance, the T-CNN model comprises numerous nodes associated in a tree-like fashion. Additionally, all nodes in the model, except the leaf nodes, include DCNN that have been qualified to classify the input samples to the node hooked on the offspring nodes of the tree. Additionally, the categorization brand of the child node is applied to subsequent steps, which are carried out until the tree's leaf node. As a result, the leaf node outcome of the secondary branch node makes up the branch NN's output node.

Initially, the trained parameters of the T-CNN model are denoted as $O = \{1, 2, 3......., n\}$ with n-amount of data.

The NN in this suggested architecture includes numerous layers and functions, such as a root node with many leaf nodes. Additionally, the task is specified to the technique for class load prediction. The lesser section of data from the training set can be supplied via the model's root node. Three-dimensional matrices are provided by the T-CNN model's output node, which is referred to as $O^{K \times A \times B}$. Let K represent the total number of root node children, A denote the number of new classes (workloads), and B represent the sample data for each class. In addition, $O(k, a, b)$ is called forecast load output of the kth neuron for the ath data to the bth load. In this equation, the value of kth neuron belongs to [1, K], bth load belongs to [1, A], and ath data belong to [1, B]. Thus, the regular predicted output of the A dataset is stated as $O_{avg}^{K \times A}$, which is calculated using Equation (3):

$$O_{avg}^{K \times A}(k, a) = \sum_{a=1}^{A} (O(k, a, b)) / A \tag{3}$$

In addition, the probability of the softmax function is calculated over $O_{avg}^{K \times b}$ and the possibility matrix $R^{K \times b}$ is divided using Equation (4),

$$R(k, b) = \left[\frac{e^{O_{avg}(k,b)}}{\sum_{k=1}^{K} e^{O_{avg}(k,b)}} \right] \tag{4}$$

where the possibility matrix is denoted as $R^{K \times b}$, and $e^{O_{avg}(k,b)}$ signifies the arithmetical content used to improve power efficiency. Afterward, the ordered list (L) of the workload, which is produced from $R^{K \times b}$, includes certain assets. Here, the ordered list (L) has data samples $[B_1, B_2, B_3]$ of each data b loads. Furthermore, the workload output values are organised in decreasing order as $[B_1 \geq B_2 \geq B_3]$. As a result, the ordering is finished to ensure that the leaf node of the T-CNN approach receives the projected workload with a high probability value. The performance of the designed technique is assessed using the training approach after comparing the actual and anticipated workload of the model. The training process is resumed when the mistake score is reduced to the desired level. The optimization procedure was necessary for the T-CNN model to enhance the network weight connections. The SFO model improves the estimation of the T-CNN parameters by enhancing the weight limits.

3.4. Sheep Flock Optimization (SFO)

The SFO is suggested in this study as a method for improving the weight limits of the T-CNN approach. Here, the proposed SFO approach is a set of two move-section and grazing-section meta-heuristic mechanisms that are inspired by nature. The previous experience of the sheep, the shepherd's advice, and the desire of the sheep to model other sheep are all factors in the movement phase of this mechanism. Additionally, the grazing process has been repeated following a few repetitions of the moving stage. So, sheep and goats are used to study behaviors such as exploitation and exploration.

Step 1: Initialization

This step starts optimizing the sheep flock's population size, maximum iterations, dimensions, and cost function. In the suggested method, the benchmark dataset's processes are created in the SFO optimization to increase parameter effectiveness. Additionally, the weight parameter variables are initialized to achieve the optimal solution described in Equation (5):

$$S = \left(A, O_{avg}^{K\times B}(k,a), R(k,b)\right) \tag{5}$$

where $O_{avg}^{K\times B}$ denotes the average predicted output of the A data, and $R^{K\times b}$ represents probability matrix.

Step 2: Random Generation

Following initialization, each member's placements are determined at random. The population is divided into segments so that each individual can be assigned to sheep or goats. Additionally, to determine the best value, the values of random placements are contrasted with earlier values. Equation (6) is used to determine how much is added to the operation's starting point.

$$P_x = \left(2 \times R_{g(S,G)}\right) \times rand - R_{g(S,G)} \tag{6}$$

where $R_{g(S,G)}$ is defined as the browsing radius of sheep (S) and goats (G), and *rand* represents the random value that lies among [0, 1]. Thus, the browsing radius of sheep (S) and goats (G) is designed using Equations (7) and (8):

$$R_{g(S)} = 0.001 \times (u_b - l_b) \times T \tag{7}$$

$$R_{g(G)} = 0.1 \times (u_b - l_b) \times T \tag{8}$$

where u_b represents the upper bound, l_b denotes the lower bound, and T is the number of iterations mentioned in Equation (9),

$$T = 1 - [iteration\_no/iteration] \tag{9}$$

Step 3: Calculating Fitness Function

The fitness function is created to find the best solutions to issues. This study uses this function to optimize the T-CNN model's parameters. To achieve the optimal answer, the initialization functions, such as the probability matrix and the average anticipated output of the data from Equations (3) and (4), are optimized here. Then, each iteration of the model's core operations is chosen using the grazing function, as shown in Equation (10).

$$iteration\_no \geq (iteration) - 5 \tag{10}$$

$$V = S\sum_{x=1}^{N}(P_x)T \quad (11)$$

where V indicates the optimum fitness function based on the number of iterations required to achieve the ideal parameter setting.

