Predicting energy consumption of a building based on historic usage rates and weather data

Tamara Šumarac
Department of Physics
Harvard University
Cambridge, MA 02138
tsumarac@g.harvard.edu

Abstract

One of the biggest problems facing the world today is global warming. With electricity and heat production being one of the biggest contributors to the greenhouse gases, there is a demand in constructing a more energy-efficient buildings and retrofitting the old ones. An estimate of how much energy would improved building consume relative to the old one can incentivize investors to pursue these renovations. For this purpose it is useful to have an accurate prediction of how much energy would a building consume in future given how much that same building was consuming in the past. In this project I used the data [1] of historic usage rates and observed weather over 1 year period across hundreds of buildings in order to make energy consumption predictions. I investigated several regression and neural network models including linear regression, decision tree regression, random forest regression, long short term memory (LSTM), and a combination of convolutional neural network (CNN) and LSTM. For the regression models I also compared dependence of model results on different input feature encoding. Finally, in order to understand what kind of predictions a non-expert in the field would be able to make, I also used an automated regression model generated by autosklearn. For the data used in this project, CNN-LSTM model gave the best prediction of the data on a short-term timescale, whereas random forest regression model gave the best predictions of the future data on longer timescales out of all the models considered.

1 Introduction

Emission of greenhouse gasses into the atmosphere is known as the main cause of global warming. Through human activity, specifically burning fossil fuels, deforestation and farming, large amounts of these gasses are released into the atmosphere every year, causing our planet to warm up at a rapid speed. Energy generated by fossil fuels is mostly used for electricity and heating, and for the purposes of optimizing this energy consumption there is a demand in constructing a more energy-efficient buildings and retrofitting already built ones. To motivate investors to pursue renovations needed to optimize energy consumption of existing buildings, an accurate estimate of how much energy would improved building consume relative to the old one is important. One can make these estimates using programs based on physical principles that take as input climate data and many building features data such as geometry, materials used for constructing, positioning of the building and its orientation. These programs have been greatly utilised, however they can become very costly to handle when one wants to define large number of retrofits [2]. An alternate approach to estimating energy consumption would be to make these estimates using historical usage data. In this method, building features, building historical usage data and climate data are used together to predict building's energy consumption by applying a learning process. In this project I am using the latter approach

Preprint. Under review.

to estimate buildings energy consumption. In subsections to follow I will describe the data I used for the project along with the objective function. I will also provide a brief description of learning algorithms used for this report, with more details on this described in Methods section.

1.1 Data

In this project I used the data [1] of historic electricity usage rates and observed weather over one year period across hundreds of buildings in order to predict how much energy would these buildings consume in the future.
The input features for the learning model can be split into three categories: consumption features, building features and weather features, where the output of this model, target data, would be the energy consumption in units of kWh. Consumption feature includes timestamp feature that will indicate at which point in time was the electricity reading taken. Building features include square feet, year built and floor count, and weather features include air temperature, cloud coverage, dew temperature, precipitation depth, sea level pressure, wind direction and wind speed.

1.2 Evaluation criteria

Since we would like to predict energy consumption of a building, which is a number, I will use root mean squared error (RMSE) as a natural evaluation metric:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(g_i - y_i)^2} \tag{1}$$

where g_i is our estimated value and y_i is the target value.

1.3 Learning algorithm

There is a vast literature of different approaches used for building energy consumption predictions. In review [3] authors have discussed the performance of Artificial Neural Networks, Support Vector Machine and hybrid methods for load forecasting. In [2], in addition to previously mentioned techniques, authors have also discussed application of Gaussian distribution regression models and hierarchical clustering for this problem. In [4] authors have investigated the effectiveness of Long Short-Term Memory (LSTM) technique and its variations in prediction of electricity consumption data. Similarly, authors in [5] have investigated the Convolutional Neural Network (CNN) and LSTM combination in order to predict energy consumption data of residential buildings.
For this report I compared predictions I get using various regression models including linear regression, decision tree regression and random forest regression model as well as predictions I get using more involved neural network models including LSTM and CNN-LSTM combination. Details on the models I used are provided in Methods section.

