Addressing Load Forecasting Challenges in Industrial Environments Using Time Series Deep Models

Mohamed Bouzid
Concordia Institute for Information Systems Engineering
Montreal, QC, Canada
mohamed.bouzid@mail.concordia.ca

Manar Amayri
Concordia Institute for Information Systems Engineering
Montreal, QC, Canada
manar.amayri@concordia.ca

Nizar Bouguila
Concordia Institute for Information Systems Engineering
Montreal, QC, Canada
nizar.bouguila@concordia.ca

ABSTRACT

Energy is one of the most important topics in the modern world, as it affects various aspects of human life and the environment. The current transition era and the growing demand for energy require innovative solutions to optimize energy use and reduce its negative impacts. Effective energy management can lead to improved consumption patterns for consumers and a better understanding and control of the demand for producers. However, energy management is a complex and challenging task that involves multiple factors and uncertainties. In this paper, we focus on the problem of electricity consumption prediction and we discuss several deep learning models that involve long short-term memory networks (LSTM), temporal convolutional networks (TCN) and attention mechanisms. We analyze and compare their performance by testing them on the Grenoble University building dataset, a non-aggregated dataset that contains electricity consumption data for different types of rooms over a period of two years. Root mean squared error (RMSE) and R-squared metrics were used to evaluate the tested models. The results show that deep learning models can achieve promising results on this dataset, and that some features and preprocessing methods can improve the performance significantly.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; *Time series analysis*; • **Applied computing** → *Smart buildings*; Energy management.

KEYWORDS

Energy management, Demand, electricity, building, Grenoble dataset, Deep learning, non-aggregated

ACM Reference Format:
Mohamed Bouzid, Manar Amayri, and Nizar Bouguila. 2023. Addressing Load Forecasting Challenges in Industrial Environments Using Time Series Deep Models. In *Proceedings of The 6th International Conference on Computational Intelligence and Intelligent Systems (CIIS 2023) (Conference acronym 'XX)*. ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, November 25-27, 2023, Tokyo, Japan

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Energy is one of the most crucial resources in modern society. Not only does it enable various activities and services that improve the quality of our life but also contributes to economic development. Particularly, the electricity sector went through major changes. The world's electricity consumption is constantly increasing due to various factors such as the development of electric cars, the electrification of industrial processes, the large use of heating systems [28], weather conditions, and economic activity [17]. According to Statista [26], the world's electricity consumption reached approximately 25,300 terawatt-hours in 2021, which is more than tripled since 1980. However, energy poses challenges to the environment and the sustainability of the planet. Therefore, it is essential to find ways to use energy more efficiently and intelligently. For example, predicting consumption can be useful to make buildings more efficient, less electricity consumption, and better manage electricity consumption through energy management systems.

One of the key aspects of smart energy use is to understand and predict electricity consumption patterns. This can help to optimize the energy supply and demand, reduce costs and emissions, and enhance the reliability and security of the power system. But, electricity consumption prediction is a complex and dynamic problem that depends on many factors, such as weather, time, season, user behaviour, future events, occupancy, etc.

In this paper, we focus on the usage of machine learning for the task of electricity consumption prediction. Machine learning is known to be a powerful tool for load forecasting, which will help in building energy-efficient management systems. One of the biggest benefits of machine learning and deep learning models is that they can learn from complex data and they do not need domain knowledge or explicit feature engineering [11]. Their flexibility to predict and plan for high/low load days, and to detect trends and abnormalities in difficult consumption situations could also be useful. Yet, they can face some challenges when dealing with noisy and incomplete datasets, which requires choosing the appropriate models and hyperparameters to ensure a better generalization of the predictions. Hence, we propose to use four deep learning models to tackle this task. The models that we considered are Deep Autoregressive model (DeepAR) [22], Multi-horizon Quantile Recurrent Forecaster (MQ-RNN) [29], Deep Temporal Convolutional Network [4], and Dual-Stage Attention-Based Recurrent Neural Network (DA-RNN) [21], which are state-of-the-art models for time series forecasting. The selection of the problem provides different approaches and architectures. These models can handle probabilistic outputs where some of them are RNN-based while others are CNN-based. Additionally, some of them use point forecasting while others use

multi-horizon forecasting. To evaluate their performance, we chose the Grenoble University building dataset that contains electricity consumption data for different types of rooms over a period of two years. This is a challenging scenario because the consumption data is non-aggregated, meaning that it reflects the individual behaviour of each room's occupants.

The main contributions of this paper are:

- We evaluate and compare several deep learning models on this dataset and show that they can achieve promising results despite the difficulty of the data.
- We provide a comprehensive analysis of the dataset and the models and discuss some possible directions for improving the prediction accuracy and interpretability.

