Comparative Analysis of Hybrid Deep Learning Frameworks for Energy Forecasting

Olamide Jogunola*
o.jogunola@mmu.ac.uk
Manchester Metropolitan University, Department of Engineering
Manchester, UK

Abimbola Susan Ajagun
bimbo.ajagun@futminna.edu.ng
Federal University of Technology
Minna, Nigeria

Bamidele Adebisi
b.adebisi@mmu.ac.uk
Manchester Metropolitan University, Department of Engineering
Manchester, UK

Abiodun Musa Aibinu
abiodun.aibinu@futminna.edu.ng
Federal University of Technology
Minna, Nigeria

John Adedapo Ojo
jaojo@lautech.edu.ng
Ladoke Akintola University
Ogbomoso, Nigeria

ABSTRACT

As energy forecasting is paramount to efficient grid planning, this work presents a comparative analysis of different hybrid deep learning frameworks for energy forecasting in applications such as energy consumption and trading. Specifically, we developed hybrid architectures comprising of Convolutional Neural Network (CNN), an Autoencoder (AE), Long Short-Term Memory (LSTM) and Bidirectional LSTM (BLSTM). We use the individual household electric power consumption dataset by University of California, Irvine to evaluate the proposed frameworks. We evaluated and compared the result of these frameworks using several error metrics. The results show an average MSE of ∼ 0.01 across all developed frameworks. In addition, the CNN-LSTM framework performed the least with a 20% and 10% higher RMSE and MAE to other frameworks respectively, while CNN-BiLSTM achieved the least computation time.

KEYWORDS

Hybrid deep learning, convolutional neural network, Bidirectional-long short-term memory, energy consumption prediction, autoencoder.

ACM Reference Format:
Olamide Jogunola, Abimbola Susan Ajagun, Bamidele Adebisi, Abiodun Musa Aibinu, and John Adedapo Ojo. 2021. Comparative Analysis of Hybrid Deep Learning Frameworks for Energy Forecasting. In *The 5th International Conference on Future Networks & Distributed Systems (ICFNDS 2021), December 15–16, 2021, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3508072.3508105

1 INTRODUCTION

Access to affordable and clean energy is not only captured as the seventh $7th$ objective of the Sustainable Development Goals (SDG) but has been linked to all elements of sustainable development [24]. This signifies that the development of any nation is directly related to its energy consumption. Globally, energy demand is on the rise due to the growing population and technological advancement [17]. To ensure energy security and stability, planning the power system against large and sudden variation in generation and demand is highly important [10]. Planning signifies a projection of how the power system should operate over a given period under certain assumptions. Unit commitment, economic dispatch and energy forecasting are employed to ensure stability, reliability, and economic benefits [21].

Energy forecasting is an essential tool in the prediction of energy demand, consumption, and electricity trading. According to the time horizon, energy forecasting can be classified into short-term, medium-term, long-term, and real-time forecasting [25]. Traditional energy forecasting models have been deployed for a variety of applications in different time range and different datasets. The traditional techniques are statistical and machine learning models, for example; Linear Regression, Autoregressive Integrated Moving Average (ARIMA), Support Vector Regression (SVM) and Artificial Neural Network (ANN) are data-driven and implemented on time-series analysis [3]. However, an increase in numerical data introduces complexities and inaccuracy in statistical data which makes these methods inefficient for the modern and growing power network [18], particularly in the surge of distributed energy resources and trading [11, 12].

With advancement in the field of artificial intelligence and machine learning, deep learning models are now used for energy forecasting [1]. Deep learning models can handle large dataset, non-linear relationship, complexities, and offer better performance in terms of computation in a timely manner [16]. The techniques range from Multilayer Perceptron (MLP), Restricted Boltzmann Machines (RBM), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Autoencoders (AE), Deep Reinforcement Learning (DRL), and Generative Adversarial Network (GAN) to hybrid deep learning models. Many studies show that these models have been applied to variety of short-term energy forecasting [15, 20], the recent study focuses on hybrid models for better accuracy [4, 26].