2 Methods

2.1 Data processing and data selection

For this project I did two types of data processing and selection: one for the features data and one for the target data. Details on the work done is provided in Appendix.

2.2 Timestamp feature encoding for regression models

For regression models used in this report, as in contrary to the LSTM models used, I did feature encoding of the timestamp feature in order to ensure good fit for the data with sinusoidal looking daily and weekly variations. For this report I did two different types of encoding and I compared how these different encodings affect the goodness of the fit. I also compared if we were to do no encoding to the features, what would the results look like in that case.
First encoding is what I called simple encoding in my code [6]. For this encoding I replaced the timestamp feature with hour, month and year features. After this, I normalized all the input features,

including these and I used this as an input for my learning algorithm.
Second encoding, that I called polynomial encoding, was inspired by Moon et al [7]. Given that energy consumption has clear daily variations, if we encode day as a numeric feature, then even though 0000hrs follows right after 2359hrs, they will numerically look like they are far apart. This encoding solves this problem by substituting day feature with day_sin and day_cos feature, with frequency that corresponds to 24h. I used this type of encoding for day, week and year feature with frequency of 24h, 7 days and 365 days respectively in order to reflect the cyclical nature of electricity consumption. I also added additional feature called is_weekday to handle lower amplitudes of energy consumption over weekend relative to weekdays. In addition to timestamp feature, in polynomial encoding case, I also did feature encoding for the wind features. Wind_direction feature in units of degrees has similar issue as described above, where 0 and 359 degrees are numerically far away but physically they are close. Thus I substituted wind_speed and wind_direction features by combining them together into wind_x and wind_y features. More details on these encodings can be found in the Github repository [6] for this project. After these encodings, I normalized all the input features before sending them to the learning algorithms.

2.3 Regression models

For this report I compared the results of linear regression, decision tree regression, random forest regression and autosklearn regression in combination with different feature encodings as described in Timestamp feature encoding for regression models section in order to find the best prediction model for the data. For all the models I dedicated month of December to be the test data, and the rest of the months of the year to serve as training data for the models.

2.4 LSTM and CNN-LSTM model

For this report I tried two different LSTM based models: plain LSTM model that I used as a benchmark and the better performing model based on CNN-LSTM combination. For both models, we make consumption prediction at variable time in advance. Input data for both models is a sequence of 24h consumption, building and weather features data prior to the time at which we want to make our prediction. A visual depiction of how the input data was shaped is given in Appendix. For both models I dedicated month of December to be the test data, and the rest of the months of the year to serve as training data for the models.

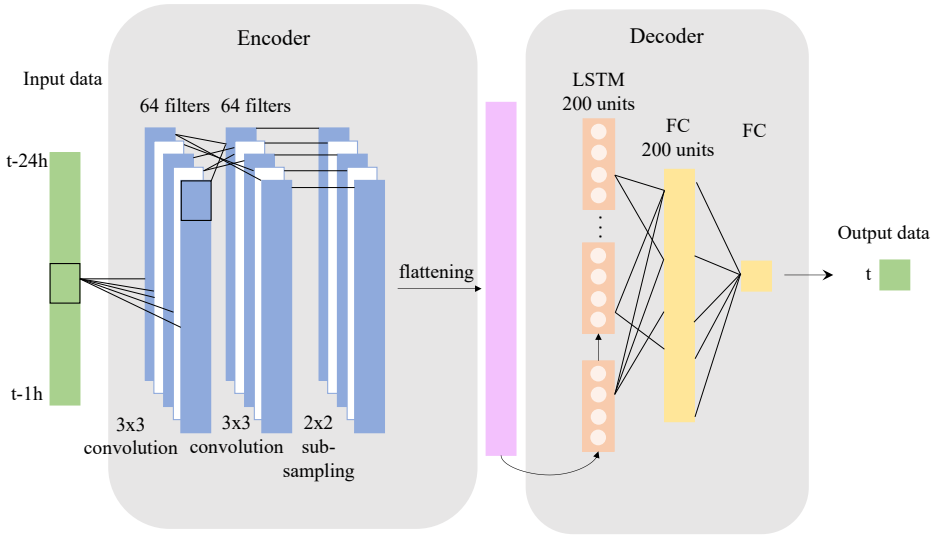

Figure 1: CNN-LSTM model architecture.