Step 4: Search for a new location

The cost value was better than the initial cost, and the sheep were encouraged to the new site in this stage. The moving part of this stage calculates the shepherd's overall best option, which aids the flock in locating a new home. The shepherd's order determines the optimal position in the move section based on the movement origin, which is computed using an Equation (12),

$$V_{Sp,1} = (1-T)C.rand(1, O_{im})\left(G_{gbest} - G\right) \quad (12)$$

where G denotes the current location of sheep, G_{gbest} represents the best fitness value, $(1, O_{im})$ is the random array that lies between the range [0, 1], and $C = 3 * rand$ and $V_{Sp,1}$ represent the shepherd order based on movement arising. Additionally, the movement that results from the sheep's interest in their most recent best practice is determined by Equation (13),

$$V_{Lbest,1} = C.rand(1, O_{im})(G_{Lbest} - G) \quad (13)$$

where G_{Lbest} represents the best fitness value, $(1, O_{im})$ is the random array that lies between the range [0, 1], and $V_{Lbest,1}$ represents the interest of the sheep in the preceding best experience. Additionally, Equation (14) is used to determine the movement caused by a sheep's desire to approach other sheep.

$$V_{other,1} = C.rand(1, O_{im})(G_{rand.S} - G) \quad (14)$$

where $G_{rand.S}$ represents a random generation of sheep's location and $V_{other,1}$ represents the sheep's interest in approaching other sheep when the number of iterations is $T > 0.3$.

Step 5: Update the location to optimize the parameters

The value of herd dispersion is reduced as $T \leq 0.3$, based on the sheep movement that is affected by two conditions, which are,

- The progression from the shepherd's order to the ideal pasture, and
- The movement results from the sheep's interest in their most recent positive experience.

Thus, Equation (15) is used to determine the movement that proceeds from the shepherd's command to the ideal place.

$$V_{Sp2,1} = C1(1-T)\left(G_{gbest} - G\right) \quad (15)$$

Additionally, Equation (16) determines the movement caused by the sheep's interest in their most recent positive experience.

$$V_{b,1} = V\left\{V_{Sp1,1} + V_{Lbest,1} + V_{other,1}T > 0.3V_{Sp2,1} + V_{Lbest,1}T \leq 0.3\right\} \quad (16)$$

where $V_{m,1}$ represents the calculated speed based on sheep movements. Additionally, Equation (17) moves the sheep from their present position to their next place.

$$G_{Iteration+1} = V(G_{Iteartion} + V_{b,1}) \quad (17)$$

Finally, the position of the sheep is updated by the speed of sheep movements. The value of $V_{b,1}$ with fitness value V is utilized to obtain the best solution by optimizing parameters. Therefore, the optimal value of the model has been attained.

Step 7: Termination

The sheep flock optimization technique is utilized in this case to continuously optimize the parameters of the prediction outcomes until the best result is obtained. The results of the sheep flock optimization technique finally give the T-CNN's parameters their excellent value.

4. Results and Discussion

The effectiveness of the suggested TCNN-SFO model is assessed in this section. Two benchmark datasets [21] are used to analyze the performance of the proposed approach. This methodology's simulation uses a Java tool, which provides practical tools for precise results. Based on performance measures including precision, accuracy, recall, correlation coefficient (CoC), energy consumption, mean squared prediction error (MPE), sum of elasticity index (SEI), and predictions in error range (PER), the TCNN-SFO approach achieved better results. The simulation parameters for the TCNN-SFO model are shown in Table 1.

Table 1. Simulation factors.

Factors	Value
Task length	1–2000
Memory VM	250–2048
Priority tasks	High, medium, and low
The total quantity of tasks	120
Number of virtual machines	50
Number of data centers	1

The performance of the proposed TCNN-SFO methodology is compared with other techniques, such as the AADE model [22], error preventive score (EPS) in time series forecasting models (EPS-TSF) [27], bi-phase adaptive learning-based NN (BALNN) [23], TSF methods for cloud data workload prediction (TSF-CDWP) [29], and the self-directed workload forecasting (SDWF) method [30].

4.1. Dataset Description

This study used two benchmark datasets, Saskatchewan HTTP traces and NASA, to validate the results [21]. Two paths from the NASA dataset have two months' worth of HTTP requests. The Saskatchewan HTTP traces dataset has seven months' worth of HTTP logs. Since there are 672,075 jobs and 48 million tasks, the paths are kept in ASCII files. The remaining data are taken for testing purposes, with 50% used for training.

4.2. Performance Metrics

Regarding accuracy, recall, precision, CoC, energy consumption, SEI, MPE, and PER, the suggested TCNN-SFO mechanism's effectiveness has been assessed (PER).

4.2.1. Accuracy

The accuracy is determined using the formula indicated in Equation (18) and is calculated to determine the efficacy of the proposed TCNN-SFO method for workload prediction in the cloud system.

$$A_c = \frac{t_P + t_n}{t_P + f_P + t_n + f_n} \tag{18}$$

Let t_P denote a true positive, f_P symbolize a false positive, t_n represent a true negative, and f_n denote a false negative.

