

Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks

Aowabin Rahman
Dept. of Mechanical Engineering
University of Utah

Vivek Srikumar
School of Computing
University of Utah

Amanda D. Smith
Dept. of Mechanical Engineering
University of Utah

Abstract

This paper presents a recurrent neural network model to make medium-to-long term predictions, i.e. time horizon of ≥ 1 week, of electricity consumption profiles in commercial and residential buildings at one-hour resolution. Residential and commercial buildings are responsible for a significant fraction of the overall energy consumption in the U.S. With advances in sensors and smart technologies, there is a need for medium to long-term prediction of electricity consumption in residential and commercial buildings at hourly intervals to support decision making pertaining to operations, demand response strategies, and installation of distributed generation systems. The modeler may have limited access to information about building's schedules and equipment, making data-driven machine learning models attractive. The energy consumption data that is available may also contain blocks of missing data, making time-series predictions difficult. Thus, the main objectives of this paper are: (a) Develop and optimize novel deep recurrent neural network (NN) models aimed at medium to long term electric load prediction at one-hour resolution; (b) Analyze the relative performance of the model for different types of electricity consumption patterns; and (c) Use the deep NN to perform imputation on an electricity consumption dataset containing segments of missing values. The proposed models were used to predict hourly electricity consumption for the Public Safety Building in Salt Lake City, Utah, and for aggregated hourly electricity consumption in residential buildings in Austin, Texas. For predicting the commercial building's load profiles, the proposed NN sequence-to-sequence models generally correspond to lower relative error when compared with the conventional multi-layered perceptron neural network. For predicting aggregate electricity consumption in residential buildings, the proposed model generally does not provide gains in accuracy compared to the multi-layered perceptron model.

Keywords: Building Energy Modeling, Machine Learning, Recurrent Neural Networks, Deep Learning, Electric Load Prediction

NOMENCLATURE

<i>DL</i>	Deep learning
<i>DNN</i>	Deep neural network
<i>EI</i>	Expected Improvement
<i>LSTM</i>	Long short term memory
<i>ML</i>	Machine Learning
<i>MLP</i>	Multi-layered perceptron
<i>NN</i>	Neural network
<i>PSB</i>	Public Safety Building
<i>RNN</i>	Recurrent neural network
<i>SMBO</i>	Sequential-model based Optimzation
<i>TPE</i>	Tree of Parzen Estimator
γ	Learning rate in gradient descent algorithm
σ	Sigmoid function serving as a gating function
\circ	Element-wise vector multiplier.
τ_i	Characteristic timescales in a periodic energy consumption profile
λ	Parameter for weight regularization.
μ	Number of training epochs, i.e. number of runs for which the model is trained using the entire training set.
θ	Set of hyper-parameters
θ^*	Optimal set of hyper-parameters
\mathbf{c}_t	Transient 'memory' value in LSTM function
e	Mean squared error in predicting electricity consumption
\mathbf{f}	Frequency-related variables used as inputs to the deep RNN model.
\mathbf{g}	Input activation function in LSTM
\mathbf{h}_t	Output of LSTM function at given timestep t
h_j^m	Value of hidden node in a neural network in node j , layer m .
<i>DNN</i>	Deep neural network
\mathbf{i}	Input gate in LSTM
n	Number if residential buildings considered when aggregate electricity consumption profile in residential buildings.
\mathbf{o}	Output gate in LSTM
p	Fraction of consecutive missing data points relative to the size of the entire training set.

q_1	Fraction of data points prior to the missing block p , relative to the size of the entire training set.
q_2	Fraction of data points after the missing block p , relative to the size of the entire training set.
s	Parameter to describe discrepancy between electricity consumption in test data and that in the corresponding training data.
w_{ji}	Weight connecting j in layer m to node i in layer $m - 1$
w_{ji}	Weight connecting j in layer m to node i in layer $m - 1$
\mathbf{w}	Weather variables used as inputs to the deep RNN model.
\mathbf{x}_t	Input to LSTM activation corresponding to a previous layer and current timestep t .
\mathbf{X}	Feature vector used as inputs to the deep RNN model.
y_p	Predicted value of electricity consumption
y_a	Actual value of electricity consumption.
\mathbf{s}	Date-related variables used as inputs to the deep RNN model.
w_{ji}	Weight connecting j in layer m to node i in layer $m - 1$
\mathbf{w}	Weather variables used as inputs to the deep RNN model.
\mathbf{x}_t	Input to LSTM activation corresponding to a previous layer and current timestep t .
\mathbf{X}	Feature vector used as inputs to the deep RNN model.
y_p	Predicted value of electricity consumption
y_a	Actual value of electricity consumption.

1 Introduction

Residential and commercial buildings in 2015 were responsible for approximately 73% of the electricity consumption and 41% of primary energy consumption in the U.S, with the values projected to increase over the next 20 years [1]. There has been a growing emphasis on the development and implementation of smart grids and smart buildings in order to meet these electricity demands in an efficient and cost-effective manner while minimizing greenhouse emissions [2, 3]. The case for smart grids are further strengthened by the increasing intermittent, renewable energy resources such as wind and solar, as well as a growing number of small-scale distributed generation systems [2, 4].

As such, dynamic planning and management of smart buildings and smart grid systems, while integrating intermittent renewables and distributed generation resources, requires accurate forecasting of electricity consumption over different time horizons [2]. Based on the time horizon of prediction, Mocanu et al. [5] grouped electricity demand forecasting into three categories: (i) Short-term forecasts ranging between one hour to one week, (ii) medium term forecasts between one week to one year and (iii) long-term forecasts spanning a time period of more than one year. Short-term forecasts are generally useful for generation capacity scheduling and short-term maintenance, evaluation of short-term energy storage usage, as well as real-time control of building energy systems and optimization of fuel purchase plans [5, 6, 7, 8]. On the other hand, medium to long term forecasts are used to make decisions pertaining to the installation of new distributed generation and

storage systems [9], as well as develop suitable demand response strategies [5]. At a regional level, forecasting of aggregated electricity consumption over medium-to-long term time horizons can be useful for planning and trading on electricity markets [10].

The approaches to estimating electricity demand in buildings can be physics-based or data-driven [11, 12]. Physics-based or deterministic models, such as those employed by EnergyPlus and eQuest, usually formulate and solve heat and mass balance equations interconnecting the different zones, air handling and equipment systems inside a building [13]. However, these physics-based models often do not account for the complex energy consumption behavior in a building, and sometimes input parameters required by these models, are difficult to obtain in practice [11]. The resulting approximations often lead to a loss in accuracy, sometimes in excess of 100% [11, 14], and as such, these models are often used as comparative tools rather than accurate predictors of building energy consumption.

Statistical and machine learning (ML) models provide an alternative to such physics-based models [11, 12]. Previous work has employed simple linear regression [12, 15, 16], multi-variate linear regression [15], non-linear regression [15, 12], support vector machines [11, 12, 17], Gaussian Process regression [18], multi-layered perceptron neural networks [19, 12, 11, 16, 17], and auto-regressive neural networks [11, 20] in predicting building energy consumption. Hybrid models that couple physical models, i.e. thermal networks, with statistical and/or ML models have also been proposed [21]. These methods, in general, have been shown to achieve high accuracy for forecasting over a time horizon of one hour [20] to one week [22], the amount of work pertaining to medium to long term predictions at hourly or sub-hourly intervals has been relatively limited. The latter is a more difficult objective, with previous work showing that the relative errors corresponding to medium to long term predictions at one-hour resolution often in excess of 40-50% [5, 15, 23].

Deep neural networks [24] could potentially improve on the performances obtained using the aforementioned machine learning methods, as they allow for modeling of more complex functions by using multiple layers of abstraction [24], and are recently being employed in the energy forecasting context. Debinet et al. [7] used a deep belief network for electricity forecasting in Macedonia over a time horizon of 24 hours, which consisted of stacks of restricted Boltzman machines (RBMs) pre-trained layer-wise. Mocanu et al. [5] employed a conditional restricted Boltzman machine (CRBM) and factored conditional Boltzmann machine (FCRBM) to predict electricity power consumption in a residential building. The two deep learning (DL) methods were used to obtain results for multiple cases, each case corresponding to a combination of time resolution and time horizon. The authors found that for a week-ahead prediction at one-hour resolution, the relative errors in predicting aggregate power corresponding to CRBM and FCRBM were 60.0% and 63.3%, whereas for a year-ahead prediction at one-day resolution, the corresponding errors were 18.2% and 17.0% [5].

The electricity consumption behavior is inherently transient in nature, and the consumption pattern (as detailed later in the paper) can be shaped by long-term dependencies. The idea of incorporating temporal dependencies of energy data along different timescales using a feature method was explored by Arahali et al. [25] and Fan et al. [26] in the context of short-term load forecasting. Recurrent neural networks (RNNs) are one family of algorithms that can accommodate dependencies between consecutive time steps. However, as mathematically shown by Hochreiter and Schmidhuber, vanilla RNNs (i.e. those which do not account for long-term dependencies) suffer from the problem of vanishing/exploding gradient, which makes learning long-term dependencies difficult [27]. Hochreiter and Schmidhuber [27] suggested recurrent neural networks with long short-term memory (LSTM) units as a possible solution to the vanishing gradient problem noticed in simple

RNNs [27, 28]. This allowed the RNN models with LSTM units to model both short and long term temporal dependencies in time-series data.

Thus, this paper proposes two novel deep RNN models with LSTM units in order to forecast building electricity consumption at one-hour resolution over medium to long-term time horizon, by treating the problem as a sequence to sequence learning problem. The model uses a multi-layered perceptron neural network stacked on top of an LSTM-based model using an encoder-decoder architecture, which has been successfully employed for sequence to sequence modeling. The deep RNNs provide a more expressive feature space, while accommodating both long and short-term temporal dependencies (as discussed by Arahal et al. [25]) using adaptive LSTM units. The sequence to sequence approach has been previously employed in speech recognition and machine translation applications [29, 30, 28], and while the approach has been previously used in short-term weather forecasting [31], its application in energy prediction context has been largely unexplored.

This paper presents a novel approach that can potentially address the limitations of longer term predictions observed in previous literature. We investigate the hypothesis that the sequence to sequence learning approach using the proposed model can take advantage of energy consumption patterns that exist over a given timescale—for instance, over the period of 24 hours, in order to make predictions over a relatively longer time horizon.