For plain LSTM model, I used single hidden LSTM layer with 200 units, followed by fully connected layer with 200 nodes and Relu activation that is intended to interpret the features extracted by LSTM

model. This is then followed by another fully connected layer that will provide output sequence we are interested in.
Results of plain LSTM model suggested that it could be beneficial to have better processing of input sequence features before sending them to the LSTM layer. This lead me to make a second LSTM based model for this project, inspired by [5], where I used a combination of CNN and LSTM layers based on encoder-decoder architecture. Encoder of the data was comprised of two convolutional layers followed by a max pooling layer, the results of which were then flattened. The first convolutional layer takes input sequence and extracts output that extracts certain features of the input. These features are then fed as input into the second convolutional layer that is trying to amplify any salient features. I used 64 filters for both convolutional layers with kernel size of three time steps. After convolutional layers I used max pooling layer with pool size of 2 to simplify the extracted features and keep only 25% of them with largest signal values. These extracted features were then sent to decoder part of the model. The decoder of the architecture was made out of single hidden LSTM layer with 200 units, followed by fully connected layer with 100 nodes and Relu activation function. Finally at the end I added one more fully connected layer whose output will provide the sequence we are interested in. A visual depiction of encoder-decoder model is presented in Figure 1.

3 Results

3.1 Regression model results

Using root mean squared error as a metric, the Table 1 provides the breakdown of results that different regression learning algorithms in combination with different feature encodings give.

Model	Polynomial	Simple	No encoding
Linear regression	0.104	0.148	1.001
Decision tree	0.037	0.048	0.114
Random forest	0.034	0.047	0.110

Table 1: RMSE of different regression learning models in combination with different feature encodings. All the data is obtained on 2-4min timescale.

Table above shows that the best results are obtained for random forest model with polynomial feature encoding. Visual comparison of predictions of this model w.r.t. target data is presented in Figure 2.

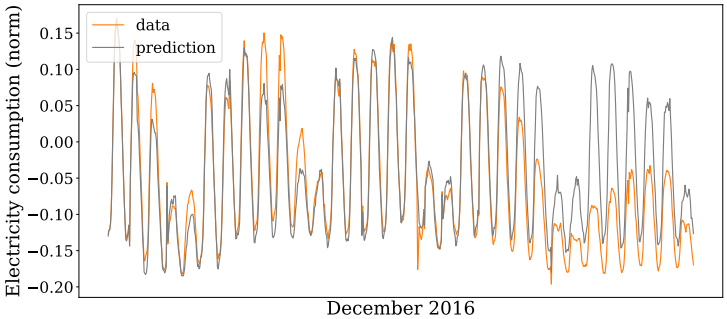

Figure 2: Random forest regression model with polynomial feature encoding fit for the month of December 2016 of averaged electricity consumption of 118 educational buildings. The values of electricity consumption presented here are normalized w.r.t. training data mean and standard deviation values.

As one can note from the graph, random forest regression model performs quite well even on the quite unusual data for the month of December, that is specially affected by holiday season towards the end of the month. This model makes the most errors in this holiday season time, and one could potentially improve these predictions by adding additional holiday feature that would know if that timestamp belongs to a holiday season.

3.2 LSTM results

Table 2 summarizes the results obtained with plain LSTM model and CNN-LSTM model as described in Methods section. For both models we looked at predictions these models make from 1h in advance to 24h in advance.

Model	1h in advance	12h in advance	24h in advance
plain LSTM	0.046	0.088	0.090
CNN-LSTM	0.022	0.092	0.111

Table 2: RMSE results of different LSTM based models of 1h, 12h and 24h predictions in advance. All the data is obtained on 10min timescale.

The best results are obtained using CNN-LSTM based model with predictions made 1h in advance. These predictions have better RMSE values compared to best regression model, and this is mainly because CNN-LSTM model handles better the last week consumption prediction for the month of December. This better handling of the consumption prediction for the last week of December is, ofcourse, a direct consequence of us making prediction with this model only 1h in advance, meaning that we rely on having all consumption data values prior to prediction timestamp. As can be noted from the Table 2, as we start to make predictions longer in advance, the model is starting to make worse predictions. These results are discussed in more detail in Discussion section. Visual depiction of how this model is performing when making predictions 1h in advance is given in Figure 3 and the results of 12h in advance and 24h in advance predictions are given in Appendix.

