iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** X **Month of publication:** October 2022

DOI: https://doi.org/10.22214/ijraset.2022.47136

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

Forecasting Energy Consumption Using Hybrid CNN and LSTM Auto-Encoder Network with Hyperband Optimization

Jyoti Prakash Mohanty[1], Saswat Dash[2]
[1]*Department of Computer Science and Engineering, National Institute of Technology, Rourkela, India*
[2]*Department of Computer Science and Engineering, National Institute of Technology, Rourkela, India*

Abstract: ***Power consumption in the modern world has significantly increased due to the rapid growth of the human population and technological advancement. The precise forecasting of rising electricity usage is a requirement for planning strategies, boosting profits, cutting down on power waste, and ensuring the energy demand management system runs steadily. Modern developments in the area of electricity consumption prediction offer great techniques for capturing chaotic trends in energy consumption and even surpassing forecasting models that have been around for a long time. But to increase prediction accuracy, some significant drawbacks in the current consumption prediction models must be addressed. In this paper, we present a hybrid deep learning design-based Auto-Encoder (AE) network model comprising of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) network for multi-step forecasting. The input dataset is de-noised within the framework using Fast Fourier Transform (FFT) and outliers are removed using Local Outlier Factor (LOF) algorithm, before learning in an unsupervised manner using the auto-encoder network. And, hyperband optimization is applied for tuning the hyperparameters. The outcomes of the numerical experiments demonstrate that the proposed architecture yields excellent prediction capabilities and great potential for generalization.***

Keywords: Energy consumption forecasting, Deep Learning, CNN, LSTM, Auto-Encoders, Fast Fourier Transform, Local Outlier Factor, Hyperband optimization

I. INTRODUCTION

Energy is essential to society's social and economic progress. In recent decades, the increase in electric energy consumption is being accelerated by numerous causes, such as strong economic expansion, population growth, the availability of more energy-intensive appliances and machinery, globalization, and climate change [1]. Unrestrained energy usage like excessive consumption, inefficient architecture, and energy wastage, can also be attributed to these events. The International Energy Agency (IEA) calculates that throughout the years 2016 to 2040, global primary energy ingestion would grow at a rate of 1.0% compounded annually [2]. Due to the physical properties of electrical energy, it ought to be utilized as soon as it is produced in the electricity power plant [3]. As a result, effective energy usage control systems are essential in delivering a consistent energy supply, effective resource consumption, and energy preservation while minimizing operating expenses. Exact energy forecasting, meantime, can aid executives in conducting market exploration and speeding up commercial growth [4]. Demand management, consumption tracking, and demand-supply synchronization are the major services supported by smart grids [5]. To forecast the patterns of energy usage, they are also in charge of keeping a database which tracks energy demand patterns. For managing resources effectively, system dependability, improved planning of forthcoming tactics, planning the construction of grids, and economic investment management, precise forecasting of energy usage is a key step. The need for energy has grown so much as global industrialization and manufacturing have advanced that it is now a significant national policy issue [6]. In the business world and in academia, predicting energy usage is seen as a crucial and difficult job. From an academic perspective, its advancement can be extended to forecasting various time series, for instance, the prediction of stock prices [7], weather [8], and solar radiation [9]. Energy consumption forecasting can be done in three diverse phases, reliant on the necessary time horizon of projection [10], [11]:

1) Long-term (years ahead prediction, ideal for planning the installation of generation systems and forecasting capacity).
2) Medium-term (one-twelve months ahead prediction, good for scheduling and designing reservoirs).
3) Short-term (day-to-week ahead prediction, good for everyday tasks and purchasing decisions).

©IJRASET: All Rights are Reserved

A time-series model could be decomposed as a vector autoregressive model, a univariate model, a multivariate model, and an auto-regressive aggregate moving average model [12]. To forecast energy use, conventional statistical and econometric models are typically applied. These strategies use the Delphi curve fitting method [13] and dynamic behaviour modelling based on physics (white-box methods). In an energy market that is changing at an accelerated rate and necessitates big data analysis of patterns of energy usage and pertinent factors using sophisticated mathematical methods, these models may have drawbacks. Unlike conventional econometric models, data-driven machine learning practices such as k-nearest neighbours, support vector regression [14], artificial neural networks [15], random forests [16], and gradient boosting [17], can efficiently differentiate random components and capture the underlying non-linear features. As a result, it has the advantage of having greater forecast accuracy than the typical time series models and being applicable to considerably wider applications. Without taking into account the internal process, those models construct the data correlation directly from historical statistics. Pano-Azucena et al. forecasted erratic time series deprived of related patterns using support vector machines and artificial neural networks [18]. In order to choose the hyperparameters and kernel parameters which outline the function to be minimized, least squares support vector machines are utilized [19]. Structural learning with forgetting is introduced by Ishikawa et al. which leverages average prediction error or knowledge criteria to create an artificial neural network model with improved generalization potential [20]. The treatment of trivial matches, complexity invariability, offset and amplitude invariance are all presented as improvements to the k-nearest neighbour algorithm [21]. All of the aforementioned methodologies, meanwhile, fall short in their ability to investigate data correlations.

The way the data is represented generally affects how well machine learning algorithms perform. The quantity of hidden layers in a neural network is augmented by deep learning techniques, which then seek to combine a number of non-linear transformations into meaningful expressions that can produce more abstract results and, eventually, greater accuracy [22]. In numerous applications, different deep learning techniques have been widely investigated. Convolutional neural networks (CNN) outperform traditional approaches in image classification, whereas recurrent neural networks (RNN) do exceptionally well in natural language processing and tasks for recognizing speech. CNN extracts abstract visual characteristics from the input data, for instance, points, faces, and lines, and keeps the associations among pixels for the image to be learned. It also gains knowledge of the weights assigned to each layer's feature maps [23]. To properly interpret, represent, and retain the lengthy sequential information, the RNN keeps the facts in hidden memory. For time information to persist, it also refreshes over time [24]. A popular enhanced model of an artificial RNN called Long Short-Term Memory (LSTM) has the capability of extracting sequential or temporal features of data [25]. LSTM presents 3 new gates in each cell, which include the input, output, and forget gates, which are proficient in catching the temporal fluctuations in extremely long sequential input. This is in contrast to original RNN models, which frequently experience problems with exploding or vanishing gradients while sequential data training. As a result, text, video, and time-series analysis have all benefited from their widespread use. For identifying faults in smart factories, Park et al. [26] presented a lightweight and real-time solution using the LSTM model of recurrent neural networks. To increase the efficiency of astute transport systems, Ran et al. [27] offer an approach grounded on LSTM having an attention mechanism for forecasting travel time. Furthermore, authors in [28] designed a neural-encoded mention-hypergraph technique, which can automatically deal with nested as well as overlapping structure mentions and recognize nested structure mention entities.

The experimental findings of numerous investigations have demonstrated that approaches that are hybrid in nature, deliver predictions that are more accurate and dependable than those produced by other current state-of-the-art data-driven methodologies. Many researchers have been working on integrating CNN and LSTM models to retrieve temporal and spatial characteristics in recent years. Authors in [29] used a local CNN-LSTM model with text input to study emotions in the arena of natural language processing. Sainath et al. used a CLDNN model, which combines CNN, LSTM, and DNN, to demonstrate a variety of voice search tasks in the area of speech processing and demonstrate its robustness against noise [30]. Oh et al. employed an architecture integrating LSTM and CNN in the medical field to precisely identify arrhythmias in the ECG [31]. Researchers in [32] created a convolutional bi-directional LSTM network via mechanical vigour monitoring technique to foretell tool wear in an industrial setting. Additionally, several hybrid approaches that integrate deep neural network models and signal processing methods like the wavelet transform and mode decomposition algorithms [33], [34] have drawn a lot of attention for modelling energy consumption. However, they have a number of drawbacks, including a high level of computing complexity, reliance on shifting techniques, the handling of long-term data dependencies, the selection of the best mode component for noise reduction, and others.

Model optimization is amongst the trickiest issues in the use of machine learning solutions. Finding the hyperparameters of a machine learning model that perform best when evaluated against a validation set is the goal of hyperparameter optimization. Contrary to model parameters, the hyperparameters are fixed prior to training. While the weights in a neural network are learned model parameters during training, the quantity of trees in a random forest algorithm is regarded as a hyperparameter.

The process of optimizing hyperparameters aims to discover a set of values of hyperparameters that together produce the best possible model, which lowers a predetermined loss function and, as a result, improves accuracy on given independent data. In the past, hyperparameters were adjusted manually through trial and error. It continues to be a prevalent practice, and researchers may guess choices of parameters that shall produce ML models with extremely high accuracy. The hunt for optimal, quicker, and further automatic ways for hyperparameters optimization is still ongoing. The most rudimentary technique for fine-tuning hyperparameters is perhaps grid search [35]. Using this method, models are created for every possible permutation of the provided hyperparameter values, each model is then evaluated, and finally, the design that yields the best outcome is chosen. To train and score the model, Random Search generates a grid of probable hyperparameter values and picks random permutations [36]. This gives you the ability to specifically regulate the number of parameter combinations that are tried. Based on available time or resources, the number of search iterations is decided. A prime candidate for optimization is random search or grid search as they don't require any tuning, are quite simple to code, and can also be run using parallelism. Their shortcoming is that there is no assurance of getting local minima to some accuracy unless the search space is comprehensively sampled. Bayesian optimization fits in a group of sequential model-based optimization procedures which permit one to utilize the outcomes of preceding iterations to enhance the sampling process of the subsequent experiment [37].