The remainder of the paper is organized as follows: Section 2 provides a brief review of different methods for forecasting and related works on load forecasting. In Section 3, we provide the reader with background knowledge for a better understanding of the topic. Then, in Section 4, we describe the dataset used to evaluate the models in addition to discussing data preprocessing. Next, in Section 5, we will review the usage of different models in this particular dataset in Section 5 and compare their performance in Section 6. Finally, we conclude the study in Section 7.

2 RELATED WORKS

Research interest in energy demand has gained so much attention during the last few years due to the increasing demand for electricity or other renewable energies. The ability to predict future energy demand is known as load forecasting which is a part of time series forecasting. In fact, it is important to know that demand is represented as a form of time series. Time series is a type of data that consists of observations that are ordered chronologically. The goal is to predict future events based on historical information. These predictions help in assessing future risks, managing them, and improving future strategies. Time-series forecasting has been used in a lot of fields, including but not restricted to energy, healthcare, finance, etc. The primary focus of this paper is to assess deep learning models' ability to make multi-horizon load forecasting in buildings on non-aggregated data. Several related research efforts have been made on this topic and we review them in the following. The early days of time-series forecasting consisted of using traditional statistical methods like autoregressive models (e.g., ARIMA) and exponential smoothing [6]. Since time series data usually exhibit complex patterns such as trends, seasonality, and non-linearity, which are challenging to capture by statistical models. In addition, these models rely on one-step-ahead predictions which is not recommended in load forecasting because it does not include any uncertainties. Deep learning models have been also used in literature since they have the potential of overcoming these challenges and handling large-scale data. Convolutional-based models [8] and Recurrent Neural Networks (RNN) [32] have been the most popular models in load forecasting. However, fully convolutional networks did not have the same success they had in computer vision due to their capture of temporal information in sequence data caused by downsampling and upsampling operations. To tackle this problem, Dilated convolution [4], which introduces gaps between kernels to capture long-term dependencies, has been used. On the other hand, RNN suffers from the gradient vanishing and exploding problem. This led to the use of long short-term memory (LSTM) [12], one of the variants of RNNs that introduces memory cell gates to capture long-term dependencies, helping alleviate the vanishing gradient problem. Moreover, one popular approach to handle time series is to use a sequence-to-sequence structure [5], also known as the encoder-decoder structure. In this approach, the input is encoded into a fixed-length representation that will be fed to the decoder RNN to generate an output sequence that can be extended to multi-horizon forecasting.

When it comes to load forecasting, different studies that have been presented in the last few years come to mind. First, [13] and [30] studied numerous statistical and machine learning techniques for energy prediction, such as linear regression, SVM, and more. Moreover, Ensemble methods [3] have been popular for load forecasting, such as Random forest [10] and some other methods [16]. Additionally, [19] provided a literature survey of building energy prediction using artificial neural networks. The authors reviewed and summarized the application of twelve Artificial Neural Networks architectures such as Multi-layer Perceptron (MLP), Wavelet neural network (WNN), RNN, LSTM, CNN, and more. When it comes to multi-step forecasting, [27] studied the performance of several multi-step techniques on an MLP. Also, a lot of work has been undertaken in recent years to include prediction uncertainty by using probabilistic load forecasting. In [9], the authors used WaveNet-like architecture to perform 24-hour-ahead predictions of load demand. One of the most popular models is the Deep autoregressive model (DeepAR) [22]. It involves an encoder-decoder structure that predicts future quantiles of load demands instead of actual values. More, multi-horizon quantile recurrent forecaster (MQ-RNN) [29] has been widely used. It similarly used a sequence-to-sequence (seq2seq) structure to make quantile predictions. Also, a convolution-based model known as DeepTCN [18] that uses stacked dilated casual convolutional layers has shown success in making multi-step ahead predictions. Moreover, following its success in natural language processing, the attention mechanism started gaining relevancy in load forecasting. The authors of [23] suggested a modified version seq2seq structure with an added attention mechanism.

