MATH 509 Project Report

Alexander Jordan (ID: 1577190)

Bowen Yang (ID: 1606931)

Real-Time and Future Building Energy Prediction Using Machine Learning Techniques

1. Introduction

1.1 Background

Building sectors account for 30%-40% of global energy consumption, contributing approximately 19% of greenhouse gas emissions [1]. Heating, ventilation, and air conditioning (HVAC), domestic hot water, lighting, and appliances are the primary services that consume energy in buildings. Globally, HVAC systems account for approximately 40% of total energy consumption [2], and a growing population and rapid urbanization are responsible for the increase in energy consumption. One of the essential parts of building energy management is energy consumption prediction, which could inspire energy policy to reduce energy consumption [3]. Furthermore, daily energy management relies on the energy demand forecast to control the appropriate energy-related equipment. For example, predicted model can be fed to a control system to pre-heat or pre-cool a building before tenants arrives or to storage generated electricity from solar panels if it predicts a peak in electrical usage. Thus, the energy prediction is a key to enable into smart and sustainable buildings design for both new and existing buildings. Much efforts have been made to accurately predict building energy consumption and design an optimal energy efficiency control system to reduce greenhouse gas emissions by managing energy emissions and conserving energy in buildings.

1.2 Problem Description

Numerous variables, such as weather, time, building occupancy, etc., make the development of a credible energy prediction model challenging. In building energy prediction systems, machine learning (ML) models, such as tree algorithms, support vector machine (SVM), random forest (RF), and artificial neural networks (ANN), which can describe the relationship between model input and output without requiring complex domain knowledge, have become increasingly important.

However, the aforementioned traditional ML approaches have limited ability to cope with time-series data for predicting future energy usage (i.e., next hour, next day and next month). Even though recurrent neural networks (RNN) are effective at handling sequentially data, they are unable to capture long-term dependencies in sequential data [4]. As a variation of the RNN, long short-term memory (LSTM) models may learn the long-term dependent information to produce accurate predictions.

1.3 Question to be Addressed

To address the concerns that mentioned above, the following research questions are proposed:

- Are the data from a single year sufficient to estimate real-time and future energy consumption?
- Which machine learning technique is optimal in terms of prediction accuracy and computing cost for predicting energy usage in a smart home?

1.4 Data Description and Visualization

The main goal of this project is to estimate the real-time and future total energy consumption of a smart building using ML approaches.

The dataset (https://www.kaggle.com/code/malekzadeh/smart-home-data-processing-weather-vs-energy/data) is fetched from Kaggle. The data collection period for a smart home was from 2016-01-01 to 2016-12-16. Nineteen features, including solar generation, appliances (dishwasher, furnace, home office, refrigerator, garage door, kitchen, barn, microwave, and living room), and weather data (temperature, humidity, visibility, pressure, wind speed, and dew point) are extracted to predict real-time and future (next hour) total energy consumption. The information on the factors that were measured is presented in Table 1. Both Fig. 1 and 2 illustrate the total amount of energy that is consumed on a daily and hourly basis, respectively.

Table 1 Information of variables.

Categories	Variables	Type	The minimum	The maximum	Unit
Energy generation	Generation	Numerical	0	613.88	W
Energy consumption	Dishwasher	Numerical	0	1401.77	W
	Furnace	Numerical	0.33	2472.63	W
	Home office	Numerical	0.08	971.75	W

	Fridge	Numerical	0.06	851.27	W
	Wine cellar	Numerical	0.02	1273.94	W
	Garage door	Numerical	0.02	1088.93	W
	Kitchen	Numerical	0	2265.87	W
	Barn	Numerical	0	7027.90	W
	Well	Numerical	0	1633.02	W
	Microwave	Numerical	0	1929.80	W
	Living room	Numerical	0	465.22	W
Weather condition	Temperature	Numerical	-12.64	93.72	°F
	Humidity	Numerical	0.13	0.98	%
	Visibility	Numerical	0.27	10	Kilometers
	Apparent temperature	Numerical	-32.08	101.12	°F
	Pressure	Numerical	986.4	1042.46	Millibar
	Windspeed	Numerical	0	22.91	Km/h
	Dewpoint	Numerical	-27.24	75.49	°F
	House overall	Numerical	0	14714.57	W