Electricity consumption data, which is used to train the model, may in practice contain significantly large segments where the electricity data is missing or corrupt [32]. Thus, as one of the secondary objectives, a simple data imputation scheme based on the proposed RNN model is suggested, that can perform imputation to fill in electricity consumption data where a comparatively large segment of data is missing.

The key contributions of this paper are:

- Develop and optimize LSTM-based deep recurrent neural network models to make predictions over the time horizon of a few months to one year at one-hour resolution.
- Quantify the performance of these models on multiple types of electricity consumption profiles corresponding to different load profiles in a commercial building, as well as aggregated electricity consumption profiles in residential buildings at a community-scale.
- Use the deep RNN model to develop a data imputation scheme for missing values in electricity consumption data.

Section 2 details the theoretical background of DL methods used in the model, including LSTM, and encoder-decoder architecture. Section 3 presents the formulation of the deep RNN models and the hyper-parameter optimization process, as well as a description of the missing-value imputation scheme. Section 4 describes the characteristics of the energy data on which proposed models were tested, and details how the model was implemented. Section 5 evaluates and analyzes how the model performed at predicting different types of load patterns. Section 6 summarizes the key results and discusses pathways for future research.

2 Theoretical Background

2.1 Neural Networks

Neural networks are a family of machine learning algorithms that can model non-linear relationships between input vectors to target values. A simple multi-layered perceptron (MLP) network, as shown

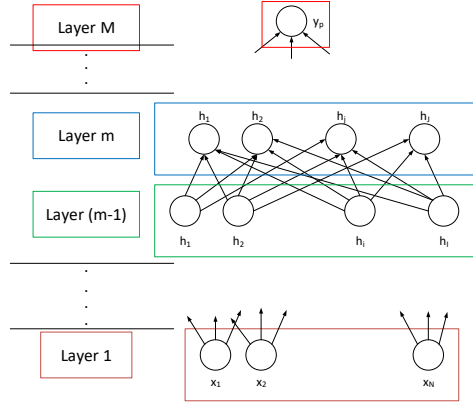


Figure 1. Schematic Diagram of a multi-layered perceptron neural network

in figure 1, consists of M layers with each layer containing a set of neurons. The output from a given node j in layer l can be computed as:

$$h_j^m = \phi \left(\sum_i w_{ji}^m h_i^{m-1} \right) = \phi(s_j) \quad (1)$$

As the outputs are continuous, the error can be computed as:

$$e = (y_p - y_a)^2 \quad (2)$$

After processing each sample during the training phase, the weights in each layer are updated as follows:

$$w_{ji}^m \leftarrow w_{ji}^m - \gamma \frac{\partial e}{\partial w_{ji}^m} \quad (3)$$

Here γ is the learning rate. The partial derivatives $\frac{\partial e}{\partial w_{ji}^m}$ is determined through back propagation [24].

2.2 Recurrent Neural Networks and Long Short-term Memory

Recurrent Neural Networks are NNs that model temporal dependencies present in time series data through the use of feedback connections, in order to ‘remember’ the values at previous time steps. For instance, an activation function in an RNN that considers the value at one prior time step can be expressed as:

$$h_{j,t}^m = \mathbf{w}_1 \phi(h_{j,t-1}^m) + \mathbf{w}_2 \phi(h_{i,t}^{m-1}) \quad (4)$$

Here, $h_{j,t}^m$ is the output of the activation function of node j in layer m at timestep t . Thus, it had been suggested that in a conventional RNN, the dependencies of node output at a given timestep relate to outputs at previous timesteps using the back-propagation through time algorithm [27].

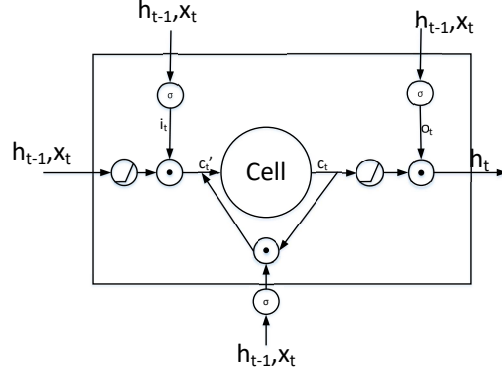


Figure 2. Schematic diagram of a long short-term memory (LSTM) activation function

However, gradients corresponding to these dependencies tend to vanish or explode over long time intervals, thus making the process of learning long-term dependencies difficult. This has been mathematically shown by Hochreiter and Schmidhuber [27]. The authors suggested that this problem of vanishing/exploding gradient can be rectified by using Long Short-term Memory (LSTM) activation functions in lieu of activation nodes in conventional RNN's. Depending on the inputs i.e., $(h_{t-1}, x_t) = (h_{j,t-1}^l, h_{i,t}^{m-1})$, an LSTM activation function adaptively scales the input, remembers or forgets the transient cell state value, and scales the activation function output. These are done using input, forget and output gates respectively.

Figure 2 shows a schematic diagram of an LSTM activation. Once an input \mathbf{x}_t enters the LSTM cell, it is separately passed through an activation function, g and an input gate i :

$$\mathbf{g} = \phi(\mathbf{w}_{g1}\mathbf{h}_{t-1} + \mathbf{w}_{g2}\mathbf{x}_t + \mathbf{b}_g) \quad (5)$$

$$\mathbf{i} = \sigma(\mathbf{w}_{i1}\mathbf{h}_{t-1} + \mathbf{w}_{i2}\mathbf{x}_t + \mathbf{b}_i) \quad (6)$$

Here σ is the sigmoid activation function applied to each element inside the parentheses. The element-wise product between \mathbf{g} and \mathbf{i} would be subsequently used to compute the transient 'memory' value of the activation function \mathbf{c}_t . However, \mathbf{c}_t also requires knowledge of the forget gate f , which can be computed as:

$$\mathbf{f} = \sigma(\mathbf{w}_{f1}\mathbf{h}_{t-1} + \mathbf{w}_{f2}\mathbf{x}_t + \mathbf{b}_f) \quad (7)$$

The output gate \mathbf{o} , which would be used to scale the output of the LSTM activation function, is expressed as:

$$\mathbf{o} = \sigma(\mathbf{w}_{o1}\mathbf{h}_{t-1} + \mathbf{w}_{o2}\mathbf{x}_t + \mathbf{b}_o) \quad (8)$$

The transient memory value, \mathbf{c}_t is given as:

$$\mathbf{c}_t = \mathbf{i} \circ \mathbf{g} + \mathbf{c}_{t-1} \circ \mathbf{f} \quad (9)$$

Here, \circ is an element-wise multiplier. Thus \mathbf{i} and \mathbf{f} effectively scale \mathbf{g} and \mathbf{c}_{t-1} , depending on the inputs to the LSTM function. The LSTM output h_t at current timestep t , is finally expressed as:

$$\mathbf{h}_t = \mathbf{o} \circ \phi(\mathbf{c}_t) \quad (10)$$

Finally, the cell output h_t , scaled by o is given as:

$$\mathbf{h}_t = \mathbf{o} \circ \psi(\mathbf{c}_t) \quad (11)$$

During the training phase, the weights $\mathbf{w} = [\mathbf{w}_{g1}, \mathbf{w}_{g2}, \mathbf{w}_{i1}, \mathbf{w}_{i2}, \mathbf{w}_{f1}, \mathbf{w}_{f2}, \mathbf{w}_{o1}, \mathbf{w}_{o2}]$ and the bias vectors $\mathbf{b} = [\mathbf{b}_g, \mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o]$ are learned through back-propagation [24]. The LSTM unit is used as a neural unit for the models proposed in this paper. The gating functions in the LSTM allow for adaptive computation of the activation function output \mathbf{h}_t and transient memory value \mathbf{c}_t . As such, in the context of energy forecasting where long-term dependencies are likely, we develop our RNN models using the LSTM activation function.

2.3 Gradient Descent Algorithm

We used the ADAM algorithm to optimize the weights in each layer - which exhibits faster convergence than the conventional stochastic gradient descent [33]. ADAM is a first-order based gradient descent optimization algorithm that is computationally efficient, and is suitable for optimizing models with a large set of parameters. Rather than naively updating the weights with a constant learning rate (as was the case for the vanilla stochastic gradient descent), ADAM considers the bias-corrected estimates of the moving average of the gradient as well as the squared gradient. Details on the ADAM algorithm can be found in other literature [33].

3 Model Description

The deep RNN models with LSTM units presented in this paper were developed in order to predict electricity consumption values at one-hour resolution over a medium-to-long term time horizon. The proposed models were tested for two cases: (i) Predicting electricity consumption at the Salt Lake City Public Safety Building (PSB), segregated by end-uses, over an 83-day period; and (ii) Predicting overall electricity consumption in residential buildings (aggregated over multiple buildings), over a time horizon of one year.

3.1 Model Inputs

3.1.1 Inputs for forecasting Load Profiles in PSB, Salt Lake City

The proposed RNN models were tested on multiple load profiles in Public Safety Building at Salt Lake City, UT (table 1). Depending upon the type of load profile being modeled, the inputs to the model, the model input \mathbf{X} was a given combination of weather variables, schedule-related variables and frequency-related variables. The weather variables (\mathbf{w}) considered were dry-bulb temperature

Table 1. Description of different load profiles

Load Profile	Type	Input variables (X)
HVAC Critical	Continuous	$[\mathbf{w}, \mathbf{s}]$
HVAC Normal	Continuous	$[\mathbf{w}, \mathbf{s}]$
Convenience Critical	Continuous	\mathbf{s}
Convenience Power Normal	Continuous	\mathbf{s}
Elevator	Discrete	$[\mathbf{s}, \mathbf{f}]$
CRAC Critical	Discrete	$[\mathbf{w}, \mathbf{s}, \mathbf{f}]$
CRAC Normal	Discrete	$[\mathbf{w}, \mathbf{s}, \mathbf{f}]$

and relative humidity; whereas the schedule-related variables (\mathbf{s}) were: hour of day (between 1 to 24), day of week, the day in a given month and the month number. The weather data was obtained from Mesowest web portal [34], corresponding to the weather station at the William Browning Building, University of Utah, Salt Lake City, UT. The schedule-related variables could be either binary or real-valued. For instance, the day of the week was introduced a concatenation of seven binary variables. This means that a data point corresponding to Sunday can be expressed as [1 0 0 0 0 0 0]. The first column corresponds to a flag whether or not the day is Sunday, and similarly, the remaining six columns correspond to a flag for each of the six other days of the week. As explained shortly, the timescale variables were used to account for the fact that a 24-hour basis was selected as the reference sequence length (i.e. electricity consumption behavior pattern is most pertinent on a daily timescale), but for a given load profile, other timescales could exist.