3 BACKGROUND

3.1 Recurrent Neural Networks (RNNs)

Since we are dealing with time series, data points are dependent on one another, especially earlier ones. Thus, data order is important because it provides useful information about the examined time step. This type of dataset inspired researchers to develop a more convenient solution, known as sequence models, to deal with sequential datasets [31]. RNNs are a type of neural network that was designed to process sequential data. In RNNs, instead of using only the immediate input to make predictions, previous cells also provide information and context to ensure that time is introduced and captured in the input [24]. Given a sequence $\{x_1, x_2, ..., x_t\}$, where $x_i \in \mathbb{R}^n$, the RNN uses a defined recurrent function f, that depends on the structure of the RNN, to calculate a hidden state $h_t \in \mathbb{R}^m$ at each time step, t, as follows:

$$h_t = f(h_{t-1}, x_t) \tag{1}$$

Long short-term memory (LSTM) [12], is one of the possible extensions of RNN, that was developed to store long-term information and tackle the vanishing and exploding gradients problems in RNNs. An LSTM is composed of different hidden and gate states. Given a time step t, the LSTM calculates the hidden state h_t and the cell state c_t by using the following architecture [2]:

- A forget gate f_t that decides what part is kept and what part is dropped from the previous cell.
- An input gate i_t that decides which part of the information is going as input to the cell state.
- An output gate o_t that decides what part of the cell is going as output to the hidden state.
- A new cell state $\hat{c}_t$.
- A cell state c_t that forgets parts of the previous cell information and inputs some new cell information
- A hidden state h_t which represents the cell output

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{2}$$
$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{3}$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{4}$$
$$\hat{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{5}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t \tag{6}$$
$$h_t = o_t \odot \tanh(c_t) \tag{7}$$

where σ represents the sigmoid activation function, W_x and b_x denote the respective weights and bias of the respective gates (x).

3.2 Encoder-Decoder

An encoder-decoder structure [5] is an architecture that has gained interest in solving seq2seq problems. As the name suggests, it consists of two parts: Encoder and Decoder. The encoder tries to find a fixed-length context vector, that is a representation of input data, that is capable of capturing the features and the meaning of the input sequence. The encoder can be a simple RNN, an LSTM, a bidirectional LST, etc. On the other hand, the decoder takes the context vector and the encoded history as its input and generates the future sequence as outputs by learning to understand the context vector.

3.3 Attention mechanism

The attention mechanism was first introduced in seq2seq models to deal with natural language processing and machine translation tasks [1]. Yet, it gained attention in other fields as well [25]. Usually, it is related to the encoder-decoder structure, where it will help the model to detect automatically which parts of the encoder output are relevant to the decoder outputs. Suppose that we have the previous hidden states at different time steps $H = \{h_1, h_2, ..., h_{t-1}\}$. The context vector v_t will be a weighted sum of h_i where the weights decide which information is important at time step t. This soft attention mechanism can be described as a scoring function $f : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$ that calculates relevancy between its input states

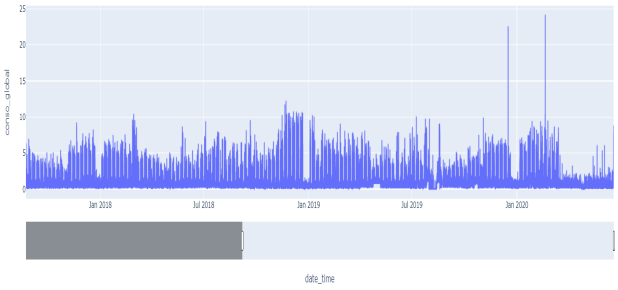

Figure 1: Sample of Grenoble Dataset

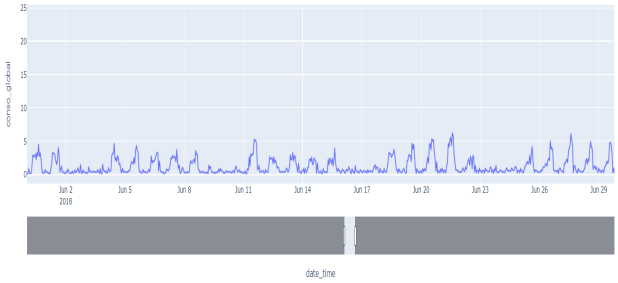

Figure 2: Sample of 1 month of Grenoble Dataset

as follows:

$$\alpha_i = \frac{\exp(f(h_i, h_t))}{\sum_{j=1}^{t-1} \exp(f(h_j, h_t))} \tag{8}$$
$$v_t = \sum_{i=1}^{t-1} \alpha_i h_i \tag{9}$$

where α is a learnable parameter that represents the weight of each input vector.

3.4 Temporal Convolutional Neural Network

4 DATASET

4.1 Dataset description

The dataset consists of measurements from a group of rooms at Green-ER building at Grenoble University [1][7, 15, 20]. The measurements are hourly load consumption collected in a time frame of 3 years (from 2017 to 2020). Figure 1 shows the global building consumption in the whole period while Figure 2 shows the consumption in one month. It can be seen that the observations are noisy, which makes it difficult to deal with. Figure 3 shows boxplots of consumption during normal days and holidays and as we can see holidays requires less electricity than normal days.