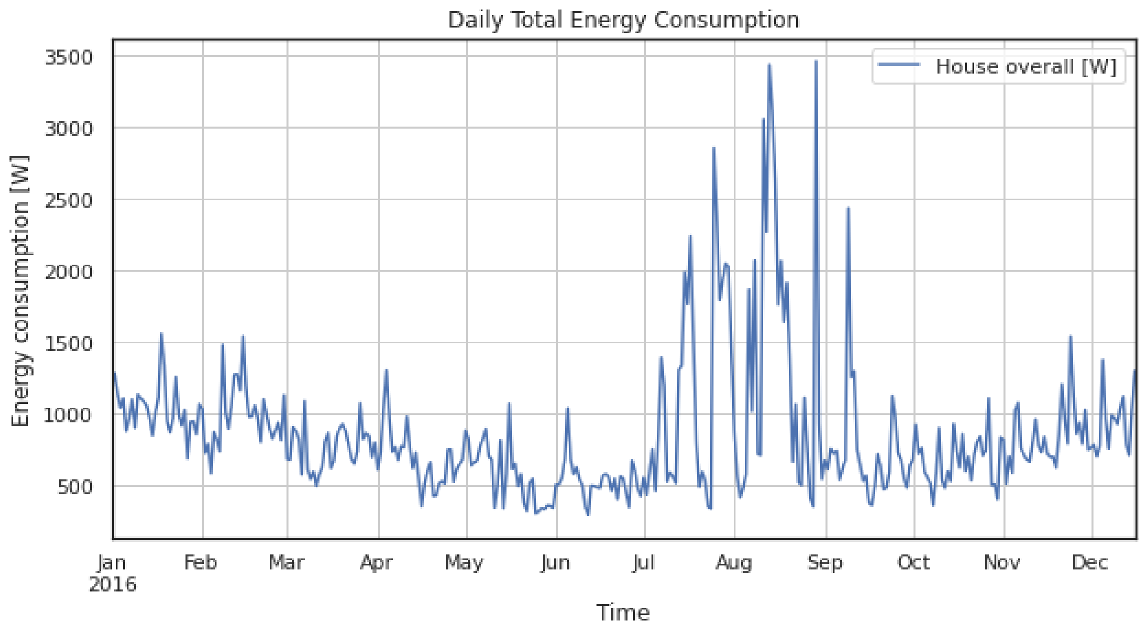


Fig. 1. Daily total energy consumption.

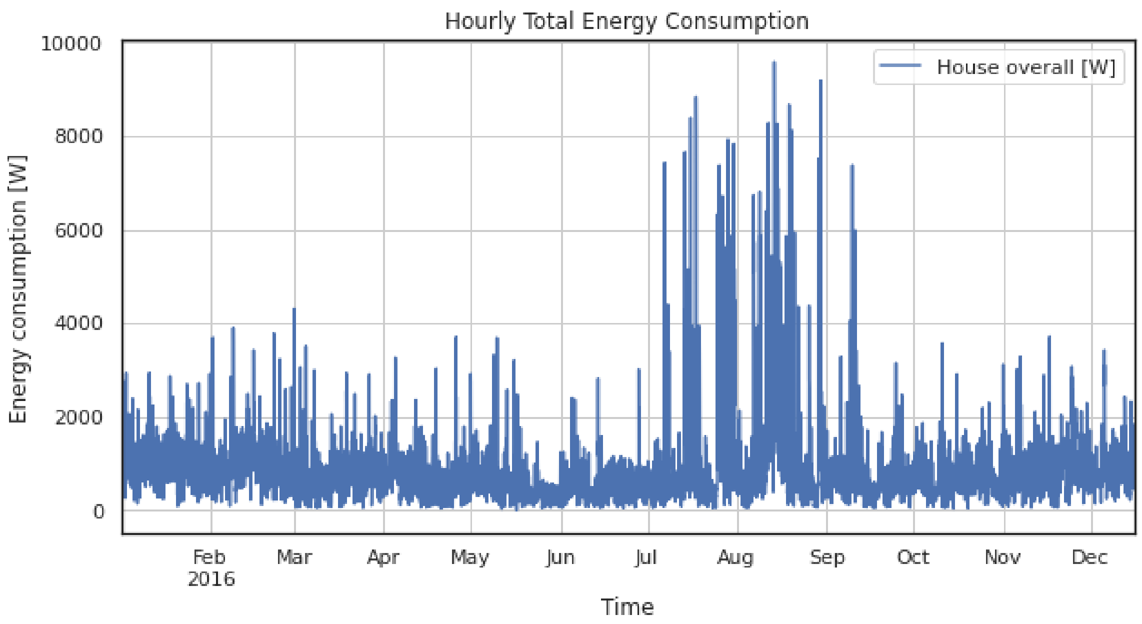


Fig. 2. Hourly total energy consumption.

2 Formulation of the Mathematical Model

2.1 LSTM Model

LSTM is specifically designed to prevent the problem of long-term dependency. The cell state and gate structure are the core concept of LSTM. Fig. 3 (a), (b), (c), and (d) depict the calculation process of the forget gate, input gate, cell state, and output gate, respectively. Using the gradient descent approach, the initial values of the weight matrices (i.e., W_f, W_i, W_C, W_o) and bias (i.e., b_f, b_i, b_C, b_o) are updated. The information from the current input x_t and previous hidden state h_{t-1} is transmitted through the sigmoid function (σ), and forget gate determines whether the gate should retain the information or not, according to the Eq. (1). The input gate is utilized to update the cell status, which includes two tasks. First, the precious cell output and current state are supplied to a classifier sigmoid function to determine what relevant information needs be updated, as shown in Eq. (2). Second, the previous cell output and current input are again sent to the activation tangent function $(tanh)$ in order to generate a new candidate value $\left(\widetilde{C}_t\right)$, Eq. (3). The cell state (C_t) is the path of information transmission, so the information can be passed in a serial connection, Eq. (4). Finally, the current output vector h_t is determined by linear transformation through sigmoid function of previous cell state and current input vector and passes the output through tangent function, and LSTM is used to predict the next-hour building energy consumption at $T + 1$ time, as shown in Eq. (5) and (6).