In addition, it is vital to compare different hybrid deep learning frameworks for performance evaluation in energy forecasting applications. For instance in [19], a comparative analysis of the

ICFNDS 2021, December 15–16, 2021, Dubai, United Arab Emirates

ACM ISBN 978-1-4503-8734-7/21/12…$15.00
https://doi.org/10.1145/3508072.3508105

statistical model (ARIMA), machine learning model (Multivariate Linear Regression) and deep learning (LSTM) models for short-term load forecasting was studied. Study [2] focused on comparison between deep learning models including multivariable CNN, LSTM, GRU and hybrids of CNN-LSTM and CNN-GRU models for energy consumption forecast in smart grids. While study [9] compares six common methods, including Persistence, ARIMA, RNN, Long Short-Term Memory (LSTM), Convolutional Neural Network-LSTM (CNN-LSTM), and CNN-Fully-Connected Network (CNN-FCN) for short-term locational marginal price (LMP) forecast.

In line with these studies, a comparative analysis of different models and hybrids for a variety of application in energy forecasting is required for quality and performance evaluation. Therefore, this study seeks to implement the following:

- Develop a variety of deep learning and hybrid deep learning frameworks comprising of CNN, LSTM, BiLSTM and AE for energy forecasting in a variety of applications, including consumption and trading.
- Discuss the architecture of the developed hybrid deep learning frameworks, whilst also evaluating the developed frameworks utilising several error metrics.
- Provide recommendations of frameworks and hyperparameters based on their performances highlighting their usefulness, outcome and trade-offs.

The remaining sections are organised as follows. Section 2 presents the methodology of the energy forecasting frameworks and their algorithms. Section 3 presents data description and architectures of the proposed deep and hybrid learning frameworks, while Section 4 discusses the result. Section 5 concludes the paper with future work.

2 LEARNING ALGORITHMS FOR ENERGY FORECASTING

This section presents the hybrid deep learning algorithms used in this article for energy forecasting. The data cleaning step is first presented followed by the description of the learning algorithms.

2.1 Data pre-processing and Rolling Window

A data pre-processing step to deal with missing values in the collected data is carried out for data smoothing. To improve prediction performance, a moving average filter [22, 23] calculated by a rolling window is employed for data cleaning in this study.

2.2 Deep learning and hybrid deep learning algorithms

Five different deep learning and hybrid learning algorithms comprising CNN, LSTM, BiLSTM and AE for energy forecasting are described below.

2.2.1 CNN. In the context of feature extraction preceding forecasting, CNN is especially skillful at extracting complex features and can store varied irregular trends. Feature extraction is an important pre-processing step to reduce the parameters needed for making predictions, therefore, reducing the network computations while orchestrating prediction accuracy. CNN has several hidden layers for its functioning. This includes a pooling layer, convolutional layer, and an activation function. The input data is fed to the convolutional layer that convert it into a features map. The features map is sampled by the pooling layer to further reduce its dimension.

Given the input vectors $x_i^m = \{x_1, x_2, \cdots, x_n\}$, where x^m represents the varied input vectors that could affect the predicted output, including energy consumption data, weather data, week index, energy price, consumer's behaviour, etc. of $m \in M$, and n is the number of normalised half-hourly unit per window of observation. Utilising the CNN framework with the input vector ${x_i}^m$, the resulting output from the first convolutional layer is expressed in (1).

$$y_{ij}^m = \sigma(b_j^m + \sum_{m=1}^{M} w_{m,j}^1 x_{i+m-1,j}^0) \tag{1}$$

where y_{ij}^m resulted from the output vector x_{ij}^m of the previous layer. b_j^m is the bias for the j^{th} feature map, m is the index value of the filter, w is the weight of the kernel and σ is the activation function for the CNN. Similarly, (2) is the output vector for the k^{th} convolutional layer of the CNN framework.