Table 1 shows the combination of model inputs used for each load profile. HVAC and computer room air-conditioning (CRAC) load profiles are likely to depend strongly on weather variables. Furthermore, the CRAC load profiles appear to have characteristic timescales other than 24-hours which can be determined by performing a Fast Fourier Transform (FFT) analysis. The time periods $[\tau_1, \tau_2]$ corresponding to the first two peaks are [325, 351] hours. To incorporate the effect of these characteristic timescales, a timescale variable f is introduced:

$$f_i = t \text{ MOD } \tau_i \quad (12)$$

Here, t is the time (in hours) relative to a reference time, and f_i is the phase relative to a given timescale τ_i .

3.1.2 Inputs for forecasting Aggregate Residential Building Electricity Consumption Profiles

The inputs for prediction of long-term aggregate electricity consumption in Austin, Texas were (i) weather variables (\mathbf{w}): dry bulb temperature, relative humidity, wind-speed and solar irradiation and (ii) schedule-related variables (\mathbf{s}): hour of day (between 1 to 24), day of week, the day in a given month and the month number. The weather data was obtained from Mesowest web portal [34] corresponding to the East Austin RAWS weather station in Austin, TX.

3.2 Model Formulation

The electricity consumption profile of a complex commercial building is inherently transient and non-linear in nature, and as such, a predictive model should have the following desirable character-

istics:

- The proposed model should be adaptive, i.e. learn from data without any human intervention.
- It should be able to model non-linear behavior in electricity consumption.
- It should be able to model both short and long-term temporal patterns in electricity consumption.
- The proposed model would need to account for temporal dependencies without having explicit information about building operation schedules or construction.

Two deep RNNs using LSTM units are proposed in this paper in order to address these challenges. As mentioned previously, the LSTM unit can adaptively learn from training data and can model both short and long term dependencies in electricity consumption. Both of the models proposed employ a sequence-based approach - more specifically an encoder-decoder architecture to take advantage of the sequential nature of electricity consumption. The encoder-decoder architecture has been previously used in applications such as machine translations [29], and consists of two blocks: an encoder-like block that converts the inputs to a fixed vector representation, and a decoder-like block that maps the vector representation to a given target sequence. The RNN blocks analogous to encoder-decoder in the model consist of LSTM units, and the outputs from encoder-decoder model are used as inputs for the subsequent MLP model. The problem of predicting energy consumption in buildings differs slightly from the machine translation problem as it is a regression problem, and the sequence length is invariant.

The intuition behind developing these two models is that the RNN layers analogous to the encoder-decoder layers would generate transient variables in 24-hour sequences, that would act as surrogates for variables that represent events happening on a schedule. The LSTM unit at each time step in both the layers analogous to encoder-decoder would allow for modeling short and long term temporal dependencies. We will call the two presented models model A and model B respectively. The first three layers, i.e. the input layer and the two RNN layers are identical for both models. Model B differs from model A in that model A feeds a linear combination of vectors (i.e. a scalar value) to the hidden MLP layer at time step (within the 24-hour sequence), and model B applies a shared MLP layer across each time step (i.e. a shared MLP layer that accepts a vector at each time step) within the same 24-hour sequence. Thus model B has fewer weight parameters to optimize, and as such, can potentially compromise expressiveness for improved generalization. The differences between the two models will be explained in more detail later in this section.

3.2.1 Model A

Figure 3 shows the schematic of model A. The description of each layer in the model is given as follows:

Layer 1: The input at one-hour resolution is introduced in layer 1. The training input data contains N samples (where N is the number of 24-hour sequences, i.e., in this context, N is the total number of days in the training set) - each sample $\mathbf{X}_{\text{train}} \in \mathbb{R}^{24 \times d}$, where d is the dimension of input $\mathbf{X}_{\text{train}}$.

Layer 2: Layer 2 is the first LSTM layer, and is analogous to the encoder layer in the encoder-decoder layer. The output $\mathbf{h}_{t,e}$ at each timestep t in this layer can be expressed as:

$$\mathbf{h}_{t,e} = LSTM(\mathbf{h}_{t-1,e}, \mathbf{X}_{\text{train},t}) \quad (13)$$

The fixed vector representation \mathbf{c} is simply the output at the final timestep, and can be expressed as: $\mathbf{c} = \mathbf{h}_{T,e}$.

Layer 3: The decoder layer in the encoder-decoder model can be expressed as:

$$\mathbf{h}_{t,d} = LSTM(\mathbf{c}, \mathbf{h}_{t-1,d}) \quad (14)$$

The outputs from layer 3, $\mathbf{h}_{t,d}$ act as surrogates for ‘transient’ variables, which are not explicitly known. These ‘transient’ variables are analogous to the schedule variables used as inputs in deterministic energy simulation models.

Layer 4: Layer 4 consists of two operations. In the first operation, the outputs from Layer 3, $\mathbf{h}_{t,d}$ are concatenated with the original input vector $\mathbf{X}_{\text{train}}$. This can be expressed as follows:

$$\mathbf{X}'_t = [\mathbf{X}_{\text{train}}; \mathbf{h}_{t,d}] \quad (15)$$

The skip connection for \mathbf{X} ensures that the dependencies on the original input are retained. Subsequently \mathbf{X}' is expressed as a linear combination of the inputs at each timestep, such that $\mathbf{X} \in \mathbb{R}^{24}$:

$$\mathbf{X}''_t = X''_{t,i=1\dots 24} \leftarrow \sum_{j=1}^J w_j X'_{t,ij} \quad (16)$$

Layers 5 and 6: These two layers correspond to a multi-layered perceptron neural network (section 2.1) with one hidden layer that accepts \mathbf{X}' as input. The final prediction, \mathbf{y}_p is a vector of dimensions (24, 1) and can be expressed as follows:

$$\mathbf{y}_p = MLP(\mathbf{X}'') \quad (17)$$

During the training phase, the weight parameters in the deep RNN model are learned through back-propagation and weight update. After the weights are optimized during training, the prediction is made during the test phase using the test input \mathbf{X}_{test} :

$$\mathbf{y}_p = \text{Model } A(\mathbf{X}_{\text{test}}) \quad (18)$$

3.2.2 Model B

As mentioned previously, the first three layers of model B are identical to those in model A. Figure 4 shows a schematic of model B. The subsequent layers, i.e. layers 4, 5 and 6 are explained as follows:

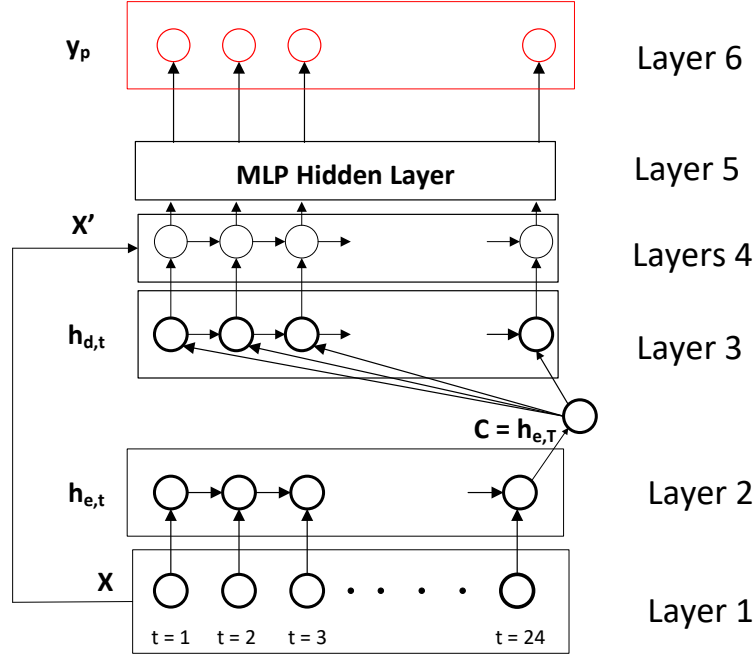


Figure 3. Schematic of Model A for Medium to Long-term forecasts. Circles in darker outlines represent a vector emitted from a given timestep, where the circles in lighter outlines represent a scalar value emitted from a given timestep. The circles in layer 1 represent the inputs to the model, as described in section 3.1. In layers 2 and 3, each circle represent an output vector at a given timestep after passing through an LSTM activation function. The outputs in layer 4 are scalar values at each time step, subsequent to concatenation and linear combination. Layer 5 is the hidden layer in a 3-layer multi-layered perceptron. The outputs in The final layer 6 is the output layer, as indicated by the red outline.

Layer 4: Unlike model A, layer 4 only consists of a single operation - concatenation of outputs from layer 3 with the original input.

$$\mathbf{X}'_t = [\mathbf{X}_{\text{train}}; \mathbf{h}_{t,d}] \quad (19)$$

Layers 5 and 6: In model B, the multi-layered perceptron is applied to the concatenated vector at each time-step. This is in contrast to model A, where the input to the MLP is a scalar at each timestep. Thus, the prediction $y_{p,t}$ for a given timestep t can be expressed as follows:

$$y_{p,t=1,2,\dots,24} = MLP(\mathbf{X}'_{t=1,2,\dots,24}) \quad (20)$$

To summarize, unlike model A, model B does not account for the temporal dependencies within a sequence beyond the recurrent layers (i.e. layers 2 and 3). The prediction using model B is made as follows:

$$\mathbf{y}_p = \text{Model } B(\mathbf{X}_{\text{test}}) \quad (21)$$

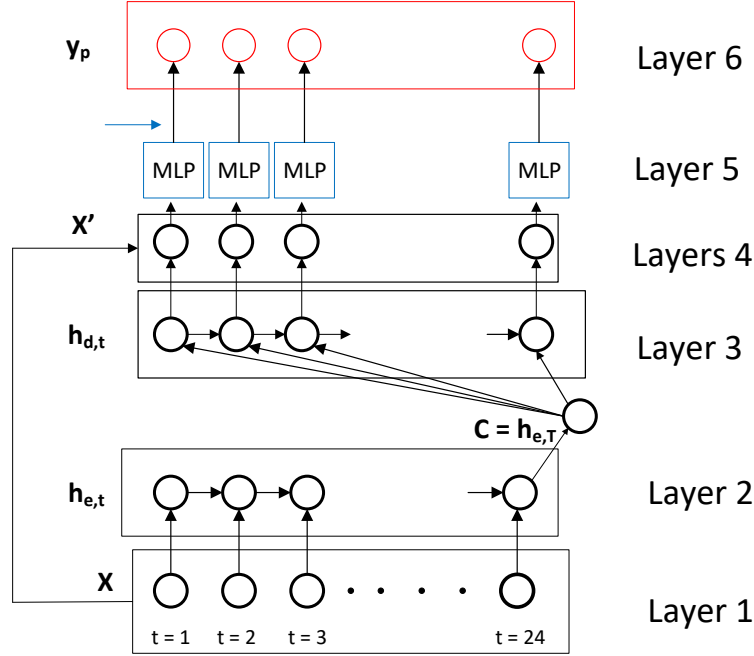


Figure 4. Schematic diagram of Model B. The layers 1,2 and 3 are identical to those in model A. Layer 4 is the concatenation layer emitting a vector at each timestep. Layer 5 is a shared MLP hidden layer applied to each timestep, as indicated by the arrow. Layer 6 is the output layer indicating the load prediction.