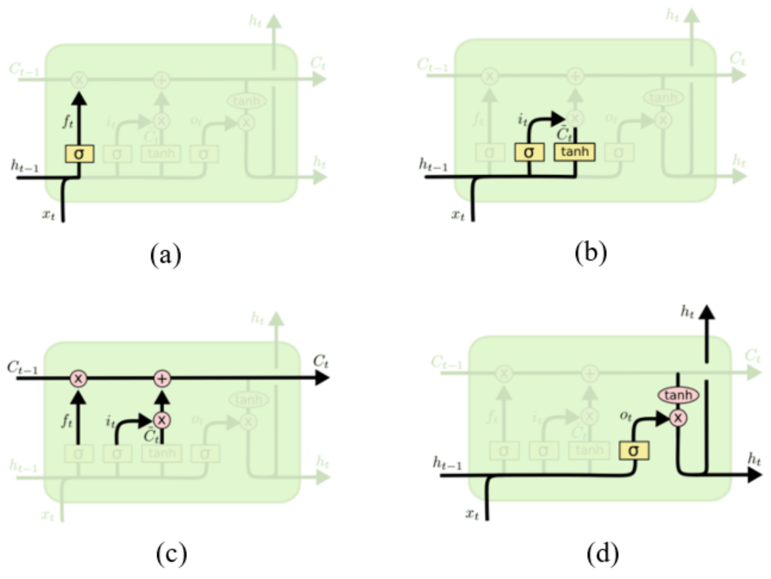


Fig. 3. Procedure of forward propagation of LSTM.

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

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\widetilde{C_t} = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh{(C_t)} \tag{6}$$

2.2 GRU Model

Gated recurrent unit (GRU) is another variant model to overcome traditional RNN problems for the time-series analysis. The three gates of the LSTM are reduced to two: reset gate and update gate. The reset gate determines how much past information we wish to keep, whereas the update gate determines how much new states are copied from the old state. Fig. 4 illustrates a typical structure of GRU, the calculation process is defined in Eq. (7), Eq. (8) and Eq. (9).

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{7}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{8}$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot tanh(W \cdot [r_t \cdot h_{t-1}, x_t]) \tag{9}$$

where x_t is the input at time step t, h_t, and h_{t-1} are the hidden states at the time step t and $t-1$, respectively. r_t is the output of the reset gate at time step t. and W, W_z, and W_r are the weight matrixes.

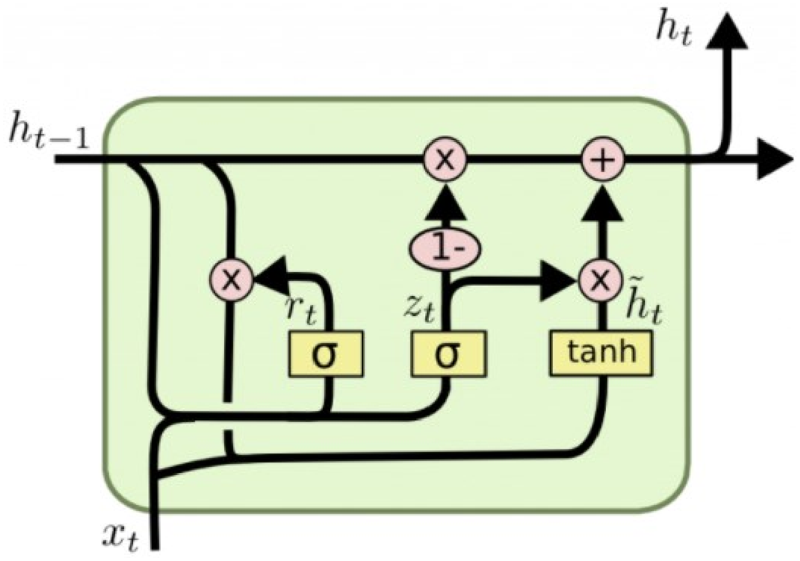


Fig. 4. Basic architecture of a GRU cell.

2.3 RNN Model

RNN is a class of ANN with a loop that repeats the same task for each element in a sequence [5], and has a memory for capturing information, whereas traditional neural networks presume that all inputs are independent of one another and hence lack the ability to process sequential data. Fig. 5 shows the general structure of estimating the building's energy using the RNN h_t, and h_{t-1} are the hidden states at the time step t and $t-1$, respectively. U and W are the weight matrix, as shown in Eq. (10).

$$h_t = tanh(Ux_z + Wh_{t-1}) \tag{10}$$

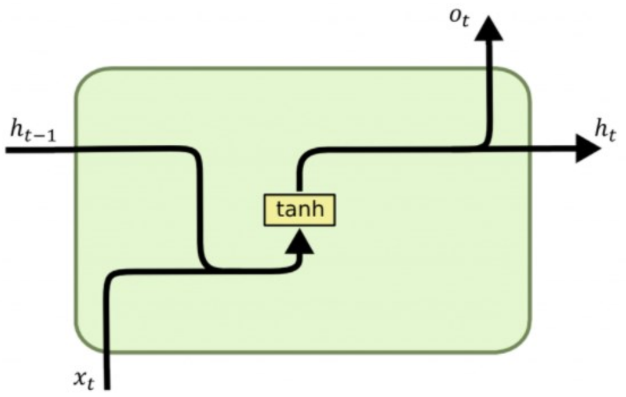


Fig. 5. A recurrent neural network.