$$y_{ij}^{m(k)} = \sigma(b_j^{m(k)} + \sum_{m=1}^{M} w_{m,j}^{m(k)} x_{i+m-1,j}^0) \tag{2}$$

Next, the output of the convolutional layer is fed to the input of the pooling layer to further down-samples the activation from feature maps. This process reduces the number of parameters and network computation costs. The max-pooling layer operation is represented by (3).

$$P_{ij}^{m(k)} = \max_{r \in R} y_{i \times T + rj}^{k-1} \tag{3}$$

where y represents the pooling size and T is the stride deciding the length of the input data. If the CNN is being combined in a hybrid model with other architectures, the output from the output $P_{ij}^{m(k)}$ will be fed as input to the next architecture.

2.2.2 LSTM. To overcome the vanishing gradient of RNN and promote preservation of long-term dependencies, LSTM is proposed by Authors in [8]. The LSTM framework is capable of learning from temporal dependencies from one sequence of information to another, which have been proved to be able to process sequence data and applied in real world problems. Essentially, LSTM architecture overcomes the RNN vanishing gradient problem by using memory cells and gates: input, forget and output. The input data to be reserved is determined by the input gate, the forget gate determines the discarded data, the processing states are stored by the memory cells, while the output gate delivered the LSTM output. LSTM is expressed as follows:

$$i_t = \sigma(W_i, [h_{t-1}, x_t] + b_i) \tag{4}$$

$$f_t = \sigma(W_f, [h_{t-1}, x_t] + b_f) \tag{5}$$

$$o_t = \sigma(W_o, [h_{t-1}, x_t] + b_0) \tag{6}$$

$$\bar{C}_t = tanh(W_c, [h_{t-1}, x_t] + b_c) \tag{7}$$

$$C_t = f_t \times C_{t-1} + i_t \times \bar{C}_i \tag{8}$$

$$h_t = o + t \times tanh(C_t) \tag{9}$$

where, i_t, f_t, and o_t are the input, forget and output gate respectively, W_i, W_f, W_c, W_o are the weight of the LSTM gates: input, forget, memory cells and the output, respectively. Also, b_i, b_f, b_c, b_o

are the bias of the respective gates. x_t is the current input vector at time t. σ is the activation function, b is the bias and $\bar{C}_t$ is the candidate memory cell, identifying the memory to store in the cell state; C_{t-1} is the old cell state and C_t is the new cell state. h_t is the hidden state of the LSTM cell, updated at every t step.

2.2.3 BiLSTM. While LSTM architecture is an enhanced version of RNN to overcome its vanishing gradient problem by using memory cells and gates, it only considers the previous state of information, thereby losing valuable information from the next state. Thus, a BiLSTM is used to combine the information in the sequence prediction in both forward and backward directions. The input vector x_t is fed to the input of the BiLSTM layer through the gate units. Similarly to LSTM, BiLSTM consists of different gate functions (input, output and forget gate) in backward and forward directions, each gate is activated when the memory cells update their states represented in (4) through (6). Where h_t is the hidden state of the LSTM cell, updated at every t step in both forward and backward directions. The hidden state and cell state determined through the gate operation of the BiLSTM is expressed in (10) and (11) for the cell and hidden state respectively.

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \sigma(W_i, [h_{t-1}, x_t] + b_i) \tag{10}$$

$$h_t = o_t \cdot \sigma(c_t) \tag{11}$$

The output of the BLSTM layer is concatenated for both forward and backward direction expressed as

$$\bar{y} = \sigma(\overleftrightarrow{W_y h_t} + b_y) \tag{12}$$

2.2.4 Encoder-decoder (AE). While CNN extracts important features from the dataset, AE, on the other hand, are specially designed for representation learning. For instance, AE are utilised to understand unsupervised inputs in a feature vector. It consist of an encoder and a decoder to first encode the input sequence before subsequently decoding it using internal representations. The encoder framework could be made up of several RNN units to encode the input sequence into a vector (C). Based on the input vector x_t and previous hidden state h_{t-1}, the current hidden state h_t is calculated as $h_t = f(x_t, h_{t-1})$. Where, f is any RNN function, like LSTM or GRU [5]. The output vector from the encoder unit serves as the input vector to the decoder unit. The decoder unit follows the encoder representations; thus, the hidden layer is expressed as $h_t = f(C, h_{t-1})$ to give an output $y_t = g(h_t)$. Where C is the output vector from the encoder unit and g is an activation function.