4.2.2. Precision

The precision is calculated to assess the effectiveness of the suggested approach when used with training data that are assessed using Equation (19),

$$p_r = \frac{t_P}{t_P + f_P} \tag{19}$$

4.2.3. Recall

Recall computation is defined as the effectiveness of the suggested technique for minimizing power usage, while workload prediction is determined using Equation (20).

$$R_e = \frac{t_P}{t_P + f_n} \tag{20}$$

4.2.4. Energy Consumption

Energy consumption is needed for workload detection from CDCs and is calculated by Equation (21),

$$Energyconsumption = E_T + N * E_R \tag{21}$$

Here, N represents the normal number of neighboring nodes for the communicating node, E_T denotes the communicating energy value, and E_R is defined as the received energy value.

4.2.5. CoC

CoC is described as the arithmetical relationship computation or the design of the grade of changes among the actual and expected workload. As a result, Equation (22) is used to compute the value of CoC.

$$CoC = c_{or}\left(k_{at}, k_{pt}\right) \tag{22}$$

Let c_{or} be the correlation function, and k_a, k_p be the actual workload and predicted workloads, respectively, at the time period t.

4.2.6. SEI

The suggested prediction approach is supported by the computation of SEI, which is done using Equation (23),

$$SEI = \sum_{t=1}^{n} \frac{\left(k_{at}, k_{pt}\right)}{\left(k_{at}, k_{pt}\right)} \tag{23}$$

where (n) denotes the total number of predictions. Additionally, the SEI falls between [0, 1], which indicates the best and worst workload predictions, respectively.

4.2.7. MPE

The suggested model is well-defined as having great accuracy when the MPE total score is close to zero and is calculated using an Equation (24).

$$MPE = 1/n \sum_{t=1}^{n} \left(k_{at} - k_{pt}\right)^2 \tag{24}$$

4.2.8. PER

The workload forecast is the percentage prediction error (PER), which is calculated using Equation (25).

$$ppe_T = \frac{\left|\left(k_{at} - k_{pt}\right)\right|}{k_{at}} \times 100 \tag{25}$$

4.3. Comparative Analyses of Existing Models

The proposed TCNN-SFO strategy compares the efficiency with other existing methods, such as the AADE algorithm [22], BALNN [23], EPS-TSF [27], TSF-CDWP [29], and SDWF [30].

Figure 2 displays a comparison of accuracy calculations obtained from paper [20,21,25,28,29]. At a prediction interval of 20 min, the suggested TCNN-SFO technique has achieved higher accuracy than the prevailing BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF approaches by 21.3%, 43.6%, 24.6%, 23.65%, and 54.64%, respectively. The proposed method outperformed existing methods with accuracy rates of 20.97%, 41.3%, 23.9%, 21.34%, and 51.09%, respectively, at a prediction interval of 40 min. The suggested method outperformed existing methods in accuracy by 19.73%, 31.6%, 20.9%, 31.53%, and 21.64%, respectively, at a prediction interval of 60 min. The TCNN-SFO approach has outperformed existing methods in accuracy by 19.46%, 30.92%, 21.45%, 13.94%, and 20.6%, respectively, at prediction intervals of 80 min. By validating at a prediction interval of 100 min, the suggested model's accuracy value is 20.64%, 32.95%, 12.05%, 32.65%, and 26.54% higher than the existing approaches, respectively. As a result, the suggested TCNN-SFO strategy has a more excellent accuracy value than the already used approaches, which include BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF, respectively.

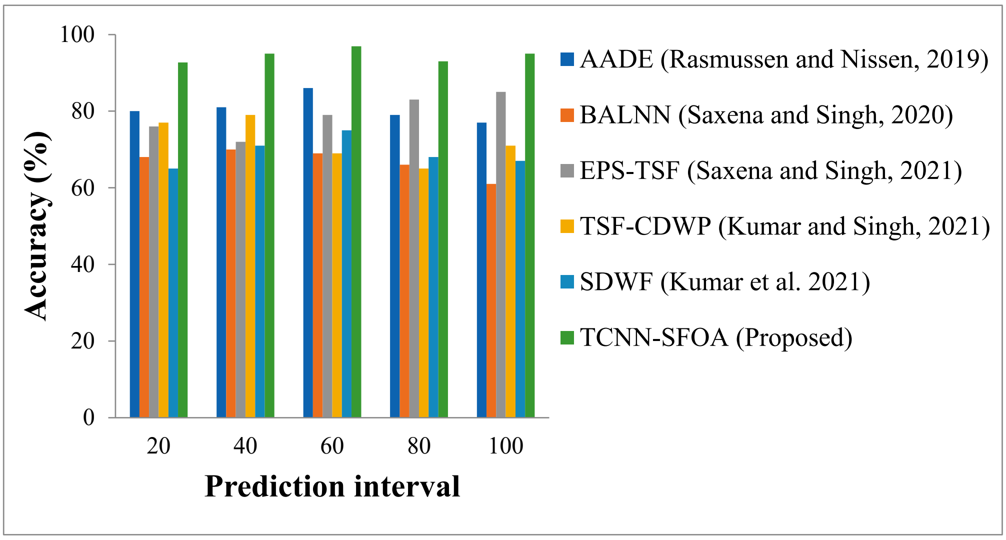

Figure 2. Accuracy comparison [20,21,25,28,29].