2.4 Traditional Artificial Neural Networks

ANN has been developed to imitate human nervous system and used into mathematical models. In its simplest form, as demonstrated in Fig.6 x_1, x_2 through x_n represent the input used to train and test the ANN model, along with their corresponding weights w_1, w_2 through w_n, bias b and activation function f (described in section 2.5) applied to the weighted sum of the inputs to compute the output y.

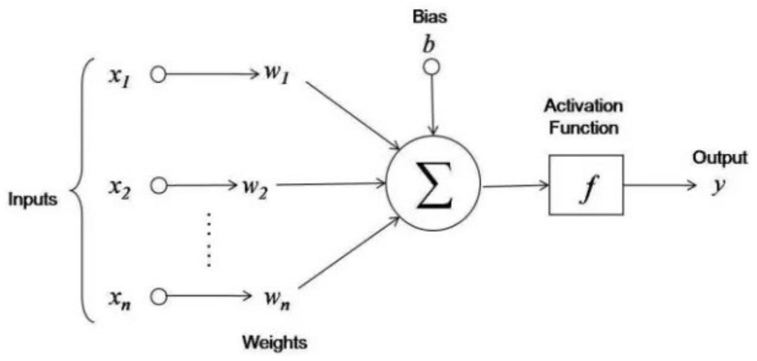


Fig. 6. Example of a neural network.

$$y = f(wx + b) \tag{11}$$

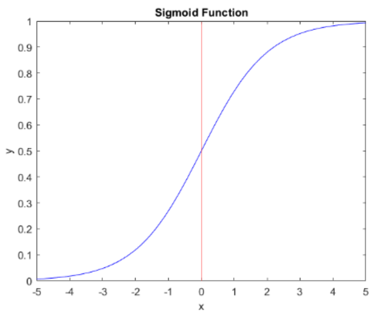
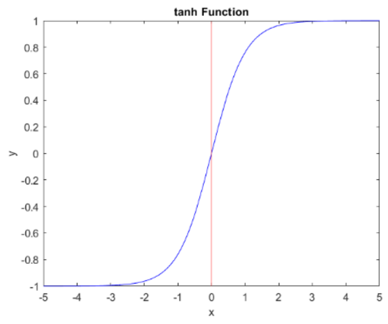
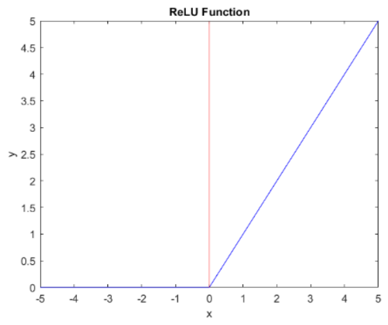
Table 2 Pros and cons of models.

Model	Advantages	Disadvantages
ANN	• Can be applied to complex non-linear problems. • Typically relatively simple models (for small hidden layers).	• Fails to model sequential data. • Typically computationally efficient compared to Deep Neural Networks.
RNN	• Can handle time-series data. • Low computational cost. • Captures short term dependencies very well (e.g. voice recognition).	• Issue of gradient vanishing. • Cannot capture ling-term dependencies.
GRU	• Handles time-series data. • Can maintain a long-term dependency.	• Slow convergence. • Low learning efficiency. • Computationally expensive.
LSTM	• Fewer issues regarding vanish / exploding gradient issue. • Handles time-series data. • Can learn long-term dependencies.	• Long training time. • Computationally expensive.

2.5 Activation Functions

In neural networks, the output from the summing function layer is then passed through an activation layer which helps define the important and output of each neuron in the hidden layer. As such, the activation layer is responsible for controlling the output of each neuron to the output layer of the system, and as such plays an important role in the training and performance of modeled systems. There are three commonly used activation functions that were considered for this project described below in Table 3.

Table 3 Common activation functions in machine learning.

Activation Function	Equation (output as z)	Graph
Sigmoid	$z(x) = \frac{1}{1+\ e^{-x}}$	
Tanh	$z(x) = \frac{e^{2x}-1}{e^{2x}+1}$	
Rectified Linear Unit (ReLU)	$z(x) = \max(0, x)$	

Different activation functions were tested for each model, as choosing a correct activation function for a specific problem is not trivial. Different activation functions can lead to different training and testing results, so comprehensively testing different activation functions is important to having a well performing model.

2.6 Performance Criteria

To evaluate the ML models, R square (R^2) and Mean Absolute Percentage Error (MAPE) were used in this study. The formula for R^2 and MAPE are:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y_i}}{y_i}\right| \tag{12}$$

where y_i is the actual value of the data point, and $\hat{y_i}$ is the predicted value of the data point.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y_i})^2} \tag{13}$$

where y_i represents the actual values, $\hat{y_i}$ represents the prediction, and $\bar{y_i}$ represents the mean of all the values.

3 Solution of the Problem

3.1 Models Used and Expected Result

In this study, multiple different model types were used for predicting both future and real-time data. For future based data, exclusively sequential neural network types were used with data split 80% training and 20% testing as per machine learning standard. Since RNN models do not have the ability to track long term dependencies, this can either have a good or bad affect for the task of predicting residential building consumption. Without the ability for long term dependency, an RNN model is expected to perform well with slow gradual weather changes, however with large weather condition changes (large shift in temperature or wind speed/direction) the model will not have past data to base a prediction on, which could lead to inaccuracies. LSTM and GRU models are the opposite where they should be able to train to meet most states of the variables and be more stable at predicting future energy consumption.