4.2 Dataset preprocessing

In addition to the hourly load consumption data, occupancy, and temperature measurements have been added to the dataset. Those features are known as exogenous input. The occupancy feature

[1]https://mhi-srv.g2elab.grenoble-inp.fr/django/API/

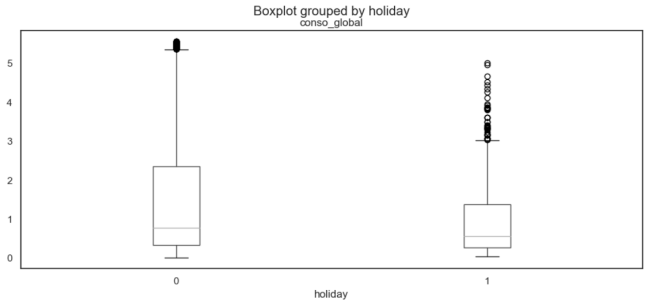

Figure 3: Boxplot grouped by holiday

assigns 0 or 1 depending on whether the room is occupied or not. Obviously, the building will be occupied mostly during work hours and working days. Before using the dataset, a lot of preprocessing was done like handling outliers and missing values. When an outlier is found, our approach was to replace the observation with other values of the same hour of the previous week. Previous week observation would be a better representation than previous day observation since the difference between different days of the same week could be high, say the difference between a successive weekend and weekday.

Moreover, the temperature feature has a direct impact on electricity consumption since heating systems would be used on colder days. Missing values of temperature data have been replaced by the temperature of the same hour of the previous day. Also, by conducting experiments, it was found that it is better to use daily average temperature instead of hourly temperature since the reaction to the variation of temperature is not instantaneous. Next, the occupancy feature is analyzed. Since it is not captured by a sensor but rather defined by the calendar of the school, there were no outliers or missing values to deal with. Yet, occupancy is very important in load forecasting since most of the time there is no need for electricity consumption in the absence of occupancy. Finally, global consumption itself is a feature. In fact, the consumption of each hour of the previous day as well as the consumption of each hour of the same day from the previous week will be fed as input to the model.

5 MODELS

5.1 Deep Autoregressive model

DeepAR [22] is an autoregressive neural network developed by Amazon. It offers multiple time-series support since it is trained on multiple time series. More, it leverages probabilistic output by predicting quantile loss rather than single prediction values. Most importantly, it offers flexibility by allowing a variable length of how far we can look back and how many steps ahead we want to make our predictions. The model uses LSTMs to make predictions.

First, the covariate $x_{i,t}$ of the current time step t, the label $z_{i,t-1}$ of the previous time step $t-1$, and the previous hidden state $h_{i,t-1}$ are fed to the LSTM cells. However, instead of using LSTM to predict outputs, DeepAR uses LSTMs to estimate the parameters $\theta = (\mu, \sigma)$ (mean and standard deviation) of the Gaussian function. In other words, at each time step t, the model goal is to estimate the best mean and standard deviation that parameterize a Gaussian function that predicts values that are close to the target value at t. The same thing is done at the next time step t+1 by feeding the hidden state of time step t, the target value at t alongside with the input value at time step t+1. The Gaussian distribution parameters estimation is also done differently. In fact, instead of using the maximum log-likelihood estimators, the model adds 2 dense layers after the LSTM to estimate those parameters. The hidden state at each time step t is fed to the first dense layer to compute weights that give the best estimation of the mean while the second dense layer calculates the weight that estimates the standard deviation. To simplify things, the estimation of the parameters by the two dense layers is equivalent to the following equations:

$$\mu(h_{i,t}) = w_\mu^T h_{i,t} + b_\mu \tag{10}$$

$$\sigma(h_{i,t}) = \log(1 + \exp(w_\sigma^T h_{i,t} + b_\sigma)) \tag{11}$$

where w_μ^T, w_σ^T, b_μ, and b_σ denote the weight and the bias of the dense layers and T represents the number of steps ahead to predict. The model then makes a distribution with these values and compares the output of the function to the actual target value. The calculation of weights of the dense layers is done by backpropagation during training. At the end of the training phase, we will have estimated distribution parameters for each time point.