3 FRAMEWORK EVALUATION

This section presents the experimental setup, dataset description and evaluation metrics for the proposed deep and hybrid learning frameworks for energy forecasting.

3.1 Dataset description

The UCI dataset [7] is used to evaluate and compare the proposed frameworks. To avoid repetition, a detailed description of the UCI dataset is provided in [13, 14]. The UCI dataset has 8 input variables and 1 output target with a total of 2,075,269 records. There are 25,979 missing values which are handled in the data cleaning steps before proceeding to the deep and hybrid learning frameworks. UCI dataset is for residential buildings. The dataset time resolution is converted to $24hr$ for short-term electricity prediction.

3.2 Experimental setup and Evaluation Metrics

The computation to train and test the developed frameworks is performed on Google Colaboratory [6] using Intel Core i7-CPU, 16 GB RAM and 64-bit operating system.

Table 1: LSTM-autoencoder and its Definition

No	Layer Type	Neurons	Param
1	Input	8	8
4	LSTM	200	167200
7	Repeat vector	200	0
8	LSTM	200	320800
9	TimeDistributed (Dense)	100	20100
10	TimeDistributed (Dense)	1	101

After extensive experimentation and analysis of different parameters, the hyperparameter meeting the optimal performance of the developed frameworks are summarised in Tables 1, 2, 3, 4, and 5.

Table 2: CNN-LSTM-AE and its Definition

No	Layer Type	Neurons	Param
1	Input	8	8
2	Convolution1D	64	1600
3	Convolution1D	64	12352
4	MaxPooling1D	64	0
5	Flatten	320	0
6	Repeat vector	320	0
7	LSTM	200	416800
8	TimeDistributed (Dense)	100	20100
9	TimeDistributed (Dense)	1	101