For real-time energy prediction, the ANN model and sequential models are designed differently, and therefore are expected to have different results. The ANN model is trained through random shuffling the data and fitting a model to an 80% training 20% testing data split. For sequential models, the data was split into 80% training, 10% testing, and 10% validation. Sequential models for GRU and LSTM were made by making a lag function, that delayed all inputs to the system by one time period. RNN was not selected for real-time prediction over LSTM or GRU based on the expectation that RNN results would be more unstable.

3.2 Tuning Hyperparameters

Table about model types used, why each type was used and some of the assumptions about the performance of each model. Then discuss training parameters, including batch size, learning rate, neuron number, epochs, and about limited tuning of hyper parameters.

When training the models, a variety of options and hyperparameters needed to be selected, and to improve effectiveness of training these need to be tuned in order for the model to produce optimal results. The first setting considered was number of epochs. An epoch is one run through of the entire data set, so the number of epochs for a system defines how many times the training will run through the data set to train the model. Learning rate adjusts how fast the modelled weight matrices change per epoch, typically set for this project around 0.05 or lower to produce stable results. As discussed previously, an activation function also needs to be chosen for each layer. All activation functions were tested, however the "ReLU" function was used in most cases as it produced the best results. These were the hyperparameters that received the most tuning, with parameters such as initialization and batch size left as the default.

One of the biggest issues experienced during training of the models was overfitting of the system to the training data. Overfitting happens when the model adheres to predicting only the training set of data, and as such loses the ability to predict future testing data. Fig.7 shows an example of this occurring, where the training function error continues to decrease as more epochs of data are run through for training the model, however the test data error begins to rise. This is a key sign of overfitting, and in cases where this was observed either the number of epochs was lowered, or the learning rate was changed.

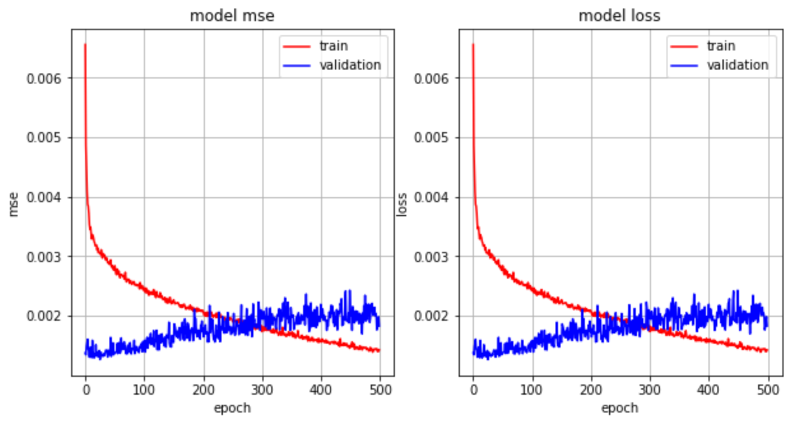


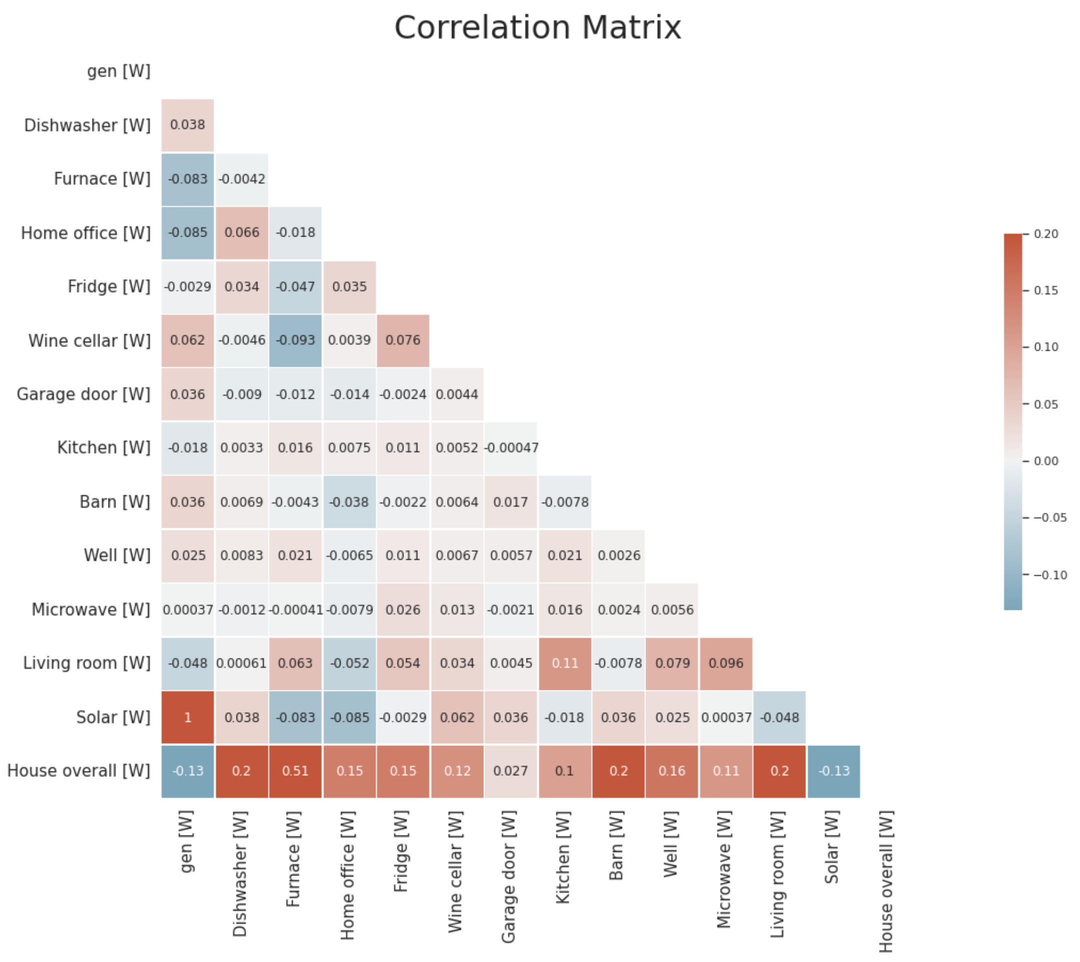
Fig. 7. Example of overfitting of a system.