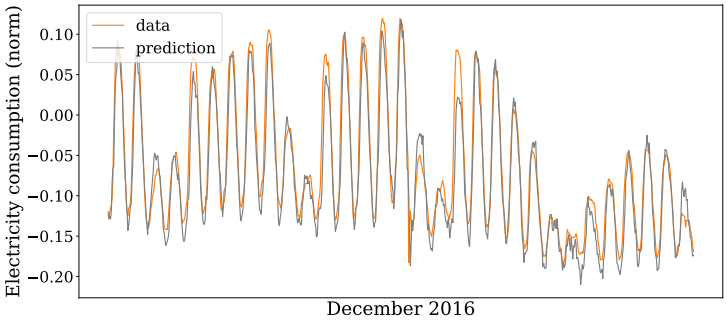

Figure 3: CNN-LSTM model prediction of averaged electricity consumption of 118 educational buildings for the month of December 2016 predicted 1h in advance. The values of electricity consumption presented here are normalized w.r.t. training data mean and standard deviation values.

3.3 Auto-ML results

An interesting question that comes to mind is whether it would be possible to find good predictor for the data without having any expert knowledge about what feature encoding to do or which model to use. An automated toolkit that can choose a good algorithm and feature processing such as autosklearn can be used to answer this question. Given that regression models, compared to LSTM based models I tested, gave better predictions of the data further in the future, I tried to automate finding right regression model and feature encoding using autosklearn's regression model. As can be seen in the Table 3, if we did no feature encoding before sending the data to autosklearn, we would not be able to make very good predictions, with metric obtained in this case being around $4\times$ larger than the best value we were able to obtain using random forest regression. However, if we were to do a polynomial feature encoding, autosklearn would be able to provide results that are close to the best results obtained in this report, with RMSE value being around $1.5\times$ larger than the RMSE value of the random forest regression model. This suggests that autosklearn could do well with automation of finding the right learning algorithm to fit the data, however, automation of feature processing is something that seems not to be optimized, at least not for this data.

Model	Polynomial	Simple	No encoding
Autosklearn	0.049	0.055	0.121

Table 3: RMSE of autosklearn model in combination with different feature encodings. All the data is obtained on 1h timescale for each table input.

4 Discussion

The aim of this project was to answer several questions: Which model out of all the ones tested gives us the best prediction of the future data and on what timescale are those predictions good? Which input features do we assume to have available to make these predictions? If a non-expert in the field was to make predictions, how well could they do without putting a lot of effort to learn the data and machine learning? In the next couple of paragraphs I will try to address these questions.

4.1 Which model performs the best and on what timescale?

Looking at the results given our RMSE metric, the best results we obtained are the ones using CNN-LSTM model for predictions made 1h in advance. However this model seems to perform well only on very short-term scale, with acceptable performance with predictions up to 6h in advance. Past this point, random forest regression has much better performance in predicting future data, with main error for this model coming from not handing data for the holiday season properly, which could be better handled by adding additional "is it holiday season" input feature. Literature in forcasting building's power consumption with LSTM based models [5, 8] suggest that one could obtain better prediction of the future data if one was to use daily or weekly power consumption data compared to hourly as used in this report. This smaller resolution data has much smaller variations over time compared to hourly data, however the focus of this report was to make hourly future data predictions so this situation was not addressed here, but could be interesting to explore further.

4.2 What input features do we assume to have available to make predictions?

An interesting question that comes to mind is, if one wants to make predictions in the future, what kind of input features can one assume to have available at the moment predictions are made.
One could imagine a situation where we would like to make prediction of the future consumption data without having climate data input, because this data is also something we would receive in the future. For the case of CNN-LSTM model, prediction values don't depend much on whether we have climate features as input or not, with only slight increase in RMSE without their inclusion. On contrary, regression based models are quite dependent on the climate data features, and without their inclusion, predictions made are several times worse depending on regression model chosen. This suggests that in situations when we want to predict future consumption without having weather data available, the best model we could use