Figure 3 displays a comparison of precision calculations obtained from papers [20,21,25,28,29]. Compared to the current BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF techniques, the suggested TCNN-SFO methodology has achieved high precision at 18.9%, 32.5%, 19%, 30.9%, and 13.75%, respectively, at prediction interval 20 min. The proposed method outperformed existing methods in high precision by 17.8%, 38.9%, 12.6%, 21.6%, and 20.9%, respectively, at a forecast interval of 40 min. At a prediction interval of 60 min, the suggested method outperformed existing methods in high precision by 17.65%, 27.8%, 18.7%, 28.6%, and 18.6%, respectively. The TCNN-SFO approach has outperformed previous algorithms in high precision by 25.6%, 32.5%, 15.8%, 21.3%, and 16.7%, respectively. at prediction intervals of 80 min. By validating at a prediction interval of 100 min, the suggested model's precision value is 32.6%, 34.5%, 17.8%, 28.9%, and 19.6% higher than that of the existing approaches, respectively. As a result, the suggested TCNN-SFO strategy has a more excellent precision value than the already used approaches, which include BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF, respectively.

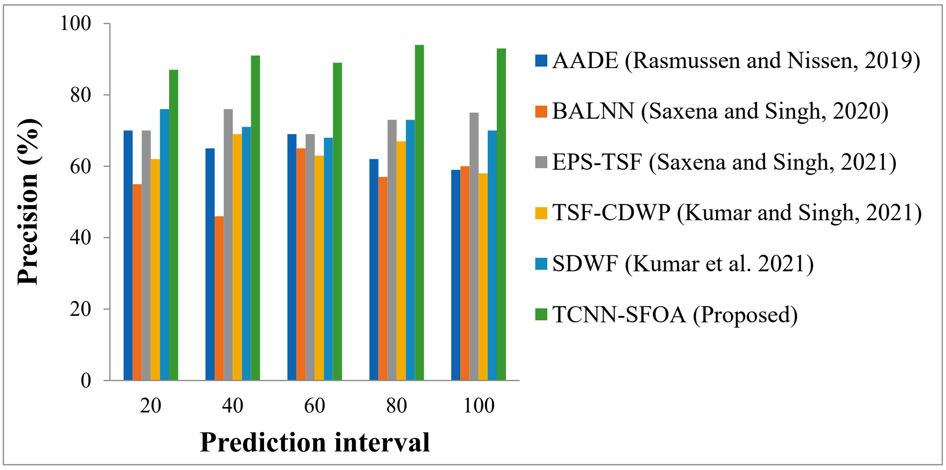

Figure 3. Precision comparison [20,21,25,28,29].

Figure 4 displays a comparison of recall calculations [20,21,25,28,29]. At a prediction interval of 20 min, the suggested TCNN-SFO technique has achieved higher recall than the current BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF approaches by 26.7%, 25.4%, 19.7%, 12.6%, and 16.5%, respectively. The planned method outperformed prevailing methods in terms of recall by 12.7%, 29.7%, 26.5%, 11.75%, and 21.79%, respectively, at a prediction interval of 40 min. The suggested strategy outperformed previous approaches in terms of recall by 29.6%, 29.7%, 25.6%, 17.67%, and 35.6%, respectively, at a prediction interval of 60 min. The TCNN-SFO strategy outperformed previous approaches in the recall by 23.4%, 34.57%, 13.7%, 20.9%, and 19.87%, respectively, at a prediction interval of 80 min. Consequently, validating at a prediction interval of 100 min, the suggested model's recall value is 21.4%, 19.6%, 23.6%, 20.9%, and 19.87% higher than the existing approaches, respectively. Therefore, compared to existing systems including BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF, the suggested TCNN-SFO methodology has a more excellent recall value.

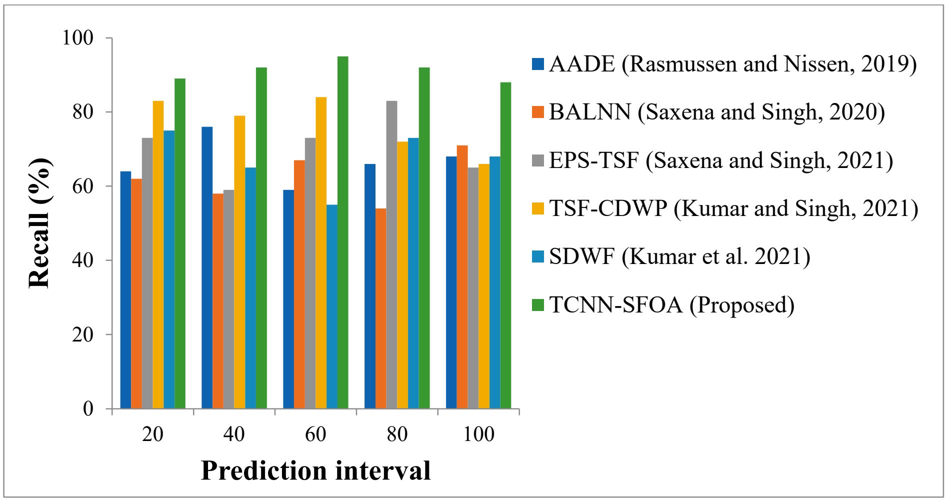

Figure 4. Recall comparison [20,21,25,28,29].

Figure 5 displays a comparison of energy usage calculations [20,21,25,28,29]. The suggested TCNN-SFO strategy has achieved lower energy usage at prediction intervals of 20 min than the current BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF approaches by 35%, 32.6%, 38.7%, 26.84%, and 55.9%, respectively. In addition, the proposed method has achieved reduced energy usage than existing methods by 27.8%, 31.5%, 37.6%, 29.2%, and 33.8%, respectively, at the 40-min forecast interval.