4 Interpretation of Results

4.1 Result of Data Preprocessing

Before constructing a future prediction model, data processing is a prerequisite. To obtain a clear view of the time-series data, we first changed the timestamp to the data time. Second, we determined the total for the submeter-sized furnace and kitchen characteristics (i.e., furnace 1, and furnace 2 and kitchen 12 and kitchen 14). Third, the input data should be standardized to account for the varying ranges of each attribute. In this investigation, min-max normalization was utilized. The dataset was separated into training and testing datasets for training and testing purposes. In this study, 80% of the data are used to train the models, whereas 20% are chosen for testing.

4.2 Result of Exploratory Data Analysis



	gen [W]	Dishwasher [W]	Furnace [W]	Home office [W]	Fridge [W]	Wine cellar [W]	Garage door [W]	Kitchen [W]	Barn [W]	Well [W]	Microwave [W]	Living room [W]	Solar [W]	House overall [W]
gen [W]														
Dishwasher [W]	0.038													
Furnace [W]	-0.083	-0.0042												
Home office [W]	-0.085	0.066	-0.018											
Fridge [W]	-0.0029	0.034	-0.047	0.035										
Wine cellar [W]	0.062	-0.0046	-0.093	0.0039	0.076									
Garage door [W]	0.036	-0.009	-0.012	-0.014	-0.0024	0.0044								
Kitchen [W]	-0.018	0.0033	0.016	0.0075	0.011	0.0052	-0.00047							
Barn [W]	0.036	0.0069	-0.0043	-0.038	-0.0022	0.0064	0.017	-0.0078						
Well [W]	0.025	0.0083	0.021	-0.0065	0.011	0.0067	0.0057	0.021	0.0026					
Microwave [W]	0.00037	-0.0012	-0.00041	-0.0079	0.026	0.013	-0.0021	0.016	0.0024	0.0056				
Living room [W]	-0.048	0.00061	0.063	-0.052	0.054	0.034	0.0045	0.11	-0.0078	0.079	0.096			
Solar [W]	1	0.038	-0.083	-0.085	-0.0029	0.062	0.036	-0.018	0.036	0.025	0.00037	-0.048		
House overall [W]	-0.13	0.2	0.51	0.15	0.15	0.12	0.027	0.1	0.2	0.16	0.11	0.2	-0.13	

Fig. 8. Correlation between features and output.

Fig. 8 describes the correlation between the inputs and output for one year. A dark color indicates the strong correlation while a light color denotes the weak relationship. Obviously, there is a strong correlation between the energy consumption of the furnace and the total energy consumption of the house (coefficient: 0.5), which indicates that the furnace consumes more energy than other appliances because it is a multipurpose device for heating, cooling, and ventilation. Consequently, furnace may become an important factor for future energy forecast. Low correlation between inputs and output does not imply that they are useless; when paired with additional features, the prediction accuracy normally can be enhanced.

In addition, solar and generation have an extraordinarily high correlation, indicating that they have identical values and that one of them should be abolished to eliminate the redundant feature. Thus, solar was eliminated to improve the accuracy of predictions and reduce computing costs.

4.3 Performance Comparison among Machine Learning Algorithms for Future and Real-Time Energy Prediction

4.3.1 Prediction Accuracy Comparison

Table 4 Performance metrics of different neural network models.

Prediction type	Algorithms	Evaluation metrics	
		R square	MAPE (%)
Real-time	ANN	0.738	3.03
	GRU	0.400	21.16
Future	LSTM	0.680	39.60
	RNN	0.720	4.15

As seen from Table 4 above, for real-time data prediction ANN outperforms RNN in terms of performance (along with GRU which had the worst results) with a higher R squared score and lower percent error. It should be noted that for real-time prediction that neither models perform well at predicting real-time data as indicated by the performance metrics compared to future data prediction. As seen in Fig. 9, the GRU model does not have good tracking performance to the actual dataset recorded, with usage peaks having the largest visible error in tracking. These results could stem from difficulty of the systems to predict real-time data due to the nature of energy consumption rapidly changing and could also show some of the difficulties in using machine learning models in complex scenarios without proper hyperparameter tuning techniques such as grid searching. Another solution would be to change the amount of training and testing data in order to achieve different possibly better results, however this was not attempted in this project.