5.2 Multi-horzion Quantile Recurrent Forecaster

Multi-horzion Quantile Recurrent Forecaster [29] is another model that uses RNNs to produce general probabilistic multi-step time series predictions. More specifically, it consists of the famous encoder-decoder structure and uses quantile regression to predict future horizons. Contrary to DeepAR which makes distributional assumptions, MQ-RNN uses quantile regression to estimate the conditional quantiles of the future values. The model is trained to minimize the total quantile loss which is defined by:

$$L_q(y, \hat{y}) = q(y - \hat{y})_+ + (1 - q)(\hat{y} - y)_+ \tag{12}$$

where $(.)_+ = max(0, .)$, y represents the target values, $\hat{y}$ represents the estimated values and q is the quantile.

Moreover, the model is trained to forecast all future horizons simultaneously, rather than iteratively. This approach helps in reducing the error propagation and computational cost of the model.

The MQ-RNN architecture is similar to the Seq2Seq structure but with some differences. The first part of the model is a Vanilla LSTM that encodes all history data into hidden state h_t. However, as opposed to the traditional seq2seq structure that uses LSTM as a recursive decoder, this model leverages two MLP branches. The first one is a global MLP that summarizes the encoder hidden state and outputs a horizon-specific context c_{t+k} for each future K (horizon) step and a horizon-agnostic context vector c_a that contains common information. The second one is a local neural network that is applied to each horizon. In other words, as an input, it takes the future input, the two context vector c_{t+k} and c_a and make quantiles prediction for each future K steps. The local MLP takes into account future seasonality and event which helps in making spikes in forecasts.

5.3 Deep Temporal Convolutional Network

Deep Temporal Convolutional Network [4] is a modified version of CNNs that is used to deal with sequential data. Mostly, CNNs do not perform well in sequence data, but with TCNN, autoregressive predictions with long-term memory are introduced. TCNN is similar to the traditional encoder-decoder structure but with CNN in this case. The encoder part stacks different convolutional layers that have the same size as the input on top of each other. Mainly, it consists of two methods:

- **Casual Convolution**: This technique ensures the output of the convolutions at time step t depends only on the past and present inputs. By padding the input sequence, casual convolutions will have the same size as the input layer and will preserve the temporal order of the data.
- **Dilated Convolution**: This type of convolution uses a dilation factor to skip some of the input values and enable a larger receptive field of the networks without increasing the model complexity. In this case, the model will capture long-term dependencies. In most cases, Dilated convolutions are combined with casual ones to form dilated casual convolutions.

On the other hand, the decoder consists of two parts. The first is a resnet-v module [14] while the second consists of a fully connected network that takes the output of the resnet-v module as input and provides probabilistic forecasts as output. The resnet module can be written as follows:

$$\delta_{t+\omega}^{(i)} = R(X_{t+\omega}^{(i)}) + h_t^{(i)} \tag{13}$$

where $h_t^{(i)}$ and $X_{t+\omega}^{(i)}$ denote the output of the encoder and the future covariates respectively. $R(\cdot)$ corresponds to the residual function.

5.4 Dual-stage Attention-based Recurrent Neural Network

As its name suggests, dual-stage attention-based recurrent neural network (DA-RNN) [21] consists of two attention layers that are added to the classical seq2seq structure. Mainly, the two stages of DA-RNN are:

- **Encoder with input attention**: In this stage, the model uses the attention mechanism described in Section 3 to select relevant driving series (input features) at each time step t. Suppose we have an input sequence $X = (x_1, x_2, ..., x_T)$ with $x_t \in \mathbb{R}^n$, where n represents the number of exogenous variables and T represents the window size. At each time step t, given n driving series, we will associate to each of the driving series a learnable attention weight that measures the importance of that feature at that time step by referring to the previous hidden state h_{t-1} of the encoder. The output of the input attention layer at time step t will be $\tilde{x}_t = (\alpha_t^1 x_t^1, \alpha_t^2 x_t^2, \ldots, \alpha_t^n x_t^n)$ which is a weighted version of the input sequence (driving series) at time step t. Next, the encoder hidden state h_t at time step t is updated using the LSTM layers that will refer to the previous hidden state h_{t-1} and the newly calculated $\tilde{x}_t$. Hence, the encoder pays more attention to some input features than the others.

Table 1: Results of 24-hour ahead predictions

Model	RMSE	MAE	R^2
DeepAR	0.179	0.128	0.518
MQ-RNN	0.156	0.112	0.672
DeepTCN	0.158	0.12	0.653
DA-RNN	0.181	0.130	0.501

- **Decoder with input attention**: Similarly, in the decoder stage, we use another LSTM network to decode the encoded input sequence. But, before going into the decoder, a temporal attention layer is added. This time, as the name indicates, the attention is performed across all time steps rather than across the driving series. In other words, we calculate the weight of each of the encoder hidden state at time step t by referring to the previous decoder hidden state d_{t-1}. Therefore, a context vector c_t is computed by the attention mechanism as a weighted sum of the encoder's hidden states. Finally, the decoder uses these context vectors in its LSTM units to estimate the target variable. This second stage helps the model capture the long-term temporal dependencies of the time series.