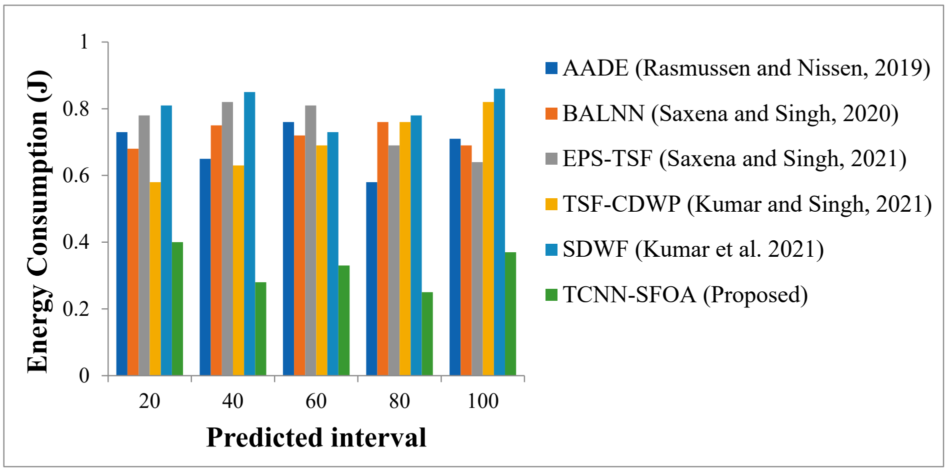

Figure 5. Energy consumption comparison [20,21,25,28,29].

The proposed method has achieved lower energy usage than existing methods by 25.6%, 24.7%, 29.95%, 23.7%, and 31.2%, respectively, at a prediction interval of 60 min. The TCNN-SFO approach has achieved lower energy usage than existing systems by 19.8%, 27.57%, 26.7%, 29.4%, and 31.8%, respectively, at prediction intervals of 80 min. Furthermore, the suggested model has achieved lower energy consumption than the current approaches by 27.4%, 26%, 23.7%, 34.7%, and 36.5%, respectively, at prediction intervals of 100 min. Therefore, compared to existing systems including BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF, the suggested TCNN-SFO approach uses less energy.

Figure 6 displays a comparison of CoC calculations [20,21,25,28,29]. At a prediction interval of 20 min, the suggested TCNN-SFO strategy has achieved higher CoC than the current BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF techniques by 64.7%, 54.6%, 53.9%, 33.6%, and 27.8%, respectively. The proposed method has achieved higher CoC than previous methods by 54.6%, 55%, 35.6%, 32.95%, and 21.7%, respectively, at a prediction interval of 40 min. The proposed strategy outperformed prevailing approaches in terms of CoC by 24.5%, 22.7%, 20.95%, 32.6%, and 29.8%, respectively, at a prediction interval of 60 min. The TCNN-SFO approach outperformed previous approaches by 34.6%, 56.7%, 29.67%, 33.6%, and 27.6%, respectively, at a prediction interval of 80 min. By validating at a prediction interval of 100 min, the suggested model's CoC value is 32.7%, 24.5%, 29.86%, 19.6%, and 21.7% higher than the existing approaches, respectively. As a result, the suggested TCNN-SFO strategy has a higher CoC value than the already used approaches, such as BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF.

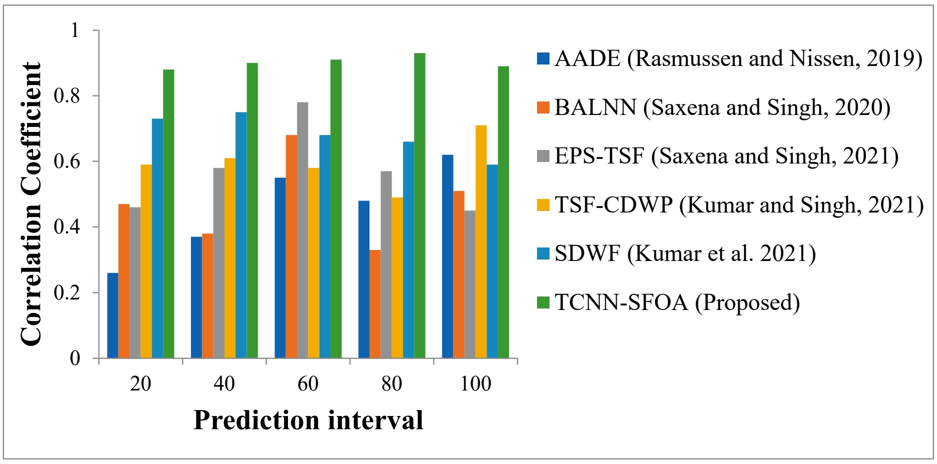

Figure 6. CoC comparison [20,21,25,28,29].