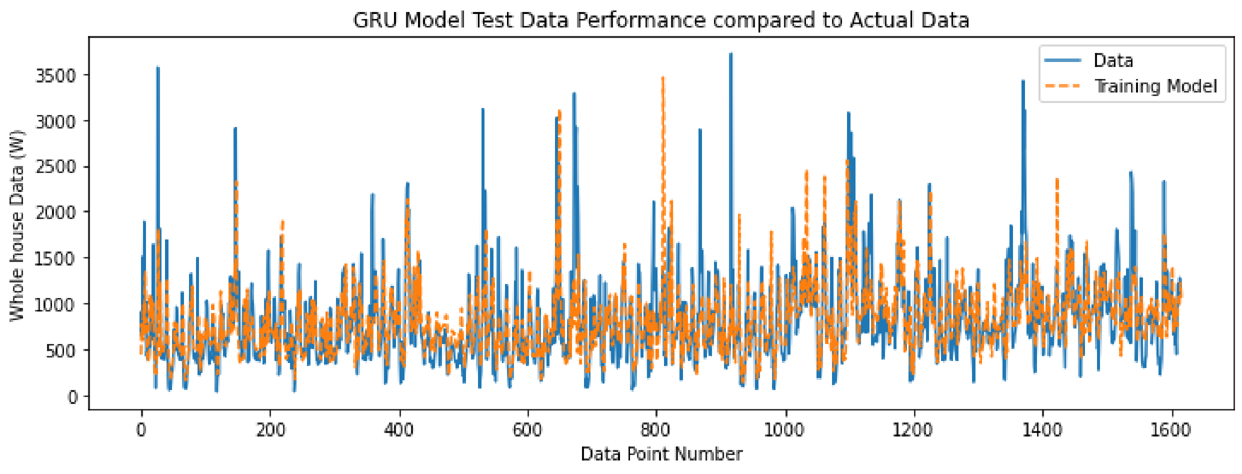


Fig. 9. GRU model real-time prediction vs. actual data.

For future based prediction models, these performed significantly better than the real-time based models, with all future prediction models outperforming their real-time prediction counterpart. RNN was found to be the best performing future prediction model based on the performance metrics selected in Table 4. This result indicated that the unstable elements of RNN benefit the modeled system in that it has the capabilities to better match energy spikes in comparison to GRU and LSTM models, which show much more stable behavior. This is showcased in Fig. 10 and 11 for RNN and LSTM respectively. Both models have different ways in which they accurately model the given data and given more tuning to hyper parameters these models could still be improved to better match the given data.

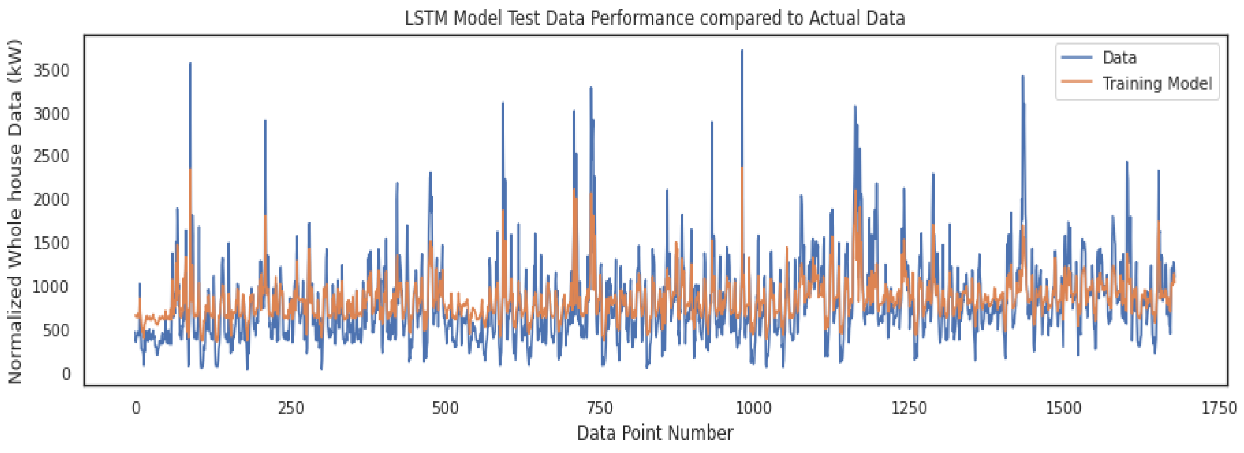


Fig. 10. LSTM model future prediction vs. actual data.