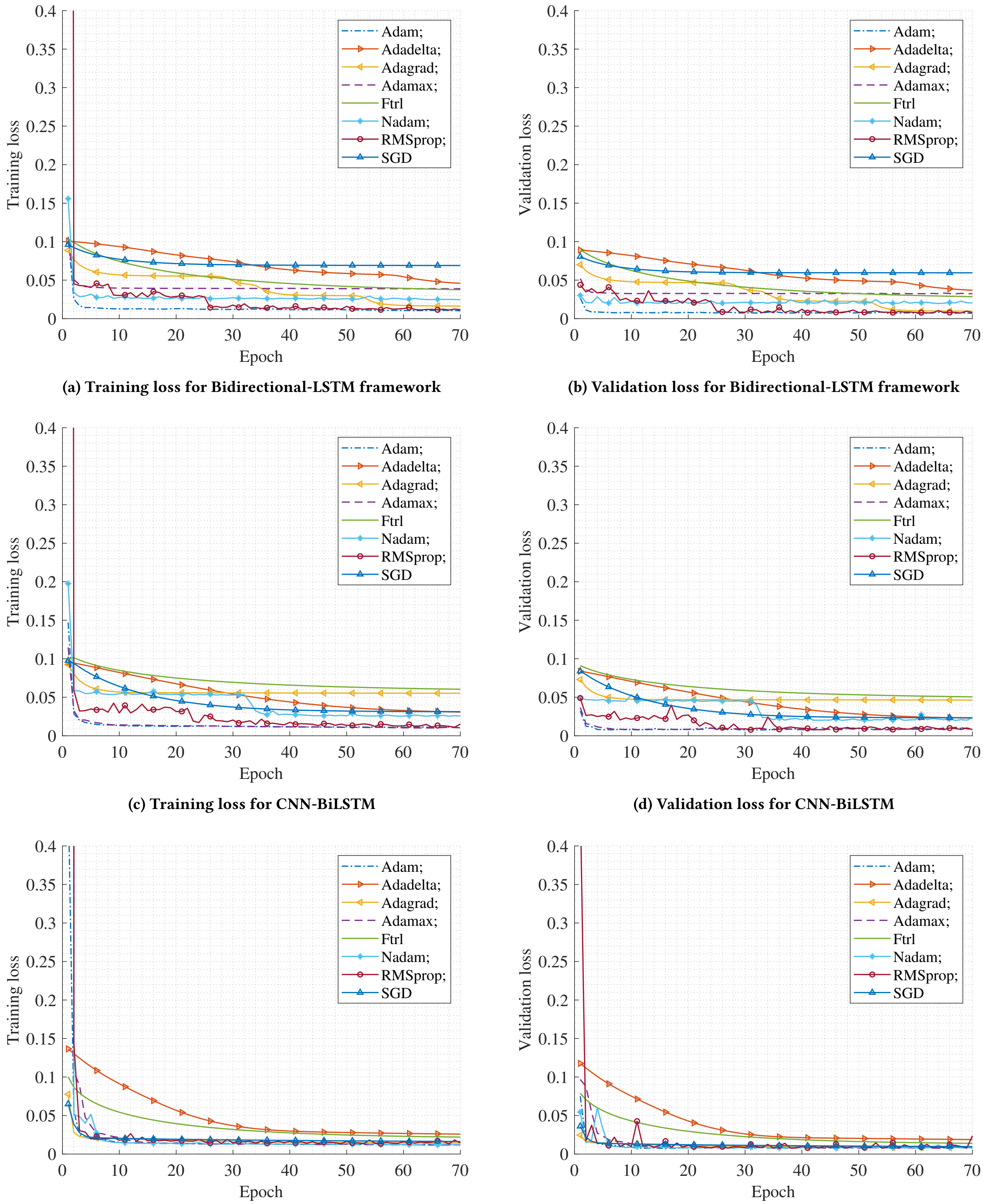
Table 1 presents the framework with LSTM and autoencoder. Table 2 presents the framework with CNN, LSTM and autoencoder. Table 3 presents the framework with BiLSTM, while Table 4 includes a CNN with BiLSTM in its framework. Finally, Table 5 presents the CNN and LSTM framework.

Table 3: Bidirectional LSTM and its Definition

No	Layer Type	Neurons	Param
1	Input	8	8
5	Bidirectional	128	37376
9	Dense	100	12900
10	Dense	7	707

After further extensive hyperparameter tuning, using a variety of optimiser, a learning rate of 0.001, 70 epoch, 160 batch-size, 0.33 validation split and ReLU activation function are selected.

The performance of the developed frameworks are evaluated using several error metrics, including mean squared error (MSE), mean absolute error (MAE), and root mean squared error (RMSE), as well as the computation time. The computation time includes



(a) Training loss for Bidirectional-LSTM framework

(b) Validation loss for Bidirectional-LSTM framework

(c) Training loss for CNN-BiLSTM

(d) Validation loss for CNN-BiLSTM

(e) Training loss for CNN-LSTM-AE

(f) Validation loss for CNN-LSTM-AE

Figure 1: Training and validation loss for different optimisers for the developed hybrid deep learning frameworks

Table 4: CNN-Bidirectional LSTM and its Definition

No	Layer Type	Neurons	Param
1	Input	8	8
2	Convolution1D	64	1600
3	Convolution1D	64	12352
4	MaxPooling1D	64	0
5	Bidirectional	128	66048
9	Dense	100	12900
10	Dense	7	707