Figure 7 displays a comparison of SEI calculations [20,21,25,28,29]. At a prediction interval of 20 min, the suggested TCNN-SFO technique has achieved higher SEI than the current BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF approaches by 23.5%, 14.7%, 24.6%, 12.4%, and 10.9%, respectively. The proposed method outperformed existing methods in terms of SEI at prediction intervals of 40 min by 28.9%, 17.8%, 21.3%, 15.7%, and 13.67%, respectively. The proposed strategy outperformed previous approaches for high SEI by 19.8%, 16.9%, 18.3%, 26.7%, and 8.9%, respectively, at a forecast interval of 60 min. The TCNN-SFO approach has achieved higher SEI than previous approaches by 21.8%, 20.5%, 18.7%, 19.6%, and 12.5%, respectively, at a prediction interval of 80 min. By validating at a prediction interval of 100 min, the suggested model's SEI value is 15.6%, 12.7%, 19.8%, 17.95%, and 21.65% higher than the existing techniques, respectively. Therefore, the suggested TCNN-SFO strategy has a higher SEI value than the already used approaches, such as BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF.

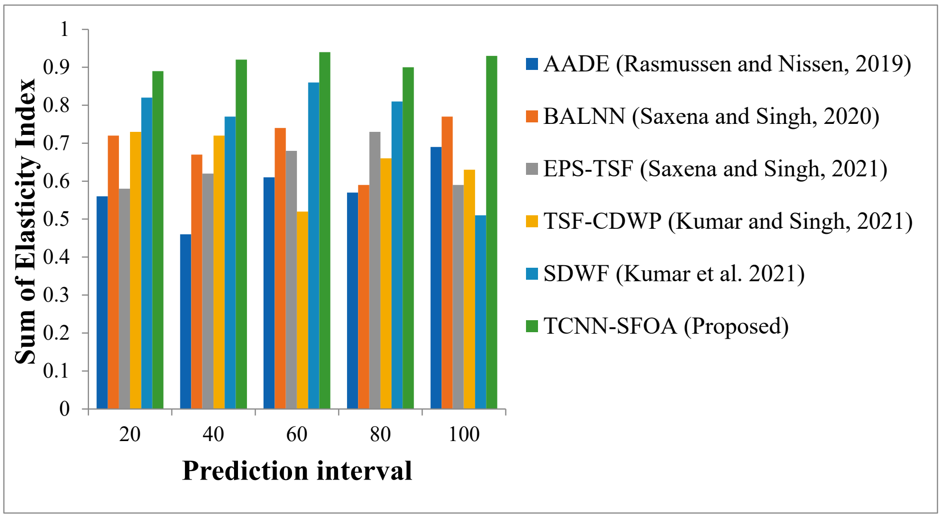

Figure 7. SEI comparison [20,21,25,28,29].

Figure 8 displays a comparison of MPE calculations [20,21,25,28,29]. At a prediction interval of 20 min, the suggested TCNN-SFO technique has achieved less MPE than the prevailing BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF methods by 10.6%, 26.7%, 24.5%, 20.9%, and 8.7%, respectively. The proposed method has achieved less MPE than existing techniques at prediction intervals of 40 min by 9.97%, 24.5%, 29.7%, 21.8%, and 19.87%, respectively. The suggested method outperformed previous methods in terms of low MPE at prediction intervals of 60 min by 8.9%, 12.6%, 17.85%, 21.64%, and 17.85%, respectively. At a prediction interval of 80 min, the TCNN-SFO technique has achieved less MPE than previous methods by 19.45%, 21.56%, 24.9%, 29.07%, and 5.6%, respectively. Consequently, when validating at a prediction interval of 100 min, the suggested model's MPE value is 18.9%, 21.6%, 17.5%, 13.6%, and 29.6% less than the currently used approaches, respectively. Therefore, compared to prevailing methods including BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF, the MPE value of the suggested TCNN-SFO approach is lower.

Figure 9 displays a comparison of PER calculations [20,21,25,28,29]. The suggested TCNN-SFO strategy has outperformed the prevailing BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF techniques by 12.46%, 15.77%, 23.65%, 20.69%, and 21.85%, respectively, at forecast interval 20 min. The suggested method has outperformed previous methods by 13.5%, 11.8%, 23.5%, 19.45%, and 15.6%, respectively, at prediction intervals of 40 min. The suggested way beat earlier approaches in terms of PER at prediction intervals of 60 min by 18.9%, 24.7%, 16.5%, 28.09%, and 17.8%, respectively.

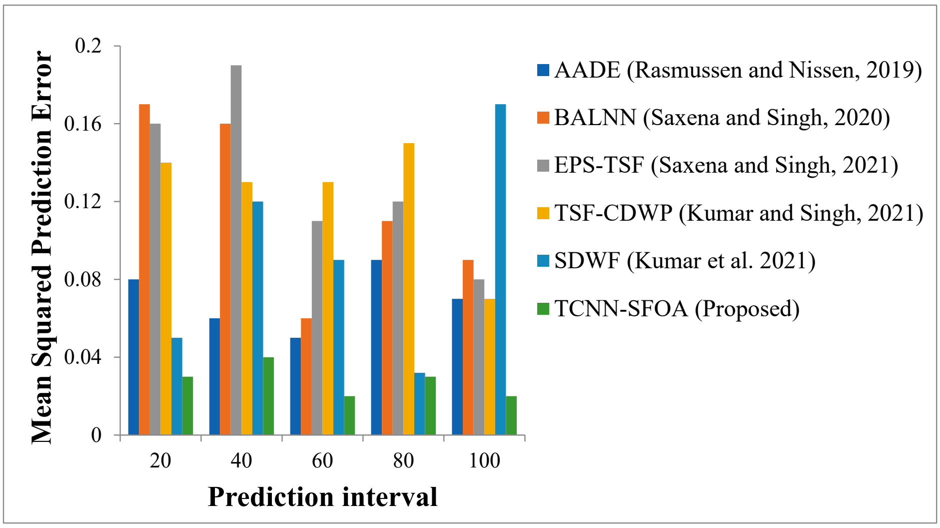

Figure 8. MPE comparison [20,21,25,28,29].