3.3 Model Regularization

Machine learning algorithms often suffer from over-fitting, which results in prediction accuracy of the algorithm in practice that is significantly worse than the training error. This occurs when the model fits the noise and small perturbations in the training data. The following regularization methods are adopted in this analysis to minimize overfitting:

- **Weight Decay Regularization:** Weight decay regularization is a simple way to constrain the model, such that the weight vectors are penalized for being too large. Thus, constraining the magnitude of the weights ensures that the outputs from each layer are not as sensitive to noise in input to that given layer. The error function can thus be expressed as:

$$e(\mathbf{w}) = (y_p(\mathbf{w}) - y_a(\mathbf{w}))^2 + \frac{\lambda}{2M} \mathbf{w}^T \mathbf{w} \quad (22)$$

In this analysis, λ is chosen to be 0.01, which is the default in Keras [35]. As the energy consumption data is normalized such that y_a and y_p is $\sim O(0.1)$, setting $\lambda = 0.01$ ensures that the error is penalized when $\|\mathbf{w}\| \gg 1$.

- **Early-stopping** : Early-stopping is a generalization method that stops the training process before completing the maximum number of iterations by monitoring the validation loss [36]. It is probable that the model is overfitting the data when the validation error starts to increase with increasing number of epochs (i.e. one iteration over the entire training data) during training. In this study, early-stopping is employed when, after μ iterations, the validation error has not improved. The early-stopping criterion is only employed after training has exceeded a minimum number of epochs. In this analysis, μ was chosen to be 10, and the minimum number of epochs was chosen to be 20.

3.4 Hyper-parameter Optimization

Hyper-parameters are higher-level modeling choices that are not optimized through the data itself, but are assigned empirically, or determined by optimization of the model structure itself. The RNN model presented is a graphical structure, and contains the following key hyper-parameters: (i) Length of output vector from layer 2 (i.e. number of nodes in layer 2) (ii) Length of output vector from layer 3 - or the number of surrogates for ‘transient’ variables (iii) Length of output from layer 5 and (iv) The selection of activation function in the hidden MLP layer (i.e. layer 5). The selection of hyper-parameters is data-specific (i.e. specific to each load pattern), and as such, there is a need for a global method to optimize hyper-parameters for all load patterns.

Hyperopt is a python library for hyper-parameter optimization, which uses a Sequential Model-based Optimization (SMBO) method [37, 38]. Hyperopt is particularly suitable for optimizing hyper-parameters with respect to a loss function in graph-structured spaces. Hyperopt utilizes a Sequential model-based Optimization method that uses a surrogate function to approximate an expensive function $f(\theta)$, which in this case, is a neural network model. Thus, the optimization process becomes one of finding θ^* that maximizes the surrogate function, with respect to an expectation improvement criterion.

The expectation improvement criterion can be expressed as follows:

$$EI_{\gamma}(\theta) = \int_{-\infty}^{\infty} \max(\gamma - score, 0) p(score|\theta) dy = \int_{-\infty}^{\infty} \max(\gamma - score, 0) \frac{p(\theta|score)p(score)}{p(\theta)} dy \quad (23)$$

Thus the EI criterion penalizes the score (i.e. accuracy) of a model for exceeding some threshold γ and seeks for $score = f(\theta)$ to be lower than γ . In this analysis, a tree of parzen estimator (TPE) approach was used to model and optimize EI. In TPE approach, the quantity $p(\theta|score)$ is modeled using non-parametric probability distributions. In this analysis, the TPE algorithm in hyperopt is used as a black-box model to optimize the set of hyper-parameters, i.e. find the optimal structure for both models A and B. Further mathematical analysis on SMBO and the TPE approach is available in [37].

3.5 Missing value imputation scheme for PSB dataset

The electricity consumption data in practice can contain missing data of varying sequences [32], as is the case with load profile data obtained for the Public Safety Building, Salt Lake City, UT. While simpler methods such as linear interpolation are adequate to fill in such missing values, they are likely to produce erroneous results where the length of missing segment is high. The overall percentage of electricity consumption data missing in the PSB is low (i.e. less than 3% of the size of

the training set), however the number of consecutive timesteps for which the electricity consumption data was missing varied between 1 hour to 7 days. The electricity consumption data in residential buildings from [39] used in this analysis did not contain any missing data.

To address the missing value problem in the PSB electricity consumption dataset, it is suggested that the proposed modeling framework can be used to determine interpolated values where large segments of data are missing (i.e. when the number of consecutive time steps for which electricity consumption data is missing ≥ 5 hours). For a given ‘large’ segment, the interpolation scheme would use the RNN model to provide a weighted average of predictions based on (i) data prior to the missing segment and (ii) data after the missing segment. The proposed interpolation scheme is presented as follows:

$$\mathbf{y}_p = \frac{q_1}{q_1 + q_2} \mathbf{y}_{p_1} + \frac{q_2}{q_1 + q_2} \mathbf{y}_{p_2} \quad (24)$$

Here, p is the fraction of consecutive missing data points normalized with respect to the total number of training points, q_1 is the fraction of data points prior to the missing block, q_2 is the fraction of data points after the missing block, y_{p_1} is the prediction of energy consumption on p after the RNN model is trained on p_1 , and y_{p_2} is the prediction on p after the RNN model is trained on p_2 in reverse order. The authors hypothesize that at a given value of (q_1, q_2) , the deep RNN model performs better than 3-layer MLP beyond certain values of p . Details on how the imputation scheme is implemented is provided below:

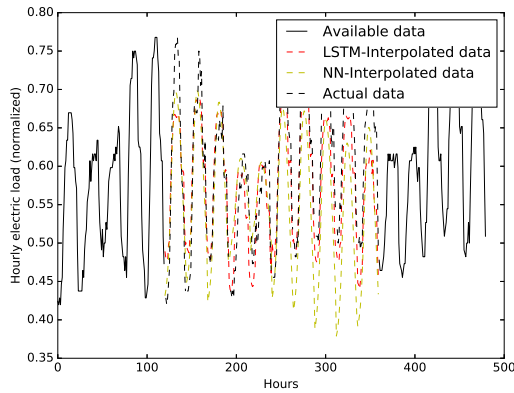


Figure 5. Results obtained for the HVAC Critical load profile using the LSTM interpolation scheme ($e_1 = 0.070$) and 3-layered MLP ($e_1 = 0.1211$) interpolation scheme between January 16 12:00 AM to January 25 11:59 PM

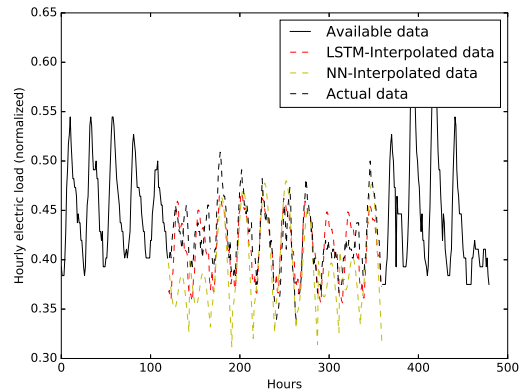


Figure 6. Results obtained for the HVAC Critical load profile using the LSTM interpolation scheme ($e_1 = 0.055$) and 3-layered MLP ($e_1 = 0.099$) between July 20 12:00 AM to July 29 11:59 PM

To evaluate the accuracy of the proposed scheme, the RNN-interpolation scheme was tested for energy consumption data (HVAC Critical Load) between January 16, 12:00 AM and January 25, 11:59 PM, corresponding to $p = 0.02$ and $q_1 = 0.042$ using model A to predict the missing segment. As we do not have access to missing data, we masked the available HVAC Critical profile data corresponding to the aforementioned 10-day period. Subsequently, the performance of the proposed scheme was compared with that of an imputation scheme using a 3-layered MLP. The