Table 5: CNN-LSTM and its Definition

No	Layer Type	Neurons	Param
1	Input	8	8
2	Convolution1D	64	1600
3	Convolution1D	64	12352
4	MaxPooling1D	64	0
5	TimeDistributed (Dense)	64	0
6	LSTM	100	66000
7	Dense	100	10100
8	Dense	7	707

the training and testing time of the frameworks on the dataset. The MSE measures the average of the squares of the difference between the predicted and actual values illustrated in (13).

$$\text{MSE} = \frac{1}{n}\sum_{1}^{n}(y - \bar{y})^2 \tag{13}$$

where $\bar{y}$ is the vector of n predictions produced from the n energy dataset, and y is the observed vector of the predicted energy variables. RMSE expressed in (14) is the standard deviation of predicted errors, i.e., the root mean square of MSE

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{1}^{n}(y - \bar{y})^2} \tag{14}$$

On the other hand, MAE expressed in (15) measures the absolute differences between the predicted and the actual values

$$\text{MAE} = \frac{1}{n}\sum_{1}^{n}|y - \bar{y}| \tag{15}$$

4 RESULTS AND DISCUSSION

In this section, an experiment to select an appropriate optimiser for the developed frameworks is first presented. Then a comparative analysis of the frameworks using the UCI energy consumption dataset against the defined error metrics is presented.

4.1 Experiment on different optimisers

Since an appropriate optimiser is paramount for efficient dimensionality reduction, the developed frameworks are analysed to determine the optimum optimiser. We performed different experiments in selecting an optimal optimiser for the frameworks. The optimisers analysed are Adam, Adadelta, Adagrad, Adamax, Ftrl, Nadam, RMSprop, and SGD. This is illustrated in Fig. 1 (a to f). Fig. 1 shows the training and validation loss for 3 of the developed frameworks. While all the training and validation loss for the optimisers are less than 0.1, Adam optimiser gave the lowest training and validation loss in all the developed frameworks. In addition, it can be observed that all the optimisers except Adadelta and Ftrl have similar result for the framework with AE (Fig. 1 (e and f)) compared to other proposed frameworks.

4.2 Comparative analysis of the frameworks

To compare the developed frameworks using the defined error metrics, we selected Adam optimiser with a learning rate of 0.001, since it performed better than other optimisers from the above analysis. Fig. 2 presents the comparison result using the UCI dataset for daily energy consumption prediction. In terms of MSE, all the frameworks have a similar result of 0.01. For RMSE and MAE, CNN-LSTM framework has the highest value of 0.1 and 0.08, respectively compared to other frameworks. Specifically, the RMSE and MAE values for CNN-LSTM are 20% higher than BiLSTM, CNN-BiLSTM and CNN-LSTM-AE, and 10% higher than LSTM-AE. However, comparing the computation time including training and testing, LSTM-AE is the longest, followed by CNN-LSTM-AE. CNN-BiLSTM achieved the lowest computation time among the developed frameworks.

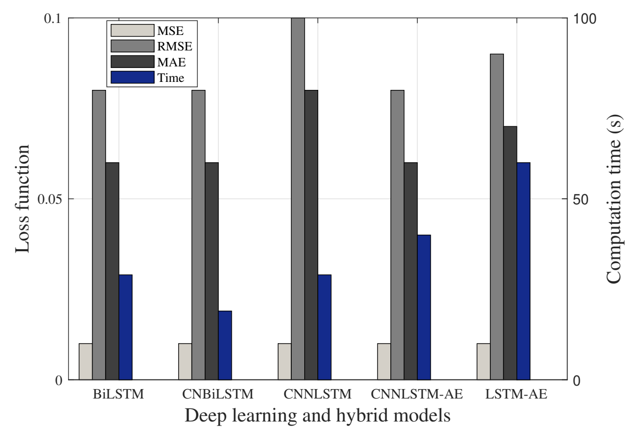


Figure 2: Performance comparison of the developed frameworks.