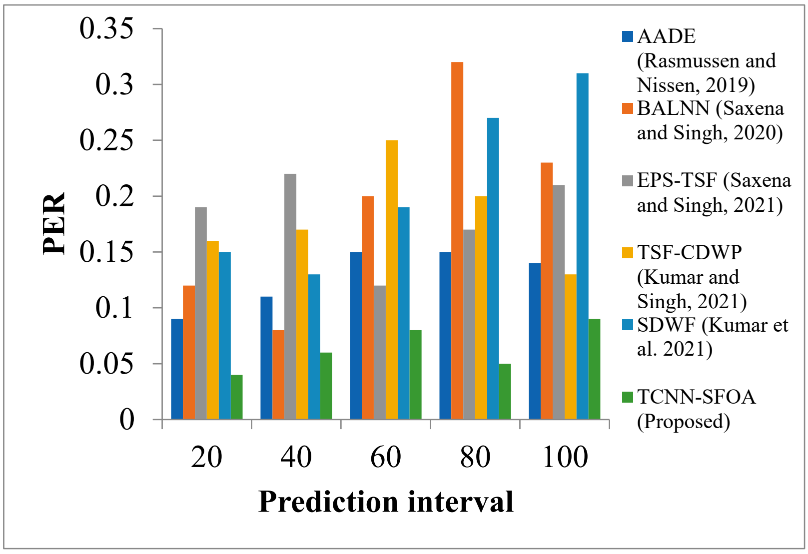

Figure 9. Predictions in error range (PER) comparison [20,21,25,28,29].

The TCNN-SFO strategy has outperformed other approaches by 17.7%, 23.67%, 20.8%, 17.69%, and 28.7%, respectively, at a prediction interval of 80 min. However, by validating at a prediction interval of 100 min, the suggested model's PER value is subsequently 19.59%, 23.07%, 21.98%, 19.9%, and 27.07% lower than the existing approaches, respectively. As a result, the suggested TCNN-SFO strategy has a lower PER value than the already used approaches, such as BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF.

5. Conclusions

Design an SFO-based T-CNN model for workload forecasting in CDCs and improve power efficiency. The dataset used for the research are benchmark datasets, Saskatchewan HTTP traces and NASA. Initially, preprocessing is processed using a kernel correlation approach to remove any redundant data from the database. Then, the suggested T-CNN model detects the workload and adjusts the weight parameters using the SFO. Thus the designed model successfully forecasts the workload in CDCs, enhancing the cloud's power efficiency. The attained results of the developed model are validated with prevailing models including BALNN, AADE, TSF-CDWP, EPS-TSF, and SDWF models. Finally, the suggested TCNN-SFO methodology has obtained 20.64%, 32.95%, 12.05%, 32.65%, and

26.54% higher accuracy, respectively, while using 27.4%, 26%, 23.7%, 34.7%, and 36.5% less energy, respectively, to validate datasets. However, the complexity of the designed model is a high-cost and non-convex problem. In the future, hybrid optimization enhances workload prediction performance and gains better experimental outcomes.