Algorithm 1 Proposed Imputation Scheme

- 1: **Identify** timesteps $\mathbf{r} = [r_1, r_2, r_3, \dots, r_N]$ during the training period for which the electricity consumption data is unavailable/missing.
 - 2: **Segregate** the timesteps \mathbf{r} into a small list $\mathbf{U} = [u_1, u_2, \dots, u_{N_1}]$ and a large list $\mathbf{V} = [v_1, v_2, \dots, v_{N_2}]$
 - 3: **for** $u \in \mathbf{U}$ **do**
 - 4: $y_p(u) = \text{linear interpolate}(u)$
 - 5: **end for**
 - 6: **Express** \mathbf{V} as $\mathbf{V} \in \mathbb{R}^{N' \times 2}$, where N' is the total number of large missing value segments. Thus $\mathbf{V}_i = [a_i, b_i]$, where a_i and b_i correspond to start and end timesteps for a ‘large’ missing segment i .
 - 7: **for** $[a, b] \in \mathbf{V}$ **do**
 - 8: **Define** $a' = 24 * \text{floor}(a/24)$
 - 9: **Define** $b' = 24 * \text{ceil}(b/24)$
 - 10: **Optimize** and **train** deep RNN model (either A or B) using training set $(\mathbf{x}_t, \mathbf{y}_t)$, where \mathbf{x}_t and \mathbf{y}_t are training input and targets prior to the missing segment, i.e. corresponding to timesteps $t = 0, 1, 2, \dots, a'$
 - 11: **Predict** using the trained deep RNN model $\mathbf{y}_{p1} = \text{predict}(\mathbf{x}_e)$ where \mathbf{x}_e is the test input corresponding to the missing segment
 - 12: **Optimize** and **train** deep RNN model using training set $(\mathbf{x}'_t, \mathbf{y}'_t)$, where \mathbf{x}'_t and \mathbf{y}'_t are training input and targets after to the missing segment in reverse order, i.e. corresponding to timesteps $t = T, T - 1, T - 2, \dots, b'$
 - 13: **Predict** using the trained deep RNN $\mathbf{y}_{p2} = \text{predict}(\mathbf{x}'_e)$ where \mathbf{x}'_e is the test input corresponding to the missing segment
 - 14: **Apply** the imputation scheme in equation to determine \mathbf{y}_p after reordering \mathbf{y}_{p2}
 - 15: **Replace** $y_{p,t=a' \dots a-1} = y_{a,t=a',a'+1,\dots,a}$ and $y_{p,t=b \dots b'-1} = y_{a,t=b,\dots,b-1}$ (Note: this step is necessary because electricity data is available within the range $a < t \leq a'$ and $b' < t \leq b$ but the deep RNN as presented can only predict in sequences of 24-hours.)
 - 16: **end for**
-

relative error (e_1) corresponding to the two schemes were 7.07% and 12.11% respectively. Figure 2 shows the corresponding results when the interpolation schemes were tested for HVAC critical load data between July 20, 12:00 AM and July 29, 11:59 PM. The relative errors for LSTM scheme and the 3-layer MLP scheme were 5.46% and 9.9% respectively.

4 Evaluation Setup

4.1 Dataset Characteristics

4.1.1 Salt Lake City Public Safety Building (PSB)

The proposed models A and B were tested on electricity consumption data obtained for the Public Safety Building in downtown Salt Lake City, UT. The Public Safety Building is a net-zero, LEED Platinum building with an area of 175,000 square feet. To evaluate the performance of the models, the data at one-hour intervals was partitioned into a training and a test set. As explained in section 3.1, the features consisted of a combination of weather variables, date-related variables and frequency-related variables. The weather variables were obtained from Mesowest [40], whereas the frequency-related variables were determined using an FFT analysis.

The training data corresponded to a time period between May 18, 2015, 12:00 AM and May 18, 2016 11:59 PM, and was at one-hour time resolution. The model was tested on data between 19 May 2016 12:00 AM to 8 August 2016 11:59 PM.

4.1.2 Aggregated Electricity Consumption in Residential Buildings

The overall hourly electricity consumption data in residential buildings in Austin, Texas at was obtained from Pecan Street Inc.’s Dataport web portal [39]. This electricity consumption data in residential buildings is then aggregated over an increasing increasing number of residential buildings, up to a maximum of thirty buildings. For a given number of buildings over which the residential building energy consumption was aggregated, the residential buildings were selected in ascending order of the building ID. Table 2 provides the details of the residential buildings considered for this analysis. The weather variables were obtained from Mesowest [40] and were measured at the weather station at East Austin RAWS.

The training data for the residential building case corresponded between 01 January 2015, 12:00 AM to 31 December 2015 11:59 PM, and the test data corresponded to a time period between January 1, 2016, 12:00 AM and December 31, 2016, 11:59 PM.

4.2 Implementation Setup

The models were developed in Python using the Keras API running on top of a Theano backend [35]. As mentioned in section 3.4, the hyper-parameters in the model are the number of units in the two LSTM layers and the hidden layer in the neural network. The search space for the hyper-parameter selection process (i.e the search space over which the algorithm in section 3.4 is applied) is provided in table 3. Other parameters (i.e. parameters other than hyper-parameters) are stated in table 4. The hyper-parameters were optimized specific to each load profile.

Table 2. List of Residential Buildings from free-share dataset in [39] selected for analysis. Source: Pecan Street Inc., Dataport [39]

Data index	Building ID	Building type
442	26	Single-Family Home
363	77	Single-Family Home
176	93	Single-Family Home
988	101	Single-Family Home
24	114	Single-Family Home
156	171	Single-Family Home
449	434	Single-Family Home
180	484	Single-Family Home
1274	503	Single-Family Home
27	585	Single-Family Home
108	624	Single-Family Home
29	744	Single-Family Home
566	781	Single-Family Home
638	821	Single-Family Home
183	871	Single-Family Home
152	890	Single-Family Home
390	946	Single-Family Home
1240	974	Single-Family Home
465	1086	Single-Family Home
290	1103	Single-Family Home
32	1192	Single-Family Home
293	1403	Town Home
33	1463	Single-Family Home
1058	1500	Apartment
456	1507	Single-Family Home
652	1629	Single-Family Home
356	1632	Single-Family Home
34	1642	Single-Family Home
792	1696	Single-Family Home

Table 3. Search Space for Hyper-parameter Selection

Length of output from Layer 2	[10, 100]
Length of output from Layer 3	[10, 100]
Length of output from Layer 5 (i.e. hidden MLP layer)	[5, 20]
Activation Function in Layer 5	[ReLU, Sigmoid]

Table 4. Miscellaneous implementation parameters used in this analysis.

Minimum number of epochs, μ_{min}	20
Maximum number of epochs, μ_{max}	200
Number of epochs for early-stopping, γ	10
Number of evaluations for hyper-parameter selection	40
Maximum number of epochs for hyper-parameter selection	20

Table 5. Relative Performances of model A, model B and multi-layered percptron in predicting different load profiles in Public Safety Building (PSB) in Salt Lake City, Utah.

Load Type	RMS (train), kWh	Model A			Model B			MLP		
		e_1 (%)	e_2 (%)	ρ	e_1 (%)	e_2 (%)	ρ	e_1 (%)	e_2 (%)	ρ
HVAC Power Critical	54.0	11.8	16.9	0.924	20.0	14.1	0.191	84.4	63.2	0.577
HVAC Power Normal	64.0	15.4	17.4	0.740	11.2	12.8	0.796	20.6	23.2	0.784
Convenience Power Critical	19.6	10.8	10.1	0.865	11.6	10.5	0.508	17.7	15.9	0.489
Convenience Power Normal	25.1	9.33	10.0	0.897	8.73	8.13	0.890	10.02	9.38	0.889
CRAC Power Critical	7.08	19.5	19	0.958	18.8	19.1	0.966	20.8	21.0	0.950
CRAC Power Normal	7.10	47.2	48.3	0.77	23.9	24.5	0.940	21.9	22.5	0.948
Elevator Power	1.01	46.8	45.8	0.765	46.8	45.8	0.765	48.30	47.3	0.740

4.3 Evaluation Metrics

The following metrics were used to evaluate the performance of the proposed deep RNN models with LSTM units:

- (1) Root mean squared error relative to RMS average of electricity consumption in test data, e_1

$$e_1 = \frac{\sqrt{\sum_{i=1}^T (y_{a,e} - y_p)^2}}{\sqrt{\sum_{i=1}^T y_{a,e}^2}} \quad (25)$$

- (2) Root mean squared error relative to RMS average of electricity consumption in training data, e_2 :

$$e_2 = \frac{\sqrt{\sum_{i=1}^T (y_{a,e} - y_p)^2}}{\sqrt{\sum_{i=1}^T y_{a,t}^2}} \quad (26)$$

- (3) Pearson Coefficient , ρ :

$$\rho(y_a, y_p) = \frac{\mathbb{E}[(y_a - \mu(y_a))(y_p - \mu(y_p))]}{\sigma_{y_a} \sigma_{y_p}} \quad (27)$$

Here, $y_{a,t}$ and $y_{a,e}$ denote the actual training and test values of electric consumption respectively.

5 Results and Discussion

5.1 Forecasting Salt Lake City PSB Electric Load Profiles

Table 5 illustrates how the performances of the proposed deep RNN models compare with those of the MLP model for multiple electric consumption profiles in the Salt Lake City Public Safety Building (PSB). The results show that in general, the proposed models perform better than the conventional multi-layered perceptron (MLP) in forecasting electricity consumption profiles over the 83-day time horizon. The table shows that both model A and model B perform comparatively better than MLP in predicting HVAC Critical and HVAC Normal load profiles - which are the two largest contributors to overall electricity consumption in the PSB building.

Figure 7 shows a comparison of predictions of HVAC Critical Load Profile made by the proposed RNN models and the MLP model between May 19, 2016 and August 8, 2016. Figure 8 focuses on a

5-day window of the aforementioned predictions between June 27, 2016 and July 2, 2016. The figures show that the proposed models A and B predict electricity consumption with a significantly higher accuracy than the MLP model. Table 5 shows that the improvement in accuracy is most noticeable in the HVAC Critical Load profile. This could be because the long-term changes associated with the HVAC Critical Load profile are more pronounced than other load profiles, and as mentioned in section 2.2, the proposed RNN models with LSTM activation functions can account for such long-term dependencies. The effect of long-term changes in electricity consumption on model accuracy will be explored quantitatively in Section 5.2.

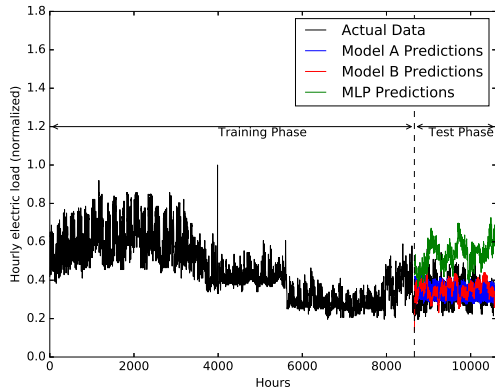


Figure 7. Predictions of HVAC Critical Load Profile by Deep RNN model ($e_2 = 14.0\%$) and MLP model ($e_2 = 63.2\%$ between May 18, 2016 and August 8, 2016). The root-mean squared (RMS) average of hourly load (in training) is 54.0 kW-h.

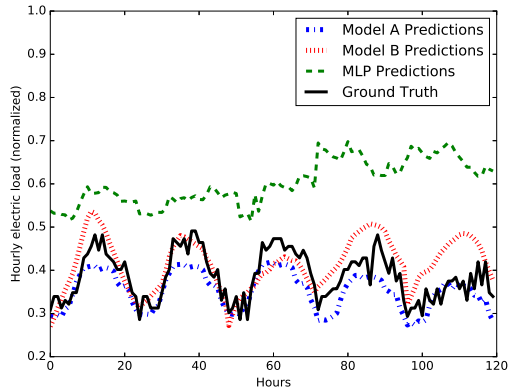


Figure 8. Predictions of HVAC Critical Load Profile by Deep RNN model and MLP model between June 27, 2016 and July 2, 2016.