This bounds the number of times model training is required for validation as exclusively those settings that are likely to produce a greater validation score are used for evaluation. It is effective at tuning a small number of hyper-parameters, but as the search dimension rises, it becomes less effective until it is on par with random search. Only continuous hyper-parameters can be optimized using Bayesian methods; categorical ones cannot. A unique configuration evaluation method was developed via framing hyperparameter optimization as a refined-investigation accommodative resource distribution task inscribing the way to disperse resources between randomly selected hyperparameter values [38]. This is known as a hyperband setup. It uses a logical early-stopping resource allocation mechanism that enables it to assess orders of magnitude more arrangements in contrast to black-box techniques similar to Bayesian optimization. With few presumptions, hyperband is a general-purpose tool, in contrast to earlier configuration assessment approaches and it works for both continuous and categorical hyperparameters. The objective in each scenario is to minimize simple regret, which is measured as the distance from the ideal course of action.

Inspired by the advantages of hybrid deep learning techniques, we propose a hybrid deep learning model of CNN and LSTM Auto-Encoder network for energy consumption forecasting. The input dataset undergoes a pre-processing step where absent, redundant, or outlier values are discarded. The data is de-noised using a fast Fourier transform, outliers are removed using the Local Outlier Factor algorithm and the data is also normalized to attain reasonable prediction outcomes. CNNs are used in the encoder part to excerpt spatial features and LSTM is used in the decoder part to model these features for final prediction. The model's hyperparameters are tuned using hyberband optimization technique.

The multi-step predictions are made in an autoregressive manner. Lastly, the time resolution is altered to see if further enhancement is possible for the proposed model. The design of the existing architecture was principally driven by two factors. First, the model training is possible to be carried out in an unsupervised fashion without the need for ground truth data on the network flow time series, statistical characterization, or spatio-temporal distribution. Secondly, the robust learning of hidden non-linear representations and subsequent reconstruction capabilities of LSTM auto-encoders make them useful. Auto-encoders are not limited to dense layers, and LSTM units enable them to grind in generative style with variable-sized input sequences, in contrast to standard neural learning approaches. Various scale-dependent and percentage-error metrics were used in the experiments to equate the proposed model to the state-of-the-art architectures. The experimental findings support the claim that the proposed method surpasses the contemporary methods for forecasting energy usage.

The remainder of the article is as follows. The associated research in the area of electric energy consumption prediction is illustrated in Section 2. The technique of the proposed systematic hybrid methodology is described in depth in Section 3 along with the theoretical underpinning of the fundamental principles used in the current study. The experimental details and results analysis are shown in Section 4. Lastly, conclusions and discussion are offered in the final section.

II. LITERATURE REVIEW

Models for predicting energy consumption involve forecasting future timestamp consumption from a set of historical consumption observations. Accurate estimation is necessary for making a variety of investment decisions, increasing revenue, enhancing the operational stability of the power grid systems, and lowering distribution inefficiency [39]. Since the 1940s, a wide range of strategies has been effectively used to predict electricity usage. These strategies can be grouped basically into four main types which are as follows:

A. Traditional Methods

Deciphering a series of equations to represent the performance of the energy grid system is the foundation of conventional consumption forecast models [13], [40]. These techniques are sometimes known as "white-box approaches" because the core logic is very transparent. For assessing energy usage and preservation strategies centered on system-environment connections and physical behaviour, several engineering tools [41], [42] have been created. The heavy reliance on input parameters and subject-matter expertise is a major problem, albeit these empirical techniques are simple to design and practice.

B. Statistical or Regression Models

Traditional methods require prior knowledge, but statistical methods do not. They are focused on making projections about future consumption using the collection of influencing factors. For time series prediction, a variety of statistical techniques have been used, including linear as well as multi-linear regression, auto-regressive (AR), auto-regressive moving average (ARMA), and auto-regressive integrated moving average (ARIMA) models.

The ARIMA model based on multiplicative seasonality and the prediction method employed by the central power authority of India were compared by Rallapalli et al. [43] in their study. The experimental findings demonstrated the multiplicative seasonal ARIMA model's increased accuracy. The inverse error-in-variables-based least-square linear regression methodology to predict electrical load was given by the authors in [44].

Amber et al. used genetic programming and multivariate linear regression to predict daily electricity use [45]. Five significant independent factors were combined using genetic programming to make predictions. Most of these techniques relied on general linear regression. Simple model construction and the removal of extraneous variables improve the predictability of performance. However, multi-collinearity issues might result from the correlation between independent variables utilized for prediction. Another drawback of utilizing linear regression is that it can be challenging to find explanatory factors. Chen et al. [46] created a method based on multi-linear regression to assess how behaviour variables affect patterns of energy usage. Oliveira et al. [47] created effective techniques to enhance univariate forecasting for medium to long-term electric energy usage by combining decomposition and Bootstrap aggregating (Bagging).

Wu et al. [48] suggested using the Adaptive Rate of Change (ARC) technique to calculate the highest electricity usage over a small period which could identify the patterns of peak usage for industrial as well as commercial clients. Using a random forest regression technique, Bogomolov et al. tried to forecast the energy use for the coming week by employing human dynamics analysis [49]. Additionally, Fumo et al. [50] evaluated the significance of several linear and quadratic regression models, as well as affecting factors, in the job of energy usage forecasting. The investigational outcomes of these strategies showed the importance of socio-economic and behavioural factors on the variability in energy usage. Statistical models produce more accurate and encouraging forecast outcomes than physical models do, but the non-linear complications and volatility of the time series data are beyond their handling capacity.

C. AI-Based Models

For predicting electricity consumption, AI-based techniques like SVR, decision trees, and ANN have become widely used during the past few years. When dealing with the non-linearity of the time series data, these strategies have been proven to be quite effective. The uses of AI-based techniques in the arena of energy usage forecasting are discussed in this section. SVM, ANN, and ARIMA were some of the usage prediction approaches that were compared and analyzed by Fu et al. [51]. Later, Hamzacebi et al. [52] published a method for predicting Turkey's monthly electricity consumption while taking seasonal trend impacts into account. Different deep learning-based models have been created and put into use in the classification and prediction of time series throughout the last few years.

Deep belief networks with back-prorogation were used by Dedinec et al. [53] for hourly load forecasting. Then, to estimate electrical energy usage, Krishnan et al. [54] developed an innovative approach based on neural network optimization. The Neural Network Based Particle Swarm Optimization, also known as NNPSO, and the Neural Network Based Genetic Algorithm, also known as NNGA, were used by the authors to optimize the weight of the neural network and boost performance. A unique methodology based on deep learning for day-ahead energy usage prediction was introduced by Tong et al. [5], which incorporated SVR with a stacked auto-encoder (for extracting features). To improve the simplification capacity of a prediction model based on deep neural networks, Chen et al. [55] suggested a multi-step method using Monte Carlo dropout. Recurrent Neural Network (RNN) models are identified as the supreme viable deep learning model for capturing non-linear patterns in energy load profiles.

Recurrent network models have been fruitfully applied to forecast energy consumption in numerous research studies [56], [57]. A neural network based on recurrent inception convolution was proposed by Kim et al. [58] to forecast the consumption of electrical energy with an interval of 30 minutes.

To aid in regulating the forecast time and the latent state vector values derived from the surrounding time steps, they coupled RNN with one dimensional convolution inception unit. A variation of the LSTM architecture, known as the multi-input and multi-output LSTM model, was put out by Bedi et al. [59] to forecast the demand for power during a particular period of the day. However, when complex variable correlations or data volumes grow, the current machine learning techniques suffer from significant overfitting. Long-term consumption is difficult to forecast when overfitting takes place.

D. Hybrid Models

For increased prediction accuracy, hybrid consumption prediction models intelligently blend traditional approaches with AI-based solutions. To estimate the power demand, Hong et al. [60] combined SVR with evolutionary methods (genetic algorithms). For better prediction, authors in [61] combined an enhanced SVR with K-nearest neighbour. Additionally, it has been demonstrated in recent years that combining mode decomposition approaches with AI-based methodologies enhances the effectiveness of power load forecasting models. Next, Moon et al. [62] recommended a hybrid model for predicting total daily electric energy use over a week.

They used the decision tree technique to classify data on electric energy use based on pattern similarity. To choose prototypes with a superior forecasting performance in time series with a comparable structure, they next employed Random Forest and Multilayer Perceptron.

They demonstrated that their hybrid design outperformed former machine learning methods in terms of prediction accuracy. To anticipate day-ahead power load, some studies have integrated wavelet transform [33], [34] with AI-based techniques such as ANN, ANFIS, and SVM. Electricity demand forecasting has also been widely modified using empirical mode decomposition (EMD). In contrast to other wavelet-based and single forecast models, the addition of EMD with deep neural network approaches has demonstrated enhanced performance [63], [64].