5 CONCLUSION

This work developed five different hybrid deep learning frameworks utilising architectures comprising CNN, AE, LSTM, and BiLSTM for energy forecasting applications. Utilising household energy consumption data to evaluate the developed frameworks, the results were compared using several error metrics. The results show an average MSE of ~ 0.01 across all developed frameworks. In addition, the CNN-LSTM framework performed the least with a 20% and 10% higher RMSE and MAE to other frameworks respectively, while CNN-BiLSTM achieved the least computation time. The future work will focus on an in-depth study on automatic fine-tuning of hyperparameters to decide on optimal values, instead of by trial and error.

ACKNOWLEDGMENTS

This work was supported by ENERGY-IQ, a UK-Canada Power Forward Smart Grid Demonstrator project funded by The Department for Business, Energy and Industrial Strategy (BEIS) under Grant number:7454460

REFERENCES

[1] Musaed Alhussein, Khursheed Aurangzeb, and Syed Irtaza Haider. 2020. Hybrid CNN-LSTM model for short-term individual household load forecasting. *IEEE Access* 8 (2020), 180544–180557.

[2] E Escobar Avalos, MA Rodríguez Licea, H Rostro González, A Espinoza Calderón, AI Barranco Gutiérrez, and FJ Pérez Pinal. 2020. Comparative Analysis of Multivariable Deep Learning Models for Forecasting in Smart Grids. In *Intl. Autumn Meeting on Power, Electronics and Computing (ROPEC)*, Vol. 4. IEEE, Ixtapa, Guerrero, Mexico, 1–6.

[3] Mengmeng Cai, Manisa Pipattanasomporn, and Saifur Rahman. 2019. Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques. *Applied energy* 236 (2019), 1078–1088.

[4] Zhaojing Cao, Can Wan, Zijun Zhang, Furong Li, and Yonghua Song. 2019. Hybrid ensemble deep learning for deterministic and probabilistic low-voltage load forecasting. *IEEE Trans. Power Syst.* 35, 3 (2019), 1881–1897.

[5] Gopal Chitalia, Manisa Pipattanasomporn, Vishal Garg, and Saifur Rahman. 2020. Robust short-term electrical load forecasting framework for commercial buildings using deep recurrent neural networks. *Applied Energy* 278 (2020), 115410.

[6] Google. 2021. Welcome to Colaboratory. Available at https://colab.research.google.com/, Accessed: 2021-10-16.

[7] G Hebrail and A Berard. 2021. Individual Household Electric Power Consumption Data Set. Available at https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption, Accessed: 2021-10-16.

[8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[9] Ying-Yi Hong and Rolando Pula. 2020. Comparative studies of different methods for short-term locational marginal price forecasting. In *Intl. Conf. Green Tech. and Sust. Dev. (GTSD)*. IEEE, Ho Chi Minh City, Vietnam, 527–532.

[10] Priyam Jain, Aman Gautam, Rahul Shukla, RK Porwal, Debasis De, SR Narasimhan, and KVS Baba. 2020. Planning and Operation of Indian Power System during the Pan India Lights Off Event. In *21st National Power Syst. Conf. (NPSC)*. IEEE, Gandhinagar, India, 1–6.

[11] Olamide Jogunola, Bamidele Adebisi, Augustine Ikpehai, Segun I. Popoola, Guan Gui, Haris Gačanin, and Song Ci. 2021. Consensus Algorithms and Deep Reinforcement Learning in Energy Market: A Review. *IEEE Internet of Things Journal* 8, 6 (2021), 4211–4227. https://doi.org/10.1109/JIOT.2020.3032162

[12] Olamide Jogunola, Yakubu Tsado, Bamidele Adebisi, and Mohammad Hammoudeh. 2021. VirtElect: A Peer-to-Peer Trading Platform for Local Energy Transactions. *IEEE Internet of Things Journal* (2021), 1.