Figures 9 and 10 show the corresponding plots for HVAC Normal load profile where model B, in particular, shows relative benefits compared to MLP. This study aims to generalize the application of the deep RNN model for multiple types of load profiles, and as such, the corresponding results for convenient normal load profile and computer-room air-conditioning (CRAC) load profiles as shown in figures 11-14. While the performance of the proposed model generalizes fairly well across multiple load profiles, as shown in table 5, its accuracy in predicting CRAC Normal load profile is comparatively poorer than that of the MLP model. This could be because the impact of long-term dependencies are likely to be less within the CRAC Normal profile. Furthermore, as figure 14 illustrates, the nature of CRAC profile illustrates that mistakes in predicting the CRAC Normal profile are likely to be more severely penalized. Table 5 and figures 7-14 also show that in general, model B performs marginally better than model A—possibly due to model B having fewer weight parameters to tune, and thus being less vulnerable to over-fitting. Thus, for the next section 5.2, where aggregated electricity consumption at a community scale is predicted, model B is used as the proposed RNN model to compare its performance with a 3-layer MLP.

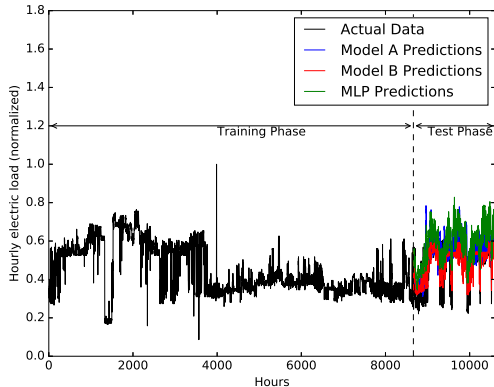


Figure 9. Predictions of HVAC Normal Load Profile by Deep RNN model ($e_2 = 11.4\%$) and MLP model ($e_2 = 23.2\%$ between May 18, 206 and August 8, 2016). The root-mean squared (RMS) average of hourly load (in training) is 64.0 kW-h.

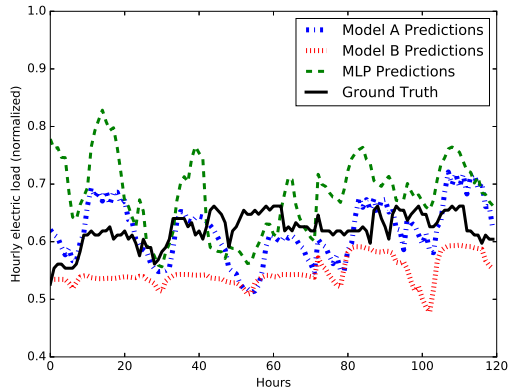


Figure 10. Predictions of HVAC Normal Load Profile by Deep RNN model and MLP model between June 27, 2016 and July 2, 2016

5.2 Forecasting Aggregate Electricity Consumption in Residential Buildings

The proposed RNN model B, which was shown to perform marginally better than model A in the load profiles presented in section 5.1, is next used to predict long-term (i.e. time horizon of 1-year) electricity consumption of aggregate load profiles for a group of residences in Austin, TX [41, 39], and its performance is compared with that obtained using the MLP model. Figure 15 shows how the predictions of model B compare with those of MLP for a single residential building in Austin, TX over a forecast period of one year. Figure 16 presents a segment of the predictions made by the models between January 20 and January 24; whereas figure 17 presents a segment of predictions over only five days between June 20 and June 24. The errors, e_2 , corresponding to the predictions of model B and MLP are very similar, at 45.3% and 46.1% respectively. Due to the stochastic nature of electricity consumption profile in a residential building, the errors in long-term forecasts for a given residential building are comparatively high compared to those obtained when predicting load profiles in a commercial building, such as those in the previous section for the PSB facility. The plots also show that the improvement in performance of model B is only marginal relative to that of MLP. Figures 18- 20 show the corresponding plots when the RNN model was applied on electricity consumption profile obtained on an aggregate of ten residential buildings. As electricity consumption profiles from multiple buildings are aggregated, the patterns in electricity consumption become more distinct, and the performance of both the RNN model and the MLP model improve.

Figure 23 shows how the relative error e_2 decays with increasing root mean squared (RMS) average of aggregate hourly electricity consumption (E_{avg}), which corresponds to the number of residential buildings (n) over which the electric consumption profiles are aggregated. It is observed that in general, model B has only marginal advantage over the MLP in predicting electricity consumption when $E_{avg} \leq 14.9$ kWh, i.e. when $n \leq 10$. With increasing n , the MLP performs significantly better than the deep RNN model. Thus, we do not observe the benefits in performance

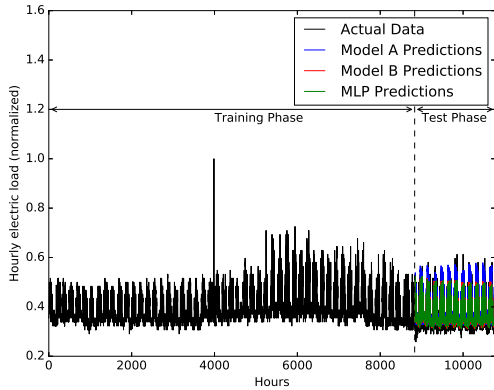


Figure 11. Predictions of Convenient Normal Load Profile by Deep RNN model ($e_2 = 8.13\%$) and MLP model ($e_2 = 9.38\%$ between May 18, 206 and August 8, 2016. The root-mean squared (RMS) average of hourly load (in training) is 19.6 kW-h

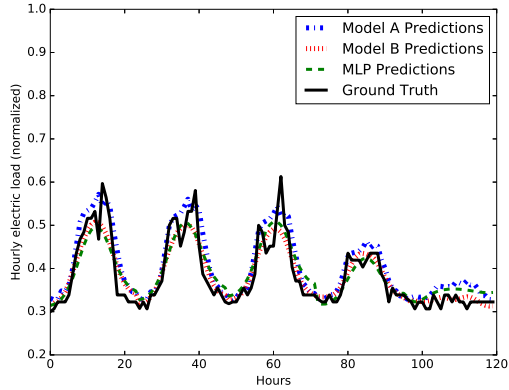


Figure 12. Predictions of Convenient Normal Load Profile by Deep RNN model and MLP model between June 27, 2016 and July 2, 2016

in case of residential building consumption aggregate profiles, as we saw in the commercial building load profiles. This may be attributed to the following: (i) Unlike commercial building load profiles, the aggregate profiles are less likely to depend on regular transient schedules, for which the deep RNN model was generating surrogates, and (ii) The aggregate profiles are less likely to experience long-term dependencies, and more likely to contain noisy data, such that an RNN model using LSTM activation functions are likely to over-fit these aggregate profiles. Nonetheless, the results presented in figure 23 can serve as benchmark accuracies for one-year ahead predictions at one-hour resolution for the Pecan Street dataset in [39].

Figure 21 shows that the relative errors e_1 of both the RNN and MLP models corresponding to winter months (such as January) are considerably higher than those corresponding to summer months. We can look at the disaggregated energy consumption data (available in Pecan Street Inc. Dataport [39]) for individual appliances to explain why the relative error is higher in a winter month such as January compared to that in June. During June, bulk of the electricity consumption is contributed by end uses associated with HVAC units such as the air compressor, which is likely to be strongly dependent on weather conditions (i.e. dry-bulb temperature and humidity). During January, the HVAC-related end uses contribute to a significantly lower fraction of the electricity consumption, and the bulk of the electricity consumption in the building is contributed by other appliances such as electric car charger that contribute to a comparatively more noisy and discontinuous profile. Figure 22 shows that although e_1 is higher for the winter months, error e_2 is actually greater for summer months. This is because e_1 uses the RMS average of the actual hourly electricity consumption during the test phase, and the overall electricity consumption is lower during the winter months [39].

To quantitatively evaluate the effect of such long-term dependencies (i.e. changes in electricity consumption patterns that occur over a timescale longer than a day), we define a parameter s to quantify the discrepancy between the test data and the corresponding training data. s is expressed as:

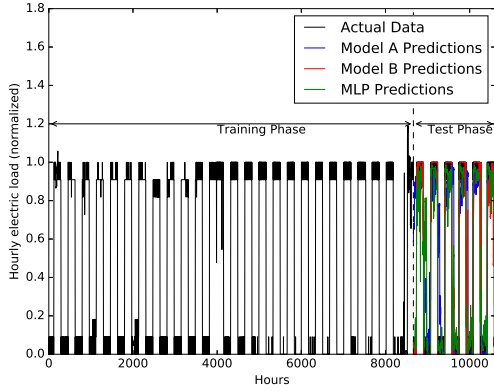


Figure 13. Predictions of CRAC Normal Load Profile by Deep RNN model ($e_2 = 24.5\%$) and MLP model ($e_2 = 22.5\%$ between May 18, 206 and August 8, 2016. The root-mean squared (RMS) average of hourly load (in training) is 7.10 kW-h

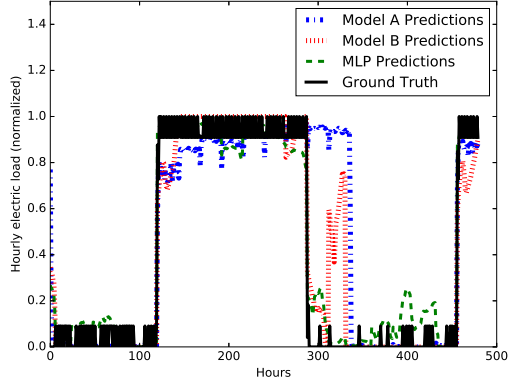


Figure 14. Predictions of CRAC Normal Load Profile by Deep RNN model and MLP model between June 27, 2016 and July 2, 2016