6 EXPERIMENTAL RESULTS

6.1 Evaluation metrics

Since we are making probabilistic predictions, we used the coefficient of determination of the regression score, also known as R^2 to quantify the accuracy of the predictions. We used root mean squared error (RMSE) and mean absolute error (MAE) as well. The metrics are defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(A_i - F_i)^2} \tag{14}$$

$$MAE = \Sigma_{i=1}^{N}|A_i - F_i| \tag{15}$$

$$R^2 = 1 - \frac{SSR}{SST} \tag{16}$$

$$SSR = \Sigma_{i=1}^{N}(A_i - F_i)^2 \tag{17}$$

$$SST = \Sigma_{i=1}^{N}(A_i - \hat{A}_i)^2 \tag{18}$$

where A_i and F_i represent the true and the predicted values of the i^{th} data point, N is the number of samples, and $\hat{A}$ is the average of the true data. SSR and SST stand for the sum of squared regression and the total sum of squared, respectively.

6.2 24-hours ahead forecasting

In the inference phase, to forecast the future 24 hours, one week of past data is fed to the models. Table 1 shows the performance of the four models in making 24-hour-ahead predictions. We can see that MQ-RNN outperforms its peers with an R^2 score of 0.67. The closest-performing model to MQ-RNN is the DeepTCN with an R^2 score of 0.65. However, DeepAR and DA-RNN did not perform as good as the first two models since the R^2 decreased by 15% compared to the first two models. Additionally, MQ-RNN has the lowest error rates of RMSE and MAE.

Table 2: Results of 1-week ahead predictions

Model	RMSE	MAE	R^2
DeepAR	0.178	0.127	0.469
MQ-RNN	0.125	0.098	0.542
DeepTCN	0.194	0.126	0.538
DA-RNN	0.202	0.133	0.455

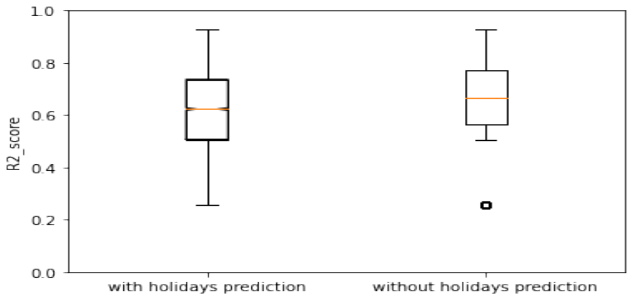

Figure 4: Influence of vacations

6.3 1-week ahead forecasting

Similarly, to forecast the next week's consumption values, the models are fed values from 1 previous week and aim to predict the next week which is equivalent to predicting 168 hours ahead. Table 2 shows the performance of the four models in making 1-week-ahead predictions. Again, we can see that MQ-RNN outperforms the other models but with a smaller gap than in the case of 24-hour ahead predictions. The R^2 score of the MQ-RNN for 1-week-ahead decreased to 0.542 versus 0.672 in 24-hour-ahead predictions. A similar decrease happened to the DeepTCN model. On the other hand, the other two models did not decrease by a lot by increasing the horizon size.

6.4 Discussions

Figure 5 and Figure 6 show samples from the predicted test results for MQ-RNN 1 day and 1 week, respectively. We can see that the model performs well in medium to high load consumption. However, it tends to suffer from spikes and low values of electricity consumption. These low values usually correspond to holidays or vacations, which indicates that these models perform poorly in such conditions due to the lack of holiday data. This was validated by conducting an ablative study where we removed predictions made for vacation periods to highlight the influence of vacation days and weeks. Figure 4 confirms that vacation periods decrease the quality of predictions. It is seen that the median of the boxplot of the R^2 score is higher when we exclude the presence of holiday periods. Moreover, the better performance of MQ-RNN compared to the other models can be explained by the fact that this model is conducting multi-horizon forecasts at a time rather than point forecasting which predicts values one by one. Finally, adding occupancy and temperature features discussed in Section 4 improved the predictions slightly but did not solve holiday issues completely.

7 CONCLUSION

In this paper, we investigate one of the challenging topics in the field of buildings and energy, which is the problem of load forecasting. To do that, we presented and discussed the Grenoble University building dataset. Given that the dataset is non-aggregated, it was difficult to handle especially in tricky situations like holidays and vacations. An extensive analysis and preprocessing of the dataset were done and several deep-learning models were tested. The models provide probabilistic predictions for load forecasting. We demonstrated that these models provide promising results in most cases except for holidays and vacations.