The shortcomings of conventional and AI-based prediction models are overcome by hybrid consumption estimation models, and they have the prospective to improve the efficiency of current forecast models [65], [66]. Nonetheless, there are numerous related problems that must be solved, including high computational complexity, choosing the best lag value, selecting the best mode component for reducing noise, and more.

Meanwhile, for the predictive model to be used in real-world applications, findings that examine and elucidate the forecast outcomes are crucial. To explain the outcomes of forecasting, we offer a model in this work that envisions a state by outlining the state centered on the date information and existing usage pattern. Our model, like all other forecasting models, uses the current energy consumption as an input to forecast future energy use.

III. METHODOLOGY AND RELATED CONCEPTS

A. Fast Fourier Transform (FFT)

According to the Fourier transform's official definition, it is a technique that enables the breakdown of functions grounded on space or time into functions grounded on frequency. It can be applied to smooth signals and interpolate functions. For instance, a two-dimensional Fourier transform can be used in handling pixelated images to quickly eradicate high spatial frequency edges of pixels. It is an effective technique for separating many patterns of seasonality from a sole time series variable. For instance, for hourly temperature data, the presence of day/night variations and summer/winter variations can be detected by the Fourier transform and it will articulate the presence of those two frequencies (seasonalities) in the dataset. Discrete Fourier Transform (DFT) is the type of Fourier transform that is required for handling time series data. The reason for naming it as discrete is that the input data is recorded at discrete intervals similar to a non-continuous function like time series data. The Fast Fourier Transform is the quicker variation of DFT. It produces the same outcome as the discrete Fourier, but it has undergone algorithmic optimization. We often need to probe the underlying mechanism that engenders a specific time series. In order to do that, we might want to examine the signal after removing the noise from the data. We can accomplish this goal with the help of the Fourier transformation. We can remove the frequencies that contaminate the data by converting our time series from the time domain to the frequency domain. Then, the inverse Fourier transform can be used to obtain a filtered version of our time series. In Fig. 1, it can be noticed how the FFT filters the noise for various levels of threshold for the coefficients. We remove more frequencies as the threshold value increases.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

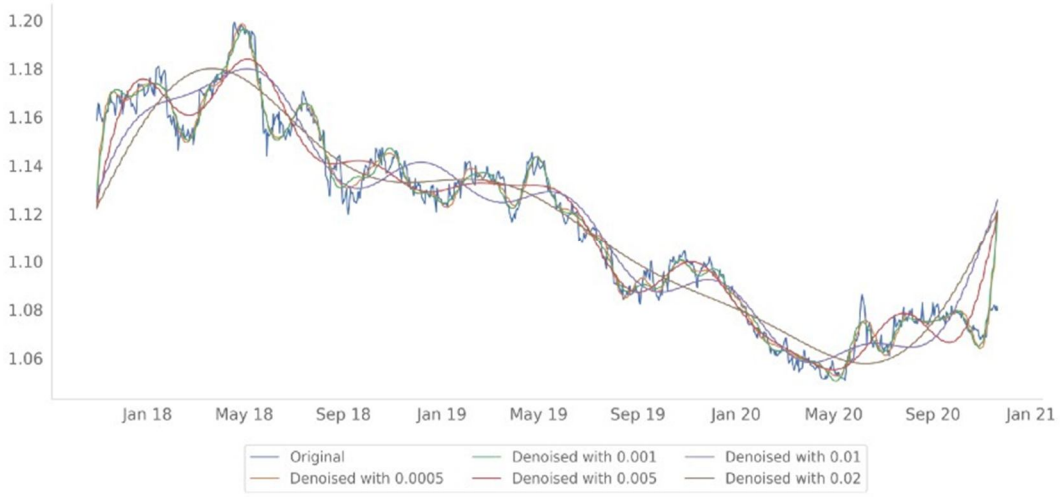


Fig. 1 Fourier transforms applied to data for different coefficient threshold values

B. Local Outlier Factor (LOF)

A data point which deviates significantly from the majority of the data points is an outlier. When handled carelessly, outliers can have a significant impact on our results, which makes them generally undesirable. If our data contains sufficiently large outliers, the Pearson correlation calculation, for instance, may yield a drastically different conclusion. To enhance data quality and provide a cleaner dataset that can be used for further processing, outliers should be removed or corrected. Since you cannot utilize future data to filter previous outliers, outlier analysis and filtering in time series require a more sophisticated method than in regular data. The LOF algorithm calculates the local density deviance of a particular data point with regard to its neighbours. It is an unsupervised technique for anomaly identification. The specimens that have a considerably lower density in contrast to their neighbours are regarded as outliers. The number of neighbours taken into account by the algorithm is typically set lower than the maximum amount of close-by samples that could possibly be local outliers and higher than the minimum amount of samples that a cluster should encompass for other samples to be regarded as local outliers corresponding to this cluster.

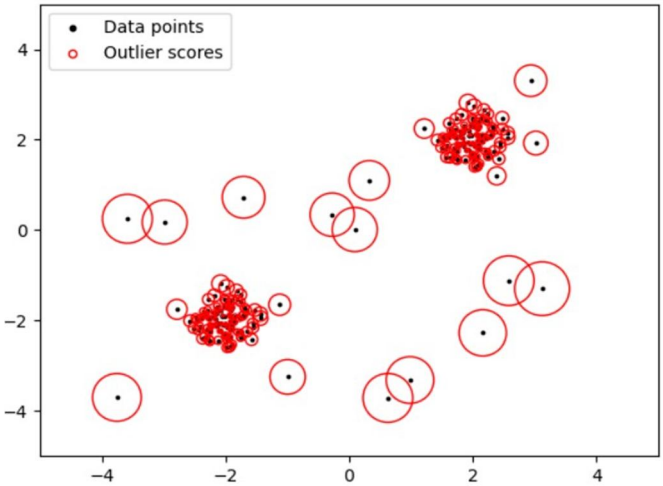


Fig. 2 Detection of outliers using LOF algorithm

C. Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) are a type of artificial neural network (ANN) utilized commonly in the arena of deep learning to construe visual data. They are even called Shift or Space Invariant Artificial Neural Networks (SIANN) due to the shared-weight convolution filter or kernels that slide alongside input features and provide feature maps also known as translation-equivariant responses. Most CNNs do not translate invariantly because of the downsampling operation they perform on the input which is in contrast to popular belief.

CNNs are regularised variants of multilayer perceptrons. Typically, multilayer perceptrons are referred to as fully connected networks, implying that each neuron in one layer is linked to every neuron in the following layer. Also, these networks are susceptible to overfitting data because of their full connectedness. Regularization or overfitting avoidance methods commonly involve penalizing training parameters like weight decay or cutting connectivity, for instance, dropout or skipped connections. By exploiting the hierarchical configuration in the data and amassing patterns of growing complexity using simpler and smaller patterns imprinted in their filters, convolutional neural networks adopt a novel approach for regularization. CNNs are hence present at the bottom end of the complexity and connectivity scale. Image processing is where convolutional neural networks got their start. They are not restricted to merely managing images, though. They have wide applications in the arena of time series, natural language processing, recommendation systems, and brain-computer interfaces. Compared to other techniques, CNNs for time series classification has a number of significant advantages. They can extract incredibly insightful and deep features that are time-independent and highly resistant to noise.