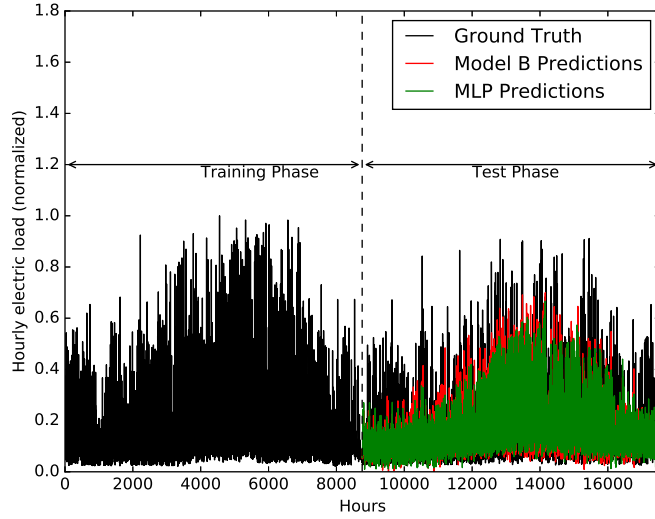


Figure 15. Predictions of hourly electricity consumption profile in a residential building (building ID: 26) by Model B ($e_2 = 45.35\%$) and MLP model ($e_2 = 46.1\%$ between Jan 01, 2016 to December 31, 2016. Source: Pecan Street Inc., Dataport [39]

$$s = \frac{\sqrt{\sum_{i=1}^T (y_{a,e} - y'_t)^2}}{y_{a,e}^{max} - y_{a,e}^{min}} \quad (28)$$

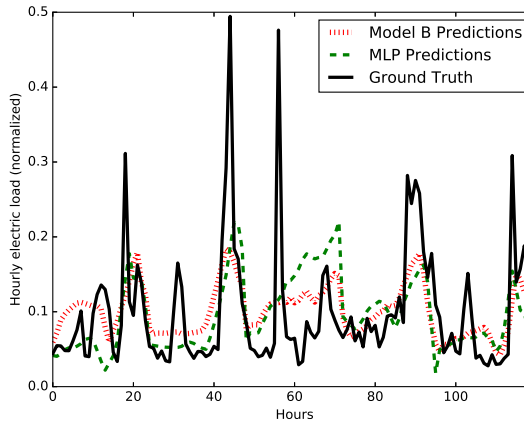


Figure 16. Predictions of hourly electricity consumption profile in a residential building (building ID: 26) by Model B and MLP model between January 20, 2016 and January 24, 2016. Source: Pecan Street Inc., Dataport [39]

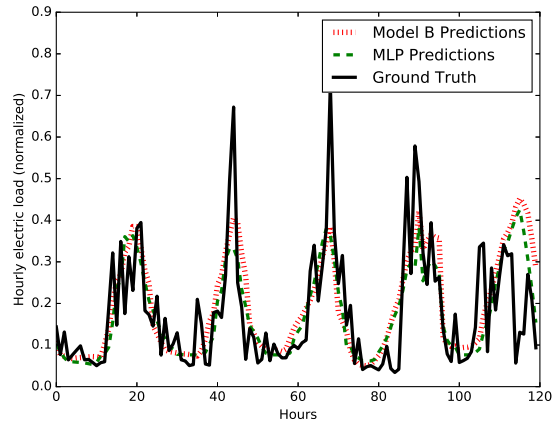


Figure 17. Predictions of hourly electricity consumption profile in a residential building (building ID: 26) by Model B and MLP model between June 20, 2016 and June 24, 2016. Source: Pecan Street Inc., Dataport [39]

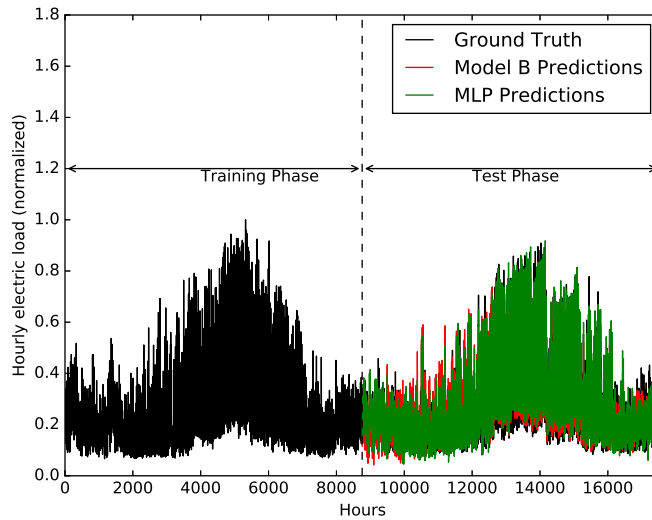


Figure 18. Predictions of aggregated hourly electricity consumption profile residential buildings ($n = 10$) by model B ($e_2 = 21.9\%$) and MLP model ($e_2 = 22.7\%$) between Jan 01, 2016 and December 31, 2016. Source: Pecan Street Inc., Dataport [39]

Here $y_{a,e}$ is the actual hourly electricity consumption in the test period, $y_{a,e}^{max}$ is the maximum value in $y_{a,e}$, $y_{a,e}^{min}$ is the minimum value in $y_{a,e}$, and y'_t is the corresponding hourly electricity consumption in the training data within the same period as y'_t . To give an example, in the case of

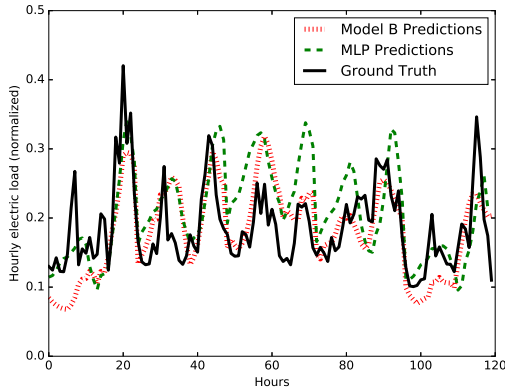


Figure 19. Predictions of aggregated hourly electricity consumption profile in residential buildings ($n = 10$) by model B and MLP model between January 20, 2016 to January 24, 2016. Source: Pecan Street Inc., Dataport [39]

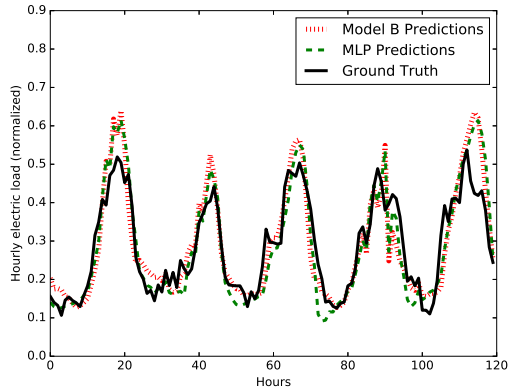


Figure 20. Predictions of aggregated hourly electricity consumption profile in residential buildings ($n = 10$) by model B and MLP model between June 20, 2016 to June 24, 2016. Source: Pecan Street Inc., Dataport [39]

PSB in SLC, where the forecast is made for the period between May 19, 2016 to August 8, 2016, y'_t is the hourly electricity consumption between May 19, 2015 to August 8, 2015. Thus the parameter s indicates a measure of long-term changes within the electricity consumption profile that might cause discrepancy between the actual test data and the corresponding training data. Figure 24 shows how the relative error e_2 in one-year ahead predictions made by the deep RNN and the MLP model varies with s - including both the commercial and residential building cases. It is observed that when the discrepancy s between the test and corresponding training data is high, (indicating that the electricity consumption profiles have long-term dependencies) the RNN model performs comparatively better than the MLP. The outliers to this observation are the CRAC load profiles, which are distinctly periodic functions with characteristic timescales that are not 24 hours.

6 Further Discussion and Limitations

In this paper, we proposed two recurrent neural network (RNN) models to forecast electricity consumption profiles in commercial buildings and aggregate electricity consumption in residential buildings. The predictions were made in sequences of 24-hours at one-hour resolution over a medium-to-long term time horizon (> 1 week). Overall, the neural network models presented in this analysis perform well in forecasting electricity consumption over medium-to-long term time horizon. However, the presented models are subject to the following limitations:

- The models assume knowledge of future weather data, and do not account for the uncertainty in weather over medium-to-long term time horizons.
- The proposed models are able to predict future electricity consumption for a given building in a medium-to-long term time horizon, after being trained on past data specific to that building. This means that a model trained on a specific building will likely produce erroneous results

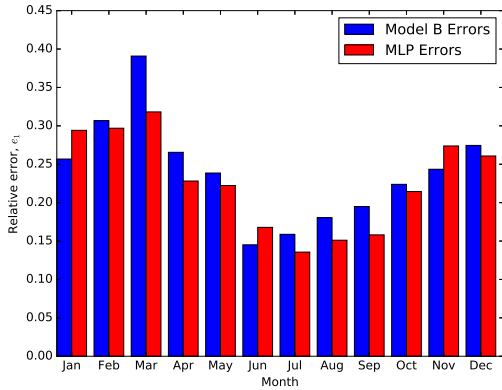


Figure 21. Relative error e_1 of deep RNN and MLP models in predicting aggregated electricity consumption ($n = 10$) during different months

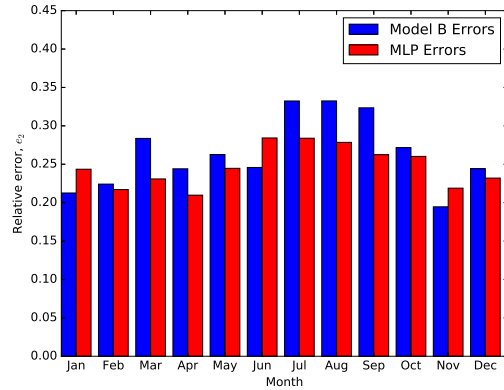


Figure 22. Relative error e_2 of deep RNN and MLP models in predicting aggregated electricity consumption ($n = 10$) during different months

when forecasting electricity consumption for a separate building, or even for the same building if significant changes have been made to its structure, equipment, occupancy or operations.

- The models will perform differently when data is aggregated, or when long-term dependencies are difficult to identify for any reason. We hypothesize that the comparatively poorer performance of the RNN model in predicting aggregate electricity consumption in buildings is due to the fact that the aggregate profiles have fewer long-term dependencies.

This means that these models would likely have decreased accuracy in application if the weather in the future was significantly different from the weather that was concurrent with the training data. Likewise, if the equipment or operational scheme for the building changes significantly, we expect a decrease in accuracy even when applied to the same building.