To conclude, future improvements could be to consider applying other state-of-the-art models in load forecasting, especially more developed attention-based ones. One could test the addition of other features too like humidity and inside temperature or the improvement of existing features like occupancy to make it take into consideration the vacations and the rate of attendance in rooms instead of just a binary feature. Finally, a clustering algorithm that groups data into different clusters (low, average, and high consumption) and then selects the best-performing model for each cluster is advised. This could alleviate the problem of holidays by applying a specific model that works well on vacation periods.

ACKNOWLEDGMENTS

The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC) and a start-up Grant from Concordia University.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
[2] Omar Bouhamed. 2022. *AI-Powered Time Series Forecasting Frameworks For Building Energy Management Systems.* Master's thesis. Concordia University.
[3] Long Cai, Jie Gu, and Zhijian Jin. 2019. Two-layer transfer-learning-based architecture for short-term load forecasting. *IEEE Transactions on Industrial Informatics* 16, 3 (2019), 1722–1732.
[4] Yitian Chen, Yanfei Kang, Yixiong Chen, and Zizhuo Wang. 2020. Probabilistic forecasting with temporal convolutional neural network. *Neurocomputing* 399 (2020), 491–501.
[5] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).
[6] Javier Contreras, Rosario Espinola, Francisco J Nogales, and Antonio J Conejo. 2003. ARIMA models to predict next-day electricity prices. *IEEE transactions on power systems* 18, 3 (2003), 1014–1020.
[7] Benoit Delinchant, Gustavo Martin, Tiansi Laranjeira, Muhammad Salman Shahid, Frederic Wurtz, et al. 2021. Machine Learning on Buildings Data for Future Energy Community Services. In *SGE 2021-Symposium de Génie Electrique.*
[8] Xishuang Dong, Lijun Qian, and Lei Huang. 2017. Short-term load forecasting in smart grid: A combined CNN and K-means clustering approach. In *2017 IEEE international conference on big data and smart computing (BigComp).* IEEE, 119–125.
[9] Fernando Dorado Rueda, Jaime Durán Suárez, and Alejandro del Real Torres. 2021. Short-term load forecasting using encoder-decoder wavenet: Application to the french grid. *Energies* 14, 9 (2021), 2524.
[10] Wei Gao, Jalal Alsarraf, Hossein Moayedi, Amin Shahsavar, and Hoang Nguyen. 2019. Comprehensive preference learning and feature validity for designing energy-efficient residential buildings using machine learning paradigms. *Applied Soft Computing* 84 (2019), 105748.
[11] Alfonso Gonzalez-Briones, Guillermo Hernandez, Juan M Corchado, Sigeru Omatu, and Mohd Saberi Mohamad. 2019. Machine learning models for electricity consumption forecasting: a review. In *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS).* IEEE, 1–6.
[12] Alex Graves and Alex Graves. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks* (2012), 37–45.