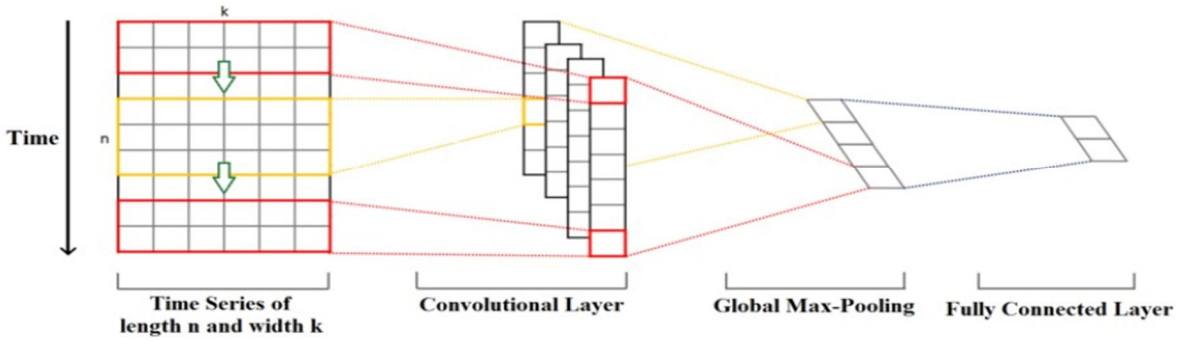


Fig. 3 1-D Convolution for Time Series

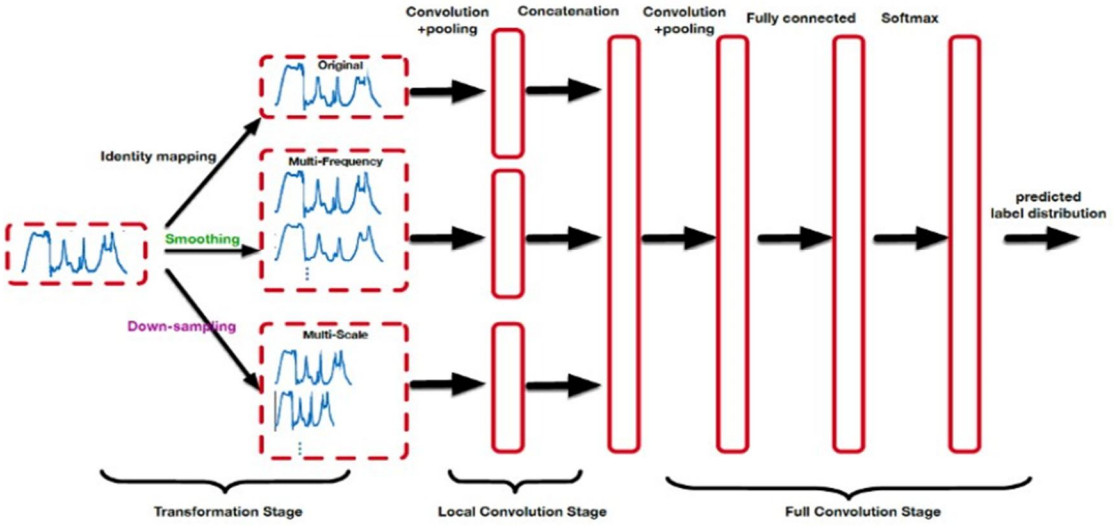


Fig. 4 Multi-Scale Convolutional Neural Network Architecture

D. Long Short-Term Memory Network (LSTM)

The output from some nodes can impact subsequent input to the same nodes in a family of ANNs known as recurrent neural networks (RNN), where relations amongst nodes can form a loop. It can exhibit temporal dynamic behaviour as a consequence. RNNs are a derivative of feed-forward neural networks and they can process input sequences of dissimilar lengths by utilizing their internal memory (state). However, they can only manage short-term data dependencies due to the vanishing gradient issue. With the creation of an RNN variation called the Long Short-Term Memory Network model, the dilemma was solved.

Memory cells and gates—input, output, and forget—make up the architecture of LSTM. Information that is no longer necessary or extraneous gets removed by the forget gate.

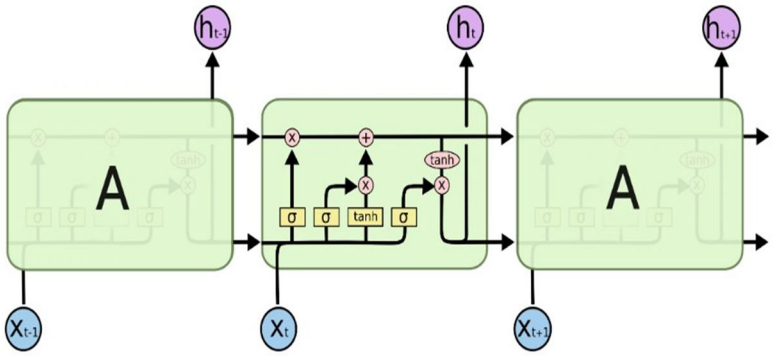


Fig. 5 The repeating module in an LSTM contains four interrelating layers.

All RNNs have a recurring neural network module and a condensed chain-like structure. LSTM also has a comparable design, but the internal iterating module differs. In contrast to a single-level depth neural network, LSTM has four different sorts of interconnected units. We can see that the three gates have a unique network structure in Fig. 5. At each checkpoint, the LSTM's gates play a critical function in allowing information to be selectively influenced. This is accomplished by turning on the activation function, such as sigmoid, in a fully connected neural network with a configuration that yields a value between 1 and 0. When the sigmoid output is 1 and information is transited, the gate opens; when it is 0 and no information is passed through, the gate closes.

E. Auto-Encoder Network

Auto-encoders are a particular type of neural network where the input and output are similar. They reduce the input's dimension before using this representation to recreate the output. The latent-space representation, also known as the code, is an accurate "summary" or "compression" of the input. An auto-encoder network consists of 3 parts namely encoder, code, and decoder. The input is compacted by the encoder, which generates a code. The decoder then reconstructs the input solely using the code. An encoding technique, a decoding method, and a loss function to compare the output with the goal are essential to construct an auto-encoder. The primary function of auto-encoders is dimensionality reduction (or compression). Auto-encoders can only effectively compress data that is identical to that on which they have been trained. They differ from a typical data compression technique like gzip because they learn features relevant to the provided training data. The auto-encoder will produce an approximate but degraded representation of the input rather than an exact match. Since auto-encoders don't necessitate explicit labels for training, they are considered an unsupervised learning method. However, since they generate their labels from the data they are trained on, they can be regarded as self-supervised.

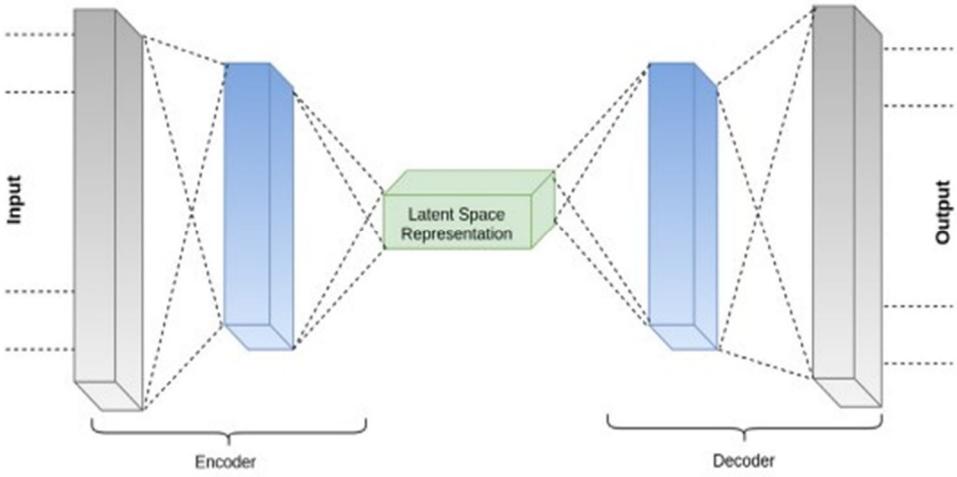


Fig. 6 Architecture of auto-encoder network

F. Hyperband Optimization

For optimizing hyperparameters, there is a complex algorithm called hyperband that is also relatively recent. It uses random sampling and tries to gain an advantage by making the greatest use of the time spent optimizing. The problem of optimizing hyperparameters was described by the method's developers as an infinite-armed bandit, non-stochastic, and pure-exploration problem. By choosing a resource (for example features, data samples, or iterations), one can use Hyperband to distribute it to randomly selected configurations. After training the model with each configuration, one stops training configurations that perform poorly and devotes more resources to configurations that perform well. Successive halving is used in Hyperband. Allocating a budget to a collection of hyperparameter configurations is how successive halving operates. This is done consistently, and after this budget is used up, performance-based criteria are used to discard half of the configurations. The top 50% are kept and given new funding for training, and the bottom 50% are then removed. The procedure is repeated until just one configuration remains.

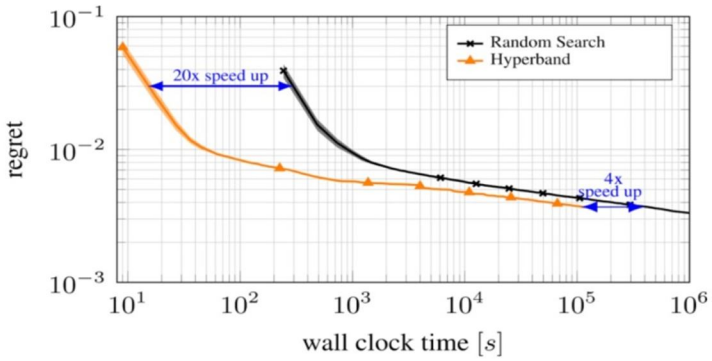


Fig. 7 Comparing performance of random search and hyperband

G. Proposed method

In our proposed framework, the input dataset undergoes a pre-processing step where the data is cleaned using a fast Fourier transform and local outlier factor algorithm. A function that converts a series of prior observations into an output observation will be learned by the model. As a result, the series of observations need to be turned into numerous examples that the LSTM can study. The data sequence can be separated into several input/output patterns termed samples, for instance, where four timesteps are used as input and two timesteps are used as output for the multi-step forecasting that is being learned. So, a mapping is created between multi-step input and multi-step output sequences and fed to the hybrid auto-encoder network consisting of CNN acting as the encoder part and LSTM as the decoder. The network consists of convolutional layers, dropout layers, a repeat vector, LSTM layers, and time-distributed dense layers. CNN will be able to abstract spatial features and feed them to LSTM which can acquire temporal reliance from one sequence to another and model these features for final prediction. Subject to the features of the learning data, changing hyperparameters might have an impact on learning speed and performance. Hence, they are tuned using the hyperband optimization method. Multi-step predictions are made in an auto-regressive manner. Lastly, the time resolution is altered to witness if any further enhancement is possible for the proposed model. Various scale-dependent and percentage-error metrics were used in the experiments to compare the proposed model to the contemporary models. The experimental findings support the claim that the proposed method surpasses the state-of-the-art methods for forecasting energy usage.