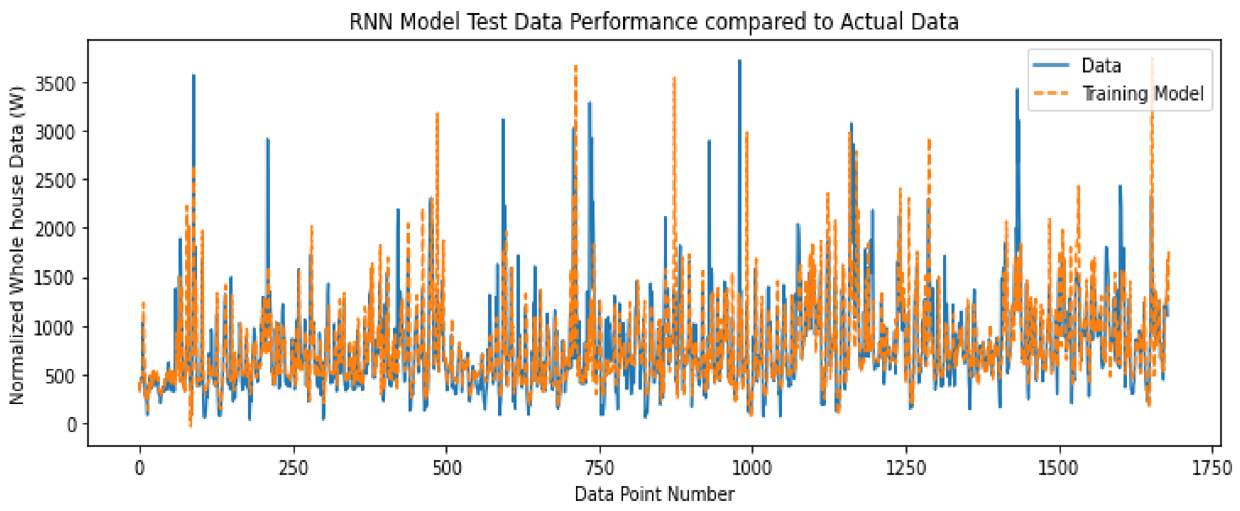


Fig. 11. RNN model future prediction vs. actual data.

4.3.2 Computational Comparison

To compare computational cost for each model type, the times recorded for training and testing are recorded below in Table 5. All times recorded were performed through Google Colab, and it is assumed the different processors used between sessions do not impact the results seen in the table. Overall, it is observed that real-time prediction models require more computational resources based on longer training times. In addition, for the same prediction types the difference in training times is not large as seen between ANN and GRU for real-time and RNN and LSTM for future in Table 5. Additionally, all times recorded are below two minutes in length, so for a project of this scale computational time can be concluded as negligible, however for larger sets of data future-based predictions are notably less complex.

Table 5. Training and testing times of different neural network models.

Algorithms	Training (s)	Testing time (s)
ANN	77.125	7.529
GRU	76.211	1.616
LSTM	43.005	1.050
RNN	37.000	0.774

5 Critique of the Model (Future Works)

For the models developed in this project, the overall quality of all models did not achieve excellent tracking compared to the data tested against. The main future goal in this project would be to further tune the parameters used in training of the model to ensure that each model was individually performing optimally. Additionally, certain aspects of the project were kept simple based on time constraint and overall knowledge of the topic area. If more areas had been explored like the creating of deeper neural networks, different activation functions, or different initialization schemes there may have been more success found. This applies mostly to the real-time prediction methods, where the performance of these models was far lower than those of the future prediction models. The reasoning for this difference could be further explored, with suspected reasonings being lack of hyper tuning parameters, more complex systems being simplified, or problems with the data that cause issues for machine learning. These issues would all require further investigation before coming to any formal diagnosis.

In conclusion, the models satisfy the answer the question posted. However, certain constraints must be addressed in the future for this topic in general.

(1) Implement feature selection/feature importance to choose the best feature combination to increase the prediction accuracy and reduce computational cost.
(2) Investigate the capability of model robustness and generalizability for long-term predictions and the energy estimation of different buildings.
(3) Integrate data-driven energy prediction model into further applications, such as model predictive and reinforcement learning control.

6 Appendices with supporting materials

The code can be found in GitHub repository (https://github.com/xyzsonic/Math509Project), and the data publicly available from a smart home with weather information (https://www.kaggle.com/datasets/taranvee/smart-home-dataset-with-weather-information). Each member's contributions are included in each commit's description.

Reference

[1] G. Pinto, D. Deltetto, A. Capozzoli, Data-driven district energy management with surrogate models and deep reinforcement learning, Applied Energy. 304 (2021) 117642. https://doi.org/10.1016/j.apenergy.2021.117642.

[2] Y. Peng, A. Rysanek, Z. Nagy, A. Schlüter, Occupancy learning-based demand-driven cooling control for office spaces, Building and Environment. 122 (2017) 145–160. https://doi.org/10.1016/j.buildenv.2017.06.010.

[3] X.J. Luo, L.O. Oyedele, Forecasting building energy consumption: Adaptive long-short term memory neural networks driven by genetic algorithm, Advanced Engineering Informatics. 50 (2021) 101357. https://doi.org/10.1016/j.aei.2021.101357.

[4] I. Karijadi, S.-Y. Chou, A hybrid RF-LSTM based on CEEMDAN for improving the accuracy of building energy consumption prediction, Energy and Buildings. 259 (2022) 111908. https://doi.org/10.1016/j.enbuild.2022.111908.

[5] Z. Pang, F. Niu, Z. O'Neill, Solar radiation prediction using recurrent neural network and artificial neural network: A case study with comparisons, Renewable Energy. 156 (2020) 279–289. https://doi.org/10.1016/j.renene.2020.04.042.