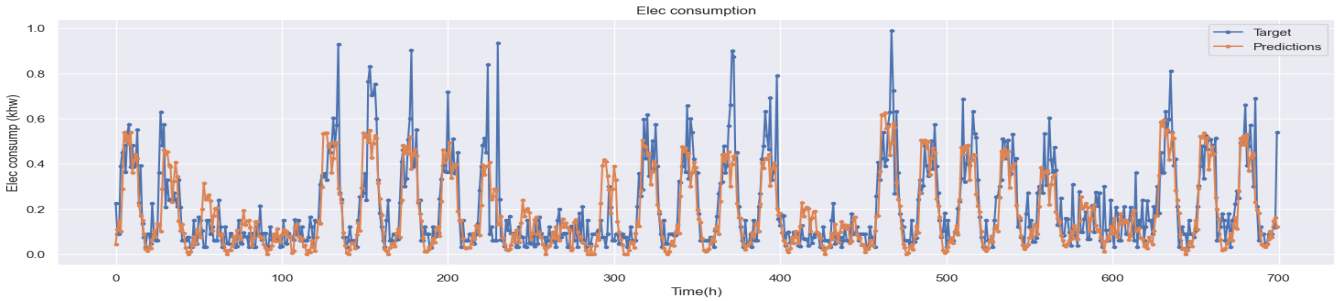

Figure 5: MQ-RNN 1-day predictions

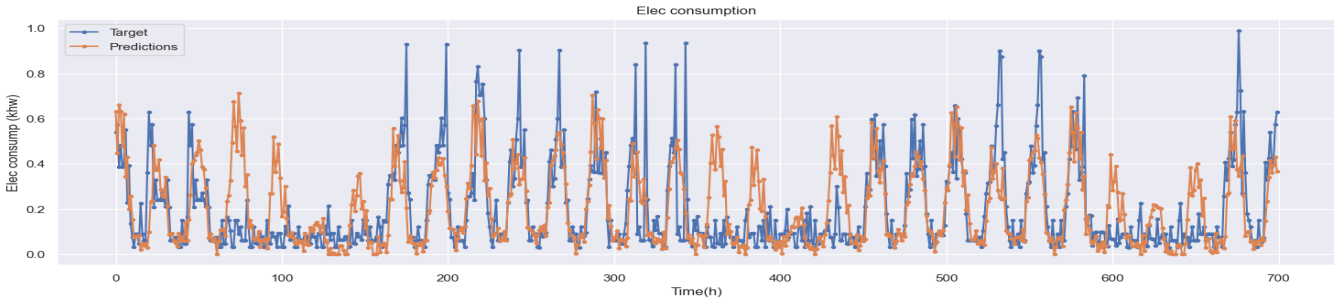

Figure 6: MQ-RNN 1-week predictions

[13] Vikas Gupta and Seema Pal. 2017. An overview of different types of load forecasting methods and the factors affecting the load forecasting. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)* 5, IV (2017), 729–733.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[15] Sacha Hodencq, Benoit Delinchant, and Frédéric Wurtz. 2021. Open and Reproducible Use Cases for Energy (ORUCE) methodology in systems design and operation: a dwelling photovoltaic self-consumption example. In *Building Simulation 2021*.

[16] Ehsan Kamel, Shaya Sheikh, and Xueqing Huang. 2020. Data-driven predictive models for residential building energy use based on the segregation of heating and cooling days. *Energy* 206 (2020), 118045.

[17] Jelena Krstić, Marija Reljić, and Sanja Filipović. 2019. Factors influencing electricity consumption: a review of research methods. *Management: Journal of Sustainable Business and Management Solutions in Emerging Economies* 24, 2 (2019), 13–22.

[18] Yang Lin, Irena Koprinska, and Mashud Rana. 2021. Temporal convolutional attention neural networks for time series forecasting. In *2021 International joint conference on neural networks (IJCNN)*. IEEE, 1–8.

[19] Chujie Lu, Sihui Li, and Zhengjun Lu. 2022. Building energy prediction using artificial neural networks: A literature survey. *Energy and Buildings* 262 (2022), 111718.

[20] Gustavo Nascimento, Benoit Delinchant, Frédéric Wurtz, Patrick Kuo-Peng, Nelson Jhoe Batistela, and Tiansi Laranjeira. 2020. GreEn-ER - Electricity Consumption Data of a Tertiary Building. *Mendeley Data* (2020). https://doi.org/doi:10.17632/h8mmnthn5w.1

[21] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. 2017. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971* (2017).

[22] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* 36, 3 (2020), 1181–1191.

[23] Ljubisa Sehovac and Katarina Grolinger. 2020. Deep learning for load forecasting: Sequence to sequence recurrent neural networks with attention. *IEEE Access* 8 (2020), 36411–36426.

[24] Alex Sherstinsky. 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena* 404 (2020), 132306.

[25] Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. 2019. Temporal pattern attention for multivariate time series forecasting. *Machine Learning* 108 (2019), 1421–1441.

[26] Statista. 2022. Net electricity consumption worldwide in select years from 1980 to 2021. https://www.statista.com/statistics/280704/world-power-consumption/

[27] Souhaib Ben Taieb and Amir F Atiya. 2015. A bias and variance analysis for multistep-ahead time series forecasting. *IEEE transactions on neural networks and learning systems* 27, 1 (2015), 62–76.

[28] U.S. Energy Information Administration. 2023. Use of electricity. https://www.eia.gov/energyexplained/electricity/use-of-electricity.php

[29] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. 2017. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053* (2017).

[30] Baran Yildiz, Jose I Bilbao, and Alistair B Sproul. 2017. A review and analysis of regression and machine learning models on commercial building electricity load forecasting. *Renewable and Sustainable Energy Reviews* 73 (2017), 1104–1122.

[31] Hana Yousuf, Michael Lahzi, Said A Salloum, and Khaled Shaalan. 2021. A systematic review on sequence-to-sequence learning with neural network and its models. *International Journal of Electrical & Computer Engineering (2088-8708)* 11, 3 (2021).

[32] Jian Zheng, Cencen Xu, Ziang Zhang, and Xiaohua Li. 2017. Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network. In *2017 51st Annual conference on information sciences and systems (CISS)*. IEEE, 1–6.