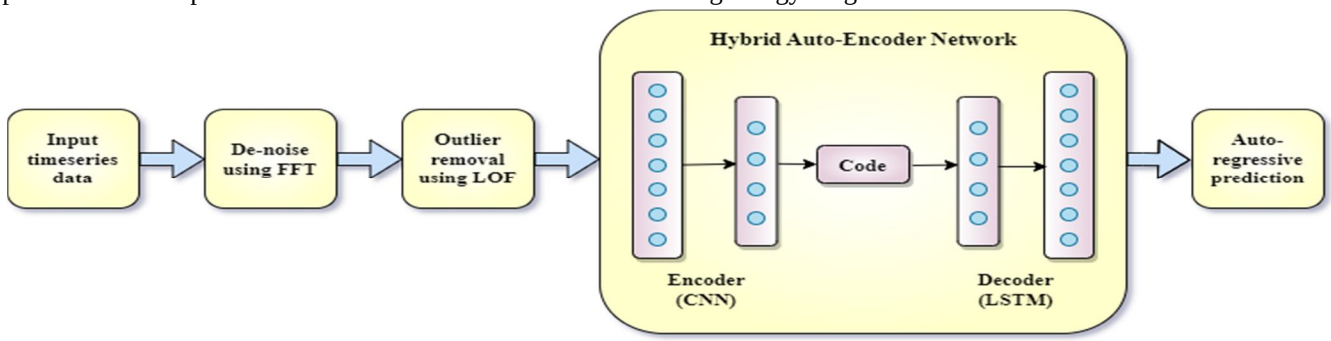


Fig. 8 The proposed architecture for forecasting energy consumption

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

IV. EXPERIMENTS AND RESULTS

A. Dataset

In the US, PJM Interconnection LLC (PJM) is a regional transmission organization (RTO). It is a component of the Eastern Interconnection grid and runs a transmission system for electricity that supplies all or portions of West Virginia, Virginia, Tennessee, Kentucky, Pennsylvania, North Carolina, New Jersey, Ohio, Michigan, Maryland, Indiana, Illinois, Delaware, and the District of Columbia. The data is available on PJM's website as well as on Kaggle [67]. The dataset contains hourly energy consumption data from October 2004 to July 2018 and the values are in megawatts (MW). Fig. 9 shows the time series plot of the dataset at an hourly granularity level.

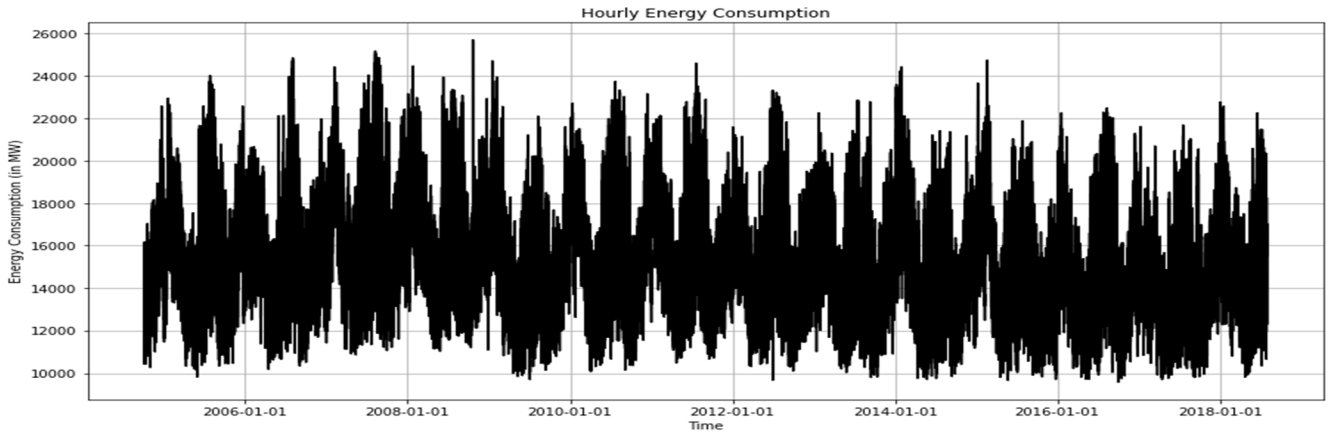
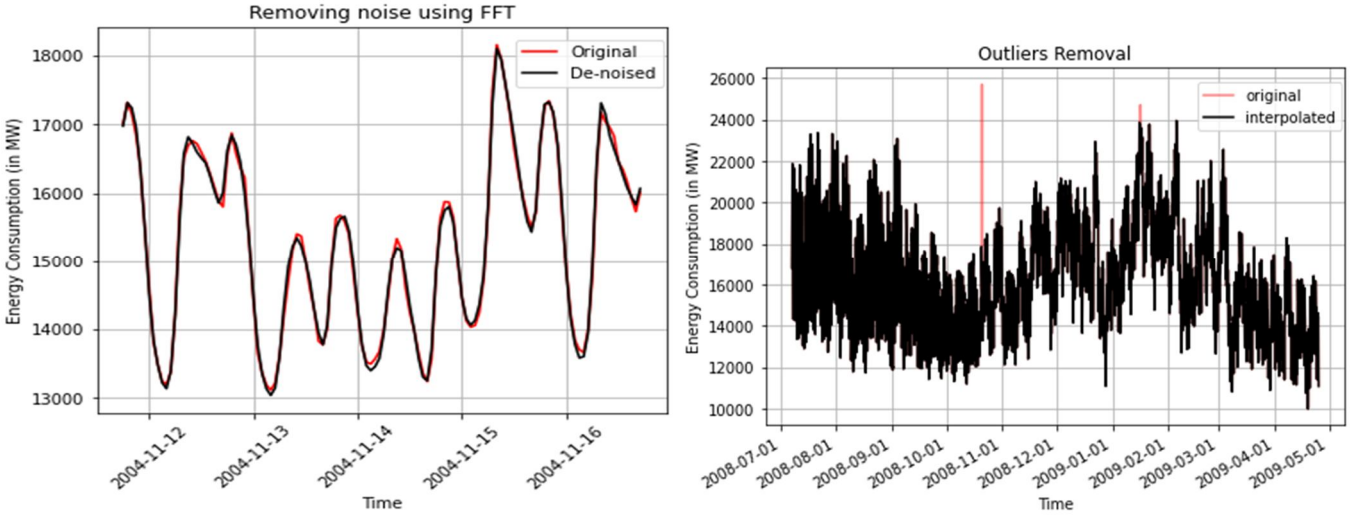


Fig. 9 Time series plot of hourly energy consumption data

B. Model Training

The dataset was fragmented into train and test sets for training and validation respectively. As a part of the pre-processing step, the input dataset underwent fast Fourier transformation to filter out the noise present. Then, the outliers were removed using the local outlier factor algorithm, in which the optimal value for the number of neighbours was chosen after iterating over a range of values. Fig. 10(a) and Fig. 10(b) show the results of the mentioned operations.



(a) De-noising the input dataset (b) Removal of outliers from the data

Fig. 10 Pre-processing operations

The proposed system architecture was implemented using the Tensorflow Keras library. Differently scaled variables might not contribute equivalently to the model learning as well as model fitting functions and could end up creating a bias in the results. Thus, to deal with this impending problem we have used Min-Max scaling, which is a feature-wise normalization process, before model fitting.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

The following model hyperparameters were tuned using the hyperband optimization algorithm:

1) Learning rate for the Adam optimizer
2) Percentage of dropouts after each hidden layer
3) Amount of units in each hidden layer
4) Amount of hidden layers

A learning curve is a graphical representation that displays how well a model learns with experience or time. For algorithms that gain knowledge incrementally from a training dataset, learning curves are an often utilized diagnostic instrument in the field of machine learning. After every update in training, the model can be tested on the training as well as a hold-out validation dataset, and graphs of the calculated performance can be made to showcase the learning curves. The learning curve for our proposed model is shown in Fig. 11, which demonstrates that the model fits the data well. A validation and training loss that diminution to a point of stability with a small difference amongst the two final loss values implies a good fit.

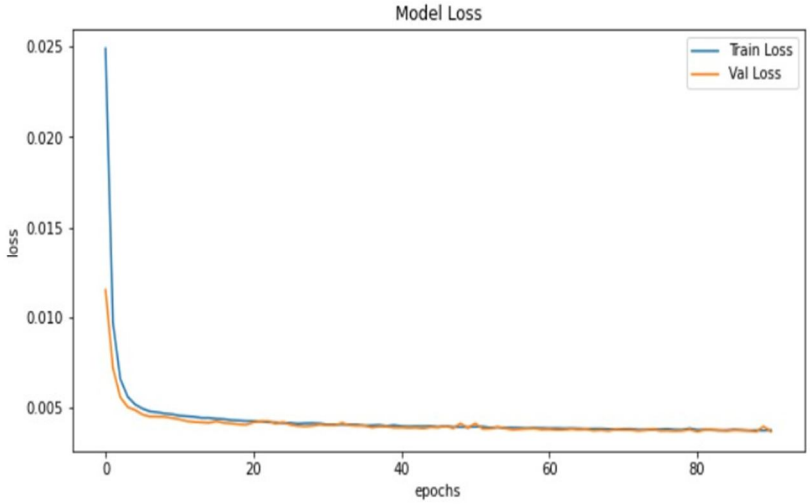


Fig. 11 Training and test loss values for each epoch

C. Results

Experimentations were performed to compare usual performance metrics for energy consumption time series forecasting which include scale-dependent metrics like MAE, MSE & RMSE, and percentage error metrics like MAPE & SMAPE.

The first metric, Mean Absolute Error (MAE), ignores the directions of the prediction errors and considers only their average size. The mean of the absolute deviations between actual and forecast values for each instance in the validation set is what determines the value. This measurement assigns equal weight to all individual variances. The equation that determines MAE is as follows, where $\hat{y}$ and y represent the predicted and actual values respectively:

$$\text{MAE} = \frac{1}{n} * \sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (1)$$

Meanwhile, the mean of the squares of the errors is determined by Mean Square Error (MSE). It is, in other words, the mean squared difference between the predicted and actual values. It is calculated using the equation:

$$\text{MSE} = \frac{1}{n} * \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (2)$$

Additionally, Root Mean Square Error (RMSE) estimates the standard deviation of prediction errors. To understand, let's assume residuals as a measure of how distant the data points are from the regression line. As a result, RMSE is a measurement of how dispersed these residuals are. The following formula is used to calculate this measure, which is frequently used in regression analysis, and forecasting to validate models:

$$\text{RMSE} = \sqrt{\frac{1}{n} * \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \quad (3)$$

A forecasting technique like time series prediction has a metric called Mean Absolute Percentage Error (MAPE) that measures how accurate predictions are made. Due to the asymmetry of MAPE, negative errors (occurring when forecasts exceed actuals) are penalized more severely than positive errors. The following equation gives the accuracy expressed as a percentage for this metric:

$$\text{MAPE} = \frac{1}{n} * \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} * 100 \right| \quad (4)$$

The last metric used is Symmetric Mean Absolute Percentage Error (SMAPE). It is intended to eliminate the MAPE's inherent asymmetry, which is the limitlessness of forecasts that are higher than actuals. It is given by the following equation:

$$\text{SMAPE} = \frac{1}{n} * \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} \quad (5)$$

To exhibit the efficacy of the proposed model, this section compares the performance metrics calculated by Equations (1)-(5) for CNN, LSTM, LSTM Auto Encoder, and the proposed model using hourly and daily datasets obtained from the input dataset. Table 1 presents the outcomes of experimental approaches for the hourly dataset. The proposed model touches the best values of MAE, MSE, RMSE, MAPE and SMAPE at 657.04, 805291.68, 897.38, 4.09, and 4.16 respectively while the second approach, LSTM auto-encoder, obtains 781.25, 970800.79, 985.29, 5.41, and 5.21 respectively. Meanwhile, LSTM and CNN achieve satisfactory results for this dataset. Fig. 12 shows the prediction results of various models for the hourly dataset.

TABLE 1
Performance metrics for different models for hourly dataset

MODEL	MAE	MSE	RMSE	MAPE	SMAPE
CNN	1183.58	2207137.81	1485.64	7.66	7.79
LSTM	1087.558	1776732.77	1332.94	7.39	7.14
LSTM-AE	781.258	970800.79	985.29	5.41	5.21
PROPOSED	657.04	805291.68	897.38	4.09	4.16

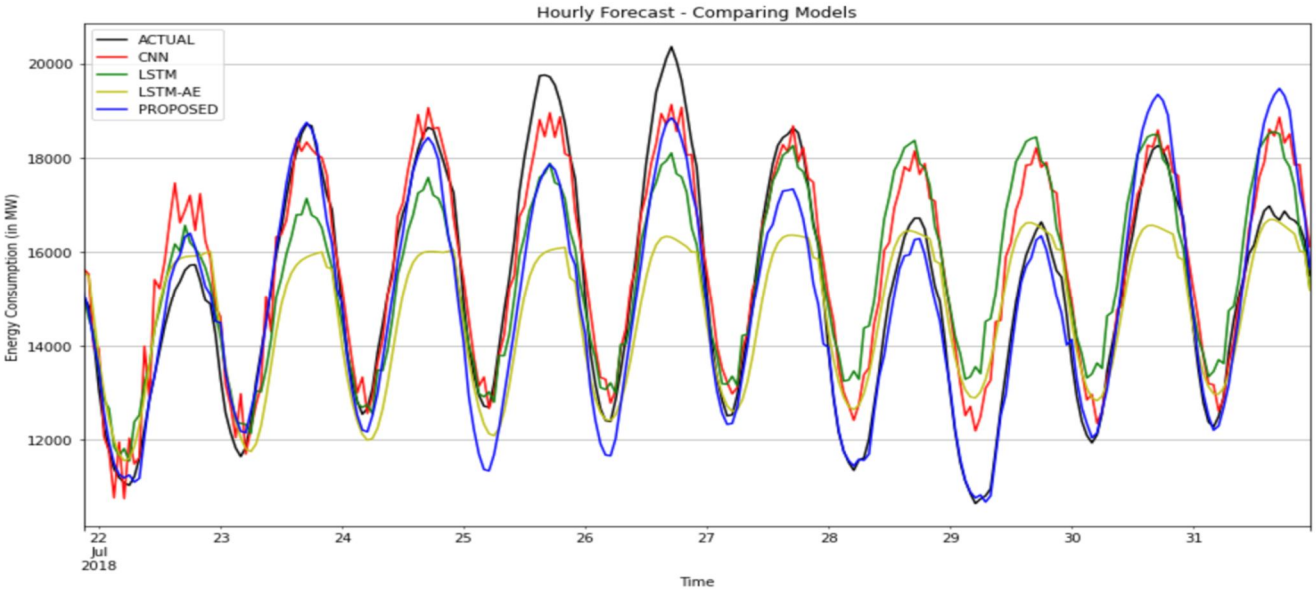


Fig. 12 Comparing the predictive results for different models for hourly dataset

Next, the performance outcomes for the daily dataset are presented in Table 2. The trends are quite similar to the hourly dataset. Our proposed model has the lowest error values. Fig. 13 shows the contrast of predictive results of different models for the daily dataset. When comparing our approach's predictive performance to that of others, there is a significant difference in the outcomes for both datasets. As a result, the proposed model performs better than the alternative approaches.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

TABLE 2
Performance metrics for different models for daily dataset

MODEL	MAE	MSE	RMSE	MAPE	SMAPE
CNN	28064.03	1024407233	32006.36	7.5	7.43
LSTM	25352.17	791058332	28125.76	6.98	6.8
LSTM-AE	23156.18	814493991.3	28539.34	6.13	6.24
PROPOSED	13033.69	336139137.1	18334.09	3.53	3.46

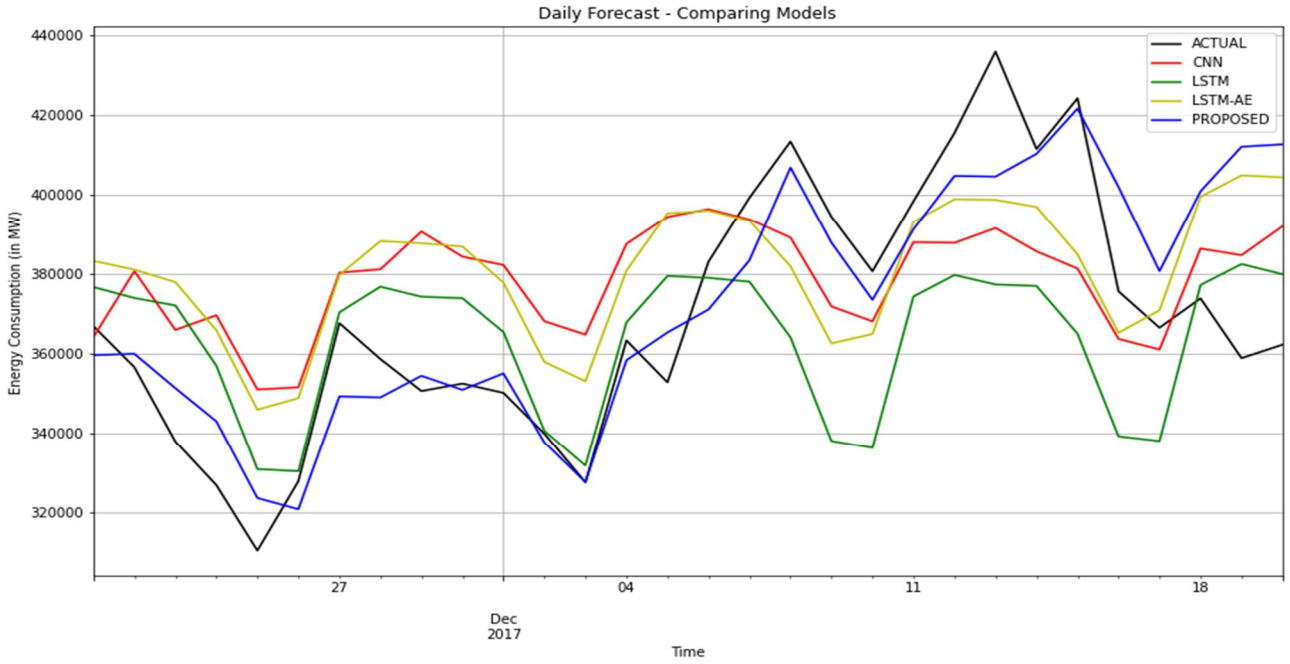


Fig. 13 Comparing the predictive results for different models for daily dataset

V. CONCLUSION

The task of energy forecasting has assumed a vital role in our daily lives because of its significant economic advantages. There are numerous ways to predict energy use. Traditional approaches, however, don't work effectively since they miss the periodicity that is concealed in the data on energy consumption. We propose a hybrid CNN-LSTM Auto-Encoder model with hyperband optimization for efficient and robust forecasting of energy usage. The hybrid auto-encoder network lessens the dimensionality of the complicated input data and extracts important features from the input by creating the compressed version of the information provided as input to it, with little information loss. By identifying factors that have an impact on power consumption, our algorithm can predict electrical energy consumption fast and reliably. It is contrasted with different machine learning practices to prove the utility and dominance of the suggested model. When equated to the current competitive benchmarks, our model estimates complicated electric energy consumption with the best accuracy for both hourly and daily unit resolutions, which also indicates that the model doesn't rely on time resolution. Finally, we have established that removing outliers, lowering the noise, and normalizing the dataset have a substantial impact on the prediction process. To tackle the issue of finding the optimal hyperparameters for the model, we have applied the hyperband optimization algorithm to find the suitable model parameters. For time series of a certain kind, other techniques might be helpful, but only the auto-encoder approach is extensive and robust enough to be used for all categories of time series data.

REFERENCES

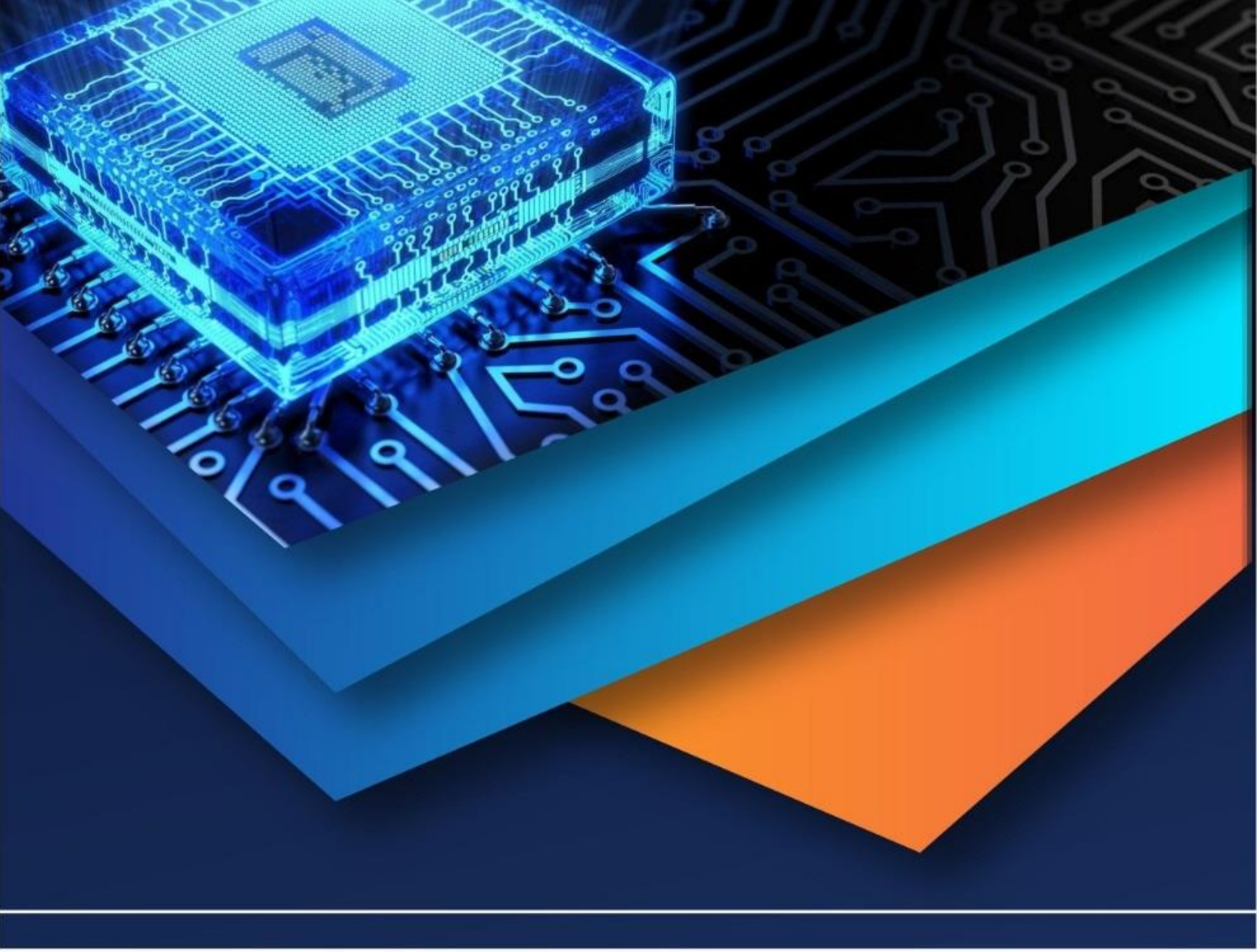
[1] G. Zhao, Z. Liu, Y. He, H. Cao, and Y. Guo, "Energy consumption in machining: Classification, prediction, and reduction strategy," Energy, vol. 133, no. C, pp. 142–157, 2017.
[2] A. Sieminski, "International Energy Outlook," International Energy Agency, pp. 5–30, 2017.
[3] H. Ibrahim, A. Ilinca, and J. Perron, "Energy storage systems—characteristics and comparisons," Renewable and Sustainable Energy Reviews, vol. 12, no. 5, pp. 1221–1250, 2008.

[4] H.-T. Pao, "Forecast of electricity consumption and economic growth in Taiwan by state space modeling," Energy, vol. 34, no. 11, pp. 1779–1791, 2009.
[5] C. Tong, J. Li, C. Lang, F. Kong, J. Niu, and J. J. Rodrigues, "An efficient deep model for day-ahead electricity load forecasting with stacked denoising auto-encoders," J. Parallel Distrib. Comput., vol. 117, no. C, p. 267–273, Jul 2018.
[6] C. Oluklulu, "A research on the photovoltaic modules that are being used actively in utilizing solar energy, sizing of the modules and architectural using means of the modules," Master's thesis, Gazi University, Ankara, Turkey, 2001.
[7] Y. Zuo and E. Kita, "Stock price forecast using Bayesian network," Expert Systems with Applications, vol. 39, no. 8, pp. 6729–6737, 2012.
[8] S. Rasp and S. Lerch, "Neural networks for postprocessing ensemble weather forecasts," Monthly Weather Review, vol. 146, no. 11, pp. 3885–3900, Oct 2018.
[9] F. Nomiyama, J. Asai, T. Murakami, and J. Murata, "A study on global solar radiation forecasting using weather forecast data," 2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 1–4, 2011.
[10] E. Mocanu, H. Nguyen, M. Gibescu, and W. Kling, "Deep learning for estimating building energy consumption," Sustainable Energy, Grids and Networks, vol. 6, p. 91–99, Jun. 2016.
[11] J. Bedi and D. Toshniwal, "Empirical mode decomposition based deep learning for electricity demand forecasting," IEEE Access, vol. PP, Aug 2018.
[12] Q. Zhu, Y. Guo, Q. Zhu, and G. Feng, "Household energy consumption in China: Forecasting with BVAR model up to 2015," in Proceedings of the 2012 Fifth International Joint Conference on Computational Sciences and Optimization, ser. CSO '12. USA: IEEE Computer Society, 2012, p. 654–659.
[13] L. Suganthi and T. R. Jagadeesan, "Energy substitution methodology for optimum demand variation using Delphi technique," International Journal of Energy Research, vol. 16, no. 9, pp. 917–928, 1992.
[14] J. Han, M. Kamber, and J. Pei, "Classification: Advanced methods," in Data Mining (Third Edition), third edition ed., ser. The Morgan Kaufmann Series in Data Management Systems. Boston: Morgan Kaufmann, 2012, pp. 393–442.
[15] F. Rodrigues, C. Cardeira, and J. Calado, "The daily and hourly energy consumption and load forecasting using artificial neural network method: A case study using a set of 93 households in Portugal," Energy Procedia, vol. 62, pp. 220–229, 2014, 6th International Conference on Sustainability in Energy and Buildings, SEB-14.
[16] A. Lahouar and J. Ben Hadj Slama, "Day-ahead load forecast using random forest and expert input selection," Energy Conversion and Management, vol. 103, pp. 1040–1051, 2015.
[17] S. Ben Taieb and R. Hyndman, "A gradient boosting approach to the Kaggle load forecasting competition," International Journal of Forecasting, vol. 30, no. 2, pp. 382–394, 2014.
[18] A. D. Pano-Azucena, E. Tlelo-Cuautle, S. X. D. Tan, B. Ovilla-Martinez, and L. G. De la Fraga, "FPGA-based implementation of a multilayer perceptron suitable for chaotic time series prediction," Technologies, vol. 6, no. 4, 2018.
[19] S. Ismail, A. Shabri, and R. Samsudin, "A hybrid model of self-organizing maps (SOM) and least square support vector machine (LSSVM) for time-series forecasting," Expert Syst. Appl., vol. 38, pp. 10 574–10 578, Aug 2011.
[20] M. Ishikawa and T. Moriyama, "Prediction of time series by a structural learning of neural networks," Fuzzy Sets and Systems, vol. 82, no. 2, pp. 167–176, 1996.
[21] A. R. S. Parmezan, V. M. Souza, and G. E. Batista, "Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model," Information Sciences, vol. 484, pp. 302–337, 2019.
[22] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," 2012.
[23] J. Donahue, L. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," Jun 2015, pp. 2625–2634.
[24] K. Greff, R. K. Srivastava, J. Koutn´ık, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," IEEE Transactions on Neural Networks and Learning Systems, vol. 28, pp. 2222–2232, 2017.
[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, pp. 1735–1780, 1997.
[26] D. Park, S. Kim, Y. An, and J.-Y. Jung, "LiReD: A light-weight real-time fault detection system for edge computing using LSTM recurrent neural networks," Sensors, vol. 18, no. 7, 2018.
[27] X. Ran, Z. Shan, Y. Fang, and C. Lin, "An LSTM-based method with attention mechanism for travel time prediction," Sensors, vol. 19, p. 861, Feb 2019.
[28] J. C.-W. Lin, Y. Shao, Y. Zhou, M. Pirouz, and H.-C. Chen, "A Bi-LSTM mention hypergraph model with encoding schema for mention extraction," Engineering Applications of Artificial Intelligence, vol. 85, pp. 175–181, 2019.
[29] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional CNN-LSTM model," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 225–230.
[30] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 4580–4584.
[31] S. L. Oh, E. Y. Ng, R. S. Tan, and U. R. Acharya, "Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats," Computers in Biology and Medicine, vol. 102, pp. 278–287, 2018.
[32] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to monitor machine health with Convolutional Bi-Directional LSTM networks," Sensors, vol. 17, no. 2, 2017.
[33] R.-A. Hooshmand, H. Amooshahi, and M. Parastegari, "A hybrid intelligent algorithm based short-term load forecasting approach," International Journal of Electrical Power Energy Systems, vol. 45, p. 313–324, Feb 2013.
[34] D. K. Chaturvedi, A. Sinha, and O. P. Malik, "Short term load forecast using fuzzy logic and wavelet transform integrated generalized neural network," International Journal of Electrical Power & Energy Systems, vol. 67, pp. 230–237, 2015.
[35] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Mu¨ller, "Efficient backprop." in Neural Networks: Tricks of the Trade (2nd ed.), 2012, vol. 7700, pp. 9–48.
[36] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," J. Mach. Learn. Res., vol. 13, p. 281–305, Feb 2012.
[37] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," 2012.

©IJRASET: All Rights are Reserved

[38] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," 2016.
[39] W. He, "Load forecasting via deep neural networks," Procedia Computer Science, vol. 122, pp. 308–314, 2017, 5th International Conference on Information Technology and Quantitative Management, ITQM 2017.
[40] N. C. Dalkey and O. Helmer, "An experimental application of the Delphi method to the use of experts," Management Science, vol. 9, pp. 458–467, 1963.
[41] J. White and R. Reichmuth, "Simplified method for predicting building energy consumption using average monthly temperatures," in IECEC 96. Proceedings of the 31st Intersociety Energy Conversion Engineering Conference, vol. 3, 1996, pp. 1834–1839 vol.3.
[42] A. Rice, S. Hay, and D. Ryder-Cook, "A limited-data model of building energy consumption," in Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building, ser. BuildSys '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 67–72.
[43] S. R. Rallapalli and S. Ghosh, "Forecasting monthly peak demand of electricity in India—A critique," Energy Policy, vol. 45, pp. 516–520, 2012.
[44] A. Papalexopoulos and T. Hesterberg, "A regression-based approach to short-term system load forecasting," IEEE Transactions on Power Systems, vol. 5, no. 4, pp. 1535–1547, 1990.
[45] K. Amber, M. Aslam, and S. Hussain, "Electricity consumption forecasting models for administration buildings of the UK higher education sector," Energy and Buildings, vol. 90, pp. 127–136, 2015.
[46] J. Chen, X. Wang, and K. Steemers, "A statistical analysis of a residential energy consumption survey study in Hangzhou, China," Energy and Buildings, vol. 66, pp. 193–202, 2013.
[47] E. M. de Oliveira and F. L. Cyrino Oliveira, "Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods," Energy, vol. 144, no. C, pp. 776–788, 2018.
[48] D. Wu, A. Amini, A. Razban, and J. Chen, "ARC algorithm: A novel approach to forecast and manage daily electrical maximum demand," Energy, 2018.
[49] A. Bogomolov, B. Lepri, R. Larcher, F. Antonelli, F. Pianesi, and A. Pentland, "Energy consumption prediction using people dynamics derived from cellular network data," EPJ Data Science, vol. 5, Mar 2016.
[50] N. Fumo and M. Rafe Biswas, "Regression analysis for prediction of residential energy consumption," Renewable and Sustainable Energy Reviews, vol. 47, pp. 332–343, 2015.
[51] Y. Fu, Z. Li, H. Zhang, and P. Xu, "Using Support Vector Machine to predict next day electricity load of public buildings with sub-metering devices," Procedia Engineering, vol. 121, pp. 1016–1022, Dec 2015.
[52] C. Hamzac¸ebi, H. A. Es, and R. C¸ akmak, "Forecasting of Turkey's monthly electricity demand by seasonal artificial neural network," Neural Comput. Appl., vol. 31, no. 7, p. 2217–2231, Jul 2019.
[53] A. Dedinec, S. Filiposka, A. Dedinec, and L. Kocarev, "Deep belief network based electricity load forecasting: An analysis of Macedonian case," Energy, vol. 115, pp. 1688–1700, 2016, Sustainable Development of Energy, Water and Environment Systems.
[54] M. Krishnan, S. Rathinasamy, and V. Rajagopal, "Neural network based optimization approach for energy demand prediction in smart grid," Neurocomputing, vol. 273, Aug 2017.
[55] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, "Short-term load forecasting with deep residual networks," IEEE Transactions on Smart Grid, May 2018.
[56] H. Shi, M. Xu, and R. Li, "Deep Learning for Household Load Forecasting—A Novel Pooling Deep RNN," IEEE Transactions on Smart Grid, vol. 9, no. 5, pp. 5271–5280, 2018.
[57] A. Rahman, V. Srikumar, and A. D. Smith, "Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks," Applied Energy, vol. 212, pp. 372–385, 2018.
[58] J. Kim, J. Moon, E. Hwang, and P. Kang, "Recurrent inception convolution neural network for multi short-term load forecasting," Energy and Buildings, vol. 194, pp. 328–341, 2019.
[59] J. Bedi and D. Toshniwal, "Deep learning framework to forecast electricity demand," Applied Energy, vol. 238, pp. 1312–1326, 2019.
[60] W.-C. Hong, "Hybrid evolutionary algorithms in a SVR-based electric load forecasting model," International Journal of Electrical Power Energy Systems, vol. 31, pp. 409–417, Sep 2009.
[61] T.-T. Chen and S.-J. Lee, "A weighted LS-SVM based learning system for time series forecasting," Information Sciences, vol. 299, Apr 2015.
[62] J. Moon, Y. Kim, M. Son, and E. Hwang, "Hybrid short-term load forecasting scheme using random forest and multilayer perceptron," Energies, vol. 11, no. 12, 2018.
[63] Y. Ren, P. Suganthan, and S. Narasimalu, "A comparative study of empirical mode decomposition-based short-term wind speed forecasting methods," Sustainable Energy, IEEE Transactions on, vol. 6, pp. 236–244, Jan 2015.
[64] Y. Yu, W. Li, D.-r. Sheng, and J.-h. Chen, "A hybrid short-term load forecasting method based on improved ensemble empirical mode decomposition and back propagation neural network," Journal of Zhejiang University SCIENCE A, vol. 17, pp. 101–114, Feb 2016.
[65] H.-F. Yang and Y.-P. P. Chen, "Hybrid deep learning and empirical mode decomposition model for time series applications," Expert Syst. Appl., vol. 120, pp. 128–138, 2019.
[66] J. Wu, Z. Cui, Y. Chen, D. Kong, and Y.-G. Wang, "A new hybrid model to predict the electrical load in five states of Australia," Energy, vol. 166, 10 2018.
[67] "Hourly energy consumption," Aug 2018. [Online]. Available: https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)