[13] Zulfiqar Ahmad Khan, Tanveer Hussain, Amin Ullah, Seungmin Rho, Miyoung Lee, and Sung Wook Baik. 2020. Towards Efficient Electricity Forecasting in Residential and Commercial Buildings: A Novel Hybrid CNN with a LSTM-AE based Framework. *Sensors* 20, 5 (2020), 1399.

[14] Tae-Young Kim and Sung-Bae Cho. 2019. Predicting residential energy consumption using CNN-LSTM neural networks. *Energy* 182 (2019), 72–81.

[15] Min-Seung Ko, Kwangsuk Lee, Jae-Kyeong Kim, Chang Woo Hong, Zhao Yang Dong, and Kyeon Hur. 2020. Deep Concatenated Residual Network With Bidirectional LSTM for One-Hour-Ahead Wind Power Forecasting. *IEEE Trans. Sust. Energy* 12, 2 (2020), 1321–1335.

[16] Mohamed Massaoudi, Haitham Abu-Rub, Shady S Refaat, Ines Chihi, and Fakhreddine S Oueslati. 2021. Deep learning in smart grid technology: A review of recent advancements and future prospects. *IEEE Access* 9 (2021), 54558–54578.

[17] RC Ney, MR Ferreira, MP Vianna, RB Orling, MA Gama, and LN Canha. 2020. Planning Energy Distribution Systems in an Environment That Accelerates the Use of Distributed Energy Resources. In *PES Trans. & Distr. Conf. and Exhibition-Latin America (T&D LA)*. IEEE, Montevideo, Uruguay, 1–6.

[18] Vladimir Popov, Mykola Fedosenko, Vadim Tkachenko, and Dmytro Yatsenko. 2019. Forecasting consumption of electrical energy using time series comprised of uncertain data. In *6th Int. Conf. Energy Smart Syst. (ESS)*. IEEE, Kyiv, Ukraine, 201–204.

[19] Rajat Sethi and Jan Kleissl. 2020. Comparison of Short-Term Load Forecasting Techniques. In *Conf. Tech. for Sustainability (SusTech)*. IEEE, Santa Ana, CA, USA, 1–6.

[20] Dabeeruddin Syed, Haitham Abu-Rub, Ali Ghrayeb, and Shady S Refaat. 2021. Household-level energy forecasting in smart buildings using a novel hybrid deep learning model. *IEEE Access* 9 (2021), 33498–33511.

[21] Philipp A Trotter, Marcelle C McManus, and Roy Maconachie. 2017. Electricity planning and implementation in sub-Saharan Africa: A systematic review. *Renewable and Sust. Energy Reviews* 74 (2017), 1189–1209.

[22] Fath U Min Ullah, Amin Ullah, Ijaz Ul Haq, Seungmin Rho, and Sung Wook Baik. 2019. Short-term prediction of residential power energy consumption via CNN and multi-layer bi-directional LSTM networks. *IEEE Access* 8 (2019), 123369–123380.

[23] Israr Ullah, Rashid Ahmad, and DoHyeun Kim. 2018. A prediction mechanism of energy consumption in residential buildings using hidden markov model. *Energies* 11, 2 (2018), 358.

[24] United Nations. 2021. Sustainable development goals report 2021. Available at https://unstats.un.org/sdgs/report/2021/, Accessed: 2021-11-1.

[25] Gao Xiuyun, Wang Ying, Gao Yang, Sun Chengzhi, Xiang Wen, and Yue Yimiao. 2018. Short-term load forecasting model of gru network based on deep learning framework. In *2nd Conf. Energy Internet and Energy Syst. Integration (EI2)*. IEEE, Beijing, China, 1–4.

[26] Yue Zhang, Chuan Qin, Anurag K Srivastava, Chenrui Jin, and Ratnesh K Sharma. 2020. Data-driven day-ahead PV estimation using autoencoder-LSTM and persistence model. *IEEE Trans. Ind. Appl.* 56, 6 (2020), 7185–7192.