To improve future use of RNN models in making electricity use forecasts, we recommend the development of a deep learning framework that can quantify the uncertainty associated with these predictions. Further study using the proposed models on a larger data set of buildings would be invaluable for providing detailed guidelines on neural network model selection.

7 Conclusion

The following conclusions result from this analysis:

- The proposed RNN models A and B, in general, perform better than a 3-layer multi-layered perceptron model in the case of electric load profiles in commercial buildings. The proposed model is able to provide surrogates for unknown transient variables that can affect load profiles in commercial buildings, and can account for long-term dependencies in electricity consumption.
- The proposed RNN model B does not perform as well in forecasting aggregate load profiles over a 1-year time horizon compared to a 3-layer multi-layered perceptron (MLP) model.

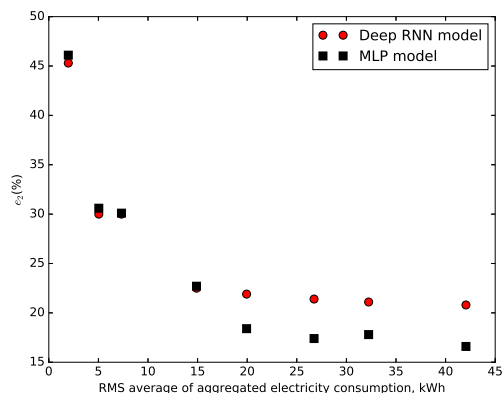


Figure 23. Relative errors of the RNN and MLP model as a function of the root-mean squared average of hourly aggregate consumption

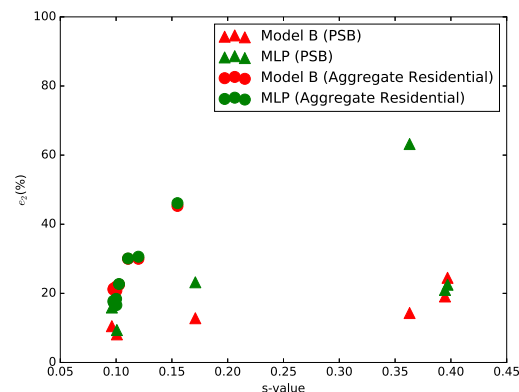


Figure 24. Relative errors of RNN and MLP model as a function of parameter s .

The MLP performs comparatively better than the RNN model as the number of buildings increases over which electricity consumption is aggregated.

- The proposed imputation scheme using the RNN model to provide missing values in time series energy consumption data may be effectively used to replace these data points. The missing-value imputation scheme has been shown to obtain higher accuracies than those obtained using a MLP model.

This work indicates that deep RNN models have significant potential for use in predicting building energy consumption.

8 Acknowledgments

The authors gratefully acknowledge the government of Salt Lake City, UT for their cooperation and assistance in obtaining data used in this work. We also wish to thank Pecan Street Inc. for providing academic licenses to their data library.

This material is based upon work supported by the National Science Foundation under the following Grant: CBET 1512740.

References

- [1] Building energy databook, U.S. Department of Energy, Last Accessed: 2017-11-23. URL <https://openei.org/doe-opendata/dataset/buildings-energy-data-book>
- [2] M. E. El-hawary, The Smart Grid State-of-the-art and Future Trends, Electric Power Components and Systems 42 (3-4) (2014) 239–250. doi:10.1080/15325008.2013.868558.

- [3] IEC smart grid standardization roadmap, International Electrotechnical Commission (IEC), Last Accessed: 2017-11-23.
URL <http://www.iec.ch/smartgrid>
- [4] E. Mocanu, P. H. Nguyen, W. L. Kling, M. Gibescu, Unsupervised energy prediction in a Smart Grid context using reinforcement cross-building transfer learning, *Energy and Buildings* 116 (2016) 646–655.
- [5] E. Mocanu, P. H. Nguyen, M. Gibescu, W. L. Kling, Deep learning for estimating building energy consumption, *Sustainable Energy, Grids and Networks* 6 (2016) 91–99.
- [6] L. Friedrich, A. Afshari, Short-term Forecasting of the Abu Dhabi Electricity Load Using Multiple Weather Variables, *Energy Procedia* 75 (2015) 3014–3026. doi:10.1016/j.egypro.2015.07.616.
- [7] A. Dedinec, S. Filiposka, A. Dedinec, L. Kocarev, Deep belief network based electricity load forecasting: An analysis of macedonian case, *Energy*.
- [8] E. A. Bakirtzis, C. K. Simoglou, P. N. Biskas, D. P. Labridis, A. G. Bakirtzis, Comparison of advanced power system operations models for large-scale renewable integration, *Electric Power Systems Research* 128 (2015) 90–99.
- [9] D. Kolokotsa, The role of smart grids in the building sector, *Energy and Buildings* 116 (2016) 703–708. doi:10.1016/j.enbuild.2015.12.033.
- [10] Y. Goude, R. Nedellec, N. Kong, Local short and middle term electricity load forecasting with semi-parametric additive models, *IEEE transactions on smart grid* 5 (1) (2014) 440–446.
- [11] H.-x. Zhao, F. Magouls, A review on the prediction of building energy consumption, *Renewable and Sustainable Energy Reviews* 16 (6) (2012) 3586–3592.
- [12] A. Foucquier, S. Robert, F. Suard, L. Stphan, A. Jay, State of the art in building modelling and energy performances prediction: A review, *Renewable and Sustainable Energy Reviews* 23 (2013) 272–288.
- [13] US Department of Energy, *EnergyPlus Documentation* (2013).
- [14] H. S. Rallapalli, A comparison of energyplus and equest whole building energy simulation results for a medium sized office building, Ph.D. thesis, Arizona State University (2010).
- [15] N. Fumo, M. Rafe Biswas, Regression analysis for prediction of residential energy consumption, *Renewable and Sustainable Energy Reviews* 47 (2015) 332–343.
- [16] C. Robinson, B. Dilkina, J. Hubbs, W. Zhang, S. Guhathakurta, M. A. Brown, R. M. Pendyala, Machine learning approaches for estimating commercial building energy consumption, *Applied Energy* 208 (Supplement C) (2017) 889 – 904.
- [17] S. Daz, J. A. Carta, J. M. Matas, Performance assessment of five mcp models proposed for the estimation of long-term wind turbine power outputs at a target site using three machine learning techniques, *Applied Energy*.

- [18] Y. Heo, V. M. Zavala, Gaussian process modeling for measurement and verification of building energy savings, *Energy and Buildings* 53 (2012) 7–18.
- [19] D. C. Park, M. A. El-Sharkawi, R. J. Marks, L. E. Atlas, M. J. Damborg, others, Electric load forecasting using an artificial neural network, *Power Systems, IEEE Transactions on* 6 (2) (1991) 442–449.
- [20] P. A. Gonzalez, J. M. Zamarreo, Prediction of hourly energy consumption in buildings based on a feedback artificial neural network, *Energy and Buildings* 37 (6) (2005) 595–601.
- [21] X. L. Lu, T. Lu, C. J. Kibert, M. Viljanen, Modeling and forecasting energy consumption for heterogeneous buildings using a physicalstatistical approach, *Applied Energy* 144 (2015) 261–275.
- [22] W. Charytoniuk, M. S. Chen, P. Van Olinda, Nonparametric regression based short-term load forecasting, *IEEE Transactions on Power Systems* 13 (3) (1998) 725–730.
- [23] K. Yun, R. Luck, P. J. Mago, H. Cho, Building hourly thermal load prediction using an indexed ARX model, *Energy and Buildings* 54 (2012) 225–233.
- [24] Y. B. I. Goodfellow, A. Courville, Deep learning, book in preparation for MIT Press, <http://www.deeplearningbook.org>, Last Accessed = 2017-11-23 (2016).
- [25] M. R. Arahall, A. Cepeda, E. F. Camacho, Input variable selection for forecasting models, *IFAC Proceedings Volumes* 35 (1) (2002) 463–468.
- [26] C. Fan, F. Xiao, Y. Zhao, A short-term building cooling load prediction method using deep learning algorithms, *Applied Energy* 195 (Supplement C) (2017) 222 – 233.
- [27] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Computation* 9 (1997) 1735–1780.
- [28] Z. C. Lipton, A critical review of recurrent neural networks for sequence learning, *CoRR* abs/1506.00019, <http://arxiv.org/abs/1506.00019>, Last Accessed = 2017-11-23.
- [29] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078. URL <http://arxiv.org/abs/1406.1078>
- [30] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: *Advances in neural information processing systems*, 2014, pp. 3104–3112.
- [31] M. A. Zaytar, C. El Amrani, Sequence to sequence weather forecasting with long-short term memory recurrent neural networks, *Internal Journal of Computer Applications* 143 (2997) 7–11.
- [32] A. Kavousian, R. Rajagopal, M. Fischer, Determinants of residential electricity consumption: Using smart meter data to examine the effect of climate, building characteristics, appliance stock, and occupants’ behavior, *Energy* 55 (2013) 184–194.
- [33] D. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

- [34] J. Horel, M. Splitt, L. Dunn, J. Pechmann, B. White, C. Ciliberti, S. Lazarus, J. Slemmer, D. Zaff, J. Burks, Mesowest: Cooperative mesonets in the Western United States, *Bulletin of the American Meteorological Society* 83 (2) (2002) 211–225.
- [35] F. Chollet, et al., Keras, <https://github.com/fchollet/keras>, Last Accessed = 2017-11-23 (2015).
- [36] A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: *Acoustics, speech and signal processing (icassp), 2013 IEEE international conference on*, IEEE, 2013, pp. 6645–6649.
- [37] J. Bergstra, D. Yamins, D. D. Cox, Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms, in: *Proceedings of the 12th Python in Science Conference*, 2013, pp. 13–20.
- [38] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, D. D. Cox, Hyperopt: a python library for model selection and hyperparameter optimization, *Computational Science & Discovery* 8 (1) (2015) 014008.
- [39] Dataport from pecan street, last Accessed = 2017-11-23.
URL <https://dataport.cloud/>
- [40] Mesowest, mesowest.utah.edu, Department of Atmospheric Sciences, University of Utah, Last Accessed = 2017-11-23.
- [41] J. D. Rhodes, C. R. Upshaw, C. B. Harris, C. M. Meehan, D. A. Walling, P. A. Navrátil, A. L. Beck, K. Nagasawa, R. L. Fares, W. J. Cole, et al., Experimental and data collection methods for a large-scale smart grid deployment: Methods and first results, *Energy* 65 (2014) 462–471.