Author Contributions: Conceptualization, T.S.C.C.; methodology, T.S.C.C. and S.S.S.; validation, V.B., T.C. and P.C.; investigation, T.S.C.C. and S.S.S.; data curation, V.B., P.C. and Y.D.; writing—original draft preparation, T.S.C.C.; writing—review and editing, V.B., V.P. and I.Y.; visualization, T.S.C.C. and V.P.; supervision, P.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ding, D.; Fan, X.; Zhao, Y.; Kang, K.; Yin, Q.; Zeng, J. Q-learning based dynamic task scheduling for energy-efficient cloud computing. *Future Gener. Comput. Syst.* **2020**, *108*, 361–371. [CrossRef]
2. Jiang, D.; Wang, Y.; Lv, Z.; Wang, W.; Wang, H. An energy-efficient networking approach in cloud services for IIoT networks. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 928–941. [CrossRef]
3. Sharma, Y.; Si, W.; Sun, D.; Javadi, B. Failure-aware energy-efficient VM consolidation in cloud computing systems. *Future Gener. Comput. Syst.* **2019**, *94*, 620–633. [CrossRef]
4. Zhou, X.; Xu, X.; Liang, W.; Zeng, Z.; Yan, Z. Deep-learning-enhanced multitarget detection for end–edge–cloud surveillance in smart IoT. *IEEE Internet Things J.* **2021**, *8*, 12588–12596. [CrossRef]
5. Devaraj, A.F.; Elhoseny, M.; Dhanasekaran, S.; Lydia, E.L.; Shankar, K. Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments. *J. Parallel Distrib. Comput.* **2020**, *142*, 36–45. [CrossRef]
6. Mohamed, N.; Al-Jaroodi, J.; Lazarova-Molnar, S. Leveraging the capabilities of industry 4.0 for improving energy efficiency in smart factories. *IEEE Access* **2019**, *7*, 18008–18020. [CrossRef]
7. Bhattacherjee, S.; Das, R.; Khatua, S.; Roy, S. Energy-efficient migration techniques for cloud environment: A step toward green computing. *J. Supercomput.* **2020**, *76*, 5192–5220. [CrossRef]
8. Zhang, Y.; Lan, X.; Ren, J.; Cai, L. Efficient computing resource sharing for mobile edge-cloud computing networks. *IEEE/ACM Trans. Netw.* **2020**, *28*, 1227–1240. [CrossRef]
9. Abualigah, L.; Diabat, A. A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments. *Clust. Comput.* **2021**, *24*, 205–223. [CrossRef]
10. Roy, P.P.; Kumar, P.; Kim, B.-G. An efficient sign language recognition (SLR) system using Camshift tracker and hidden Markov model (hmm). *SN Comput. Sci.* **2021**, *2*, 79. [CrossRef]
11. Gao, Y.; Mosalam, K.; Chen, Y.; Wang, W.; Chen, Y. Auto-regressive integrated moving-average machine learning for damage identification of steel frames. *Appl. Sci.* **2021**, *11*, 6084. [CrossRef]
12. Devi, K.L.; Valli, S. Multi-objective heuristics algorithm for dynamic resource scheduling in the cloud computing environment. *J. Supercomput.* **2021**, *77*, 8252–8280. [CrossRef]
13. Priya, V.; Kumar, C.S.; Kannan, R. Resource scheduling algorithm with load balancing for cloud service provisioning. *Appl. Soft Comput.* **2019**, *76*, 416–424. [CrossRef]
14. Attaran, M.; Woods, J. Cloud computing technology: Improving small business performance using the Internet. *J. Small Bus. Entrep.* **2019**, *31*, 495–519. [CrossRef]
15. Alsaidy, S.A.; Abbood, A.D.; Sahib, M.A. Heuristic initialization of PSO task scheduling algorithm in cloud computing. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 2370–2382. [CrossRef]
16. Tuli, S.; Ilager, S.; Ramamohanarao, K.; Buyya, R. Dynamic scheduling for stochastic edge-cloud computing environments using a3c learning and residual recurrent neural networks. *IEEE Trans. Mob. Comput.* **2020**, *21*, 940–954. [CrossRef]
17. Ebadi, Y.; Navimipour, N.J. An energy-aware method for data replication in the cloud environments using a tabu search and particle swarm optimization algorithm. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e4757. [CrossRef]
18. Raghavendra, R.J.; Kunte, R.S. Extended local ternary co-relation pattern: A novel feature descriptor for face anti-spoofing. *J. Inf. Secur. Appl.* **2020**, *52*, 102482. [CrossRef]
19. Sahoo, A.K.; Pradhan, C.; Das, H. Performance evaluation of different machine learning methods and deep-learning based convolutional neural network for health decision making. In *Nature Inspired Computing for Data Science*; Springer: Cham, Switzerland, 2020; pp. 201–212.
20. Benisha, R.B. An efficient Sheep Flock Optimization-based hybrid deep RaNN for secure and enhanced video transmission quality. *Neural Comput. Appl.* **2023**, *6*, 1–16. [CrossRef]

21. Rasmussen, S.; Nissen, J.N. Benchmark Datasets. 2019. Available online: https://figshare.com/articles/dataset/Benchmark_datasets/11409360 (accessed on 20 March 2023).
22. Saxena, D.; Singh, A.K. Auto-adaptive learning-based workload forecasting in dynamic cloud environment. *Int. J. Comput. Appl.* **2020**, *44*, 541–551. [CrossRef]
23. Kumar, J.; Saxena, D.; Singh, A.K.; Mohan, A. Biphase adaptive learning-based neural network model for cloud datacenter workload forecasting. *Soft Comput.* **2020**, *24*, 14593–14610. [CrossRef]
24. Banerjee, S.; Roy, S.; Khatua, S. Efficient resource utilization using multi-step-ahead workload prediction technique in cloud. *J. Supercomput.* **2021**, *77*, 10636–10663. [CrossRef]
25. Hsieh, S.Y.; Liu, C.S.; Buyya, R.; Zomaya, A.Y. Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers. *J. Parallel Distrib. Comput.* **2020**, *139*, 99–109. [CrossRef]
26. Saxena, D.; Singh, A.K. A proactive autoscaling and energy-efficient VM allocation framework using online multi-resource neural network for cloud data center. *Neurocomputing* **2021**, *426*, 248–264. [CrossRef]
27. Kumar, J.; Singh, A.K. Cloud datacenter workload estimation using error preventive time series forecasting models. *Clust. Comput.* **2020**, *23*, 1363–1379. [CrossRef]
28. Lin, W.; Zhang, Y.; Wu, W.; Fong, S.; He, L.; Chang, J. An adaptive workload-aware power consumption measuring method for servers in cloud data centers. *Computing* **2020**, *27*, 1–24. [CrossRef]
29. Kumar, J.; Singh, A.K. Performance assessment of time series forecasting models for cloud datacenter networks' workload prediction. *Wirel. Pers. Commun.* **2021**, *116*, 1949–1969. [CrossRef]
30. Kumar, J.; Singh, A.K.; Buyya, R. Self directed learning based workload forecasting model for cloud resource management. *Inf. Sci.* **2021**, *543*, 345–366. [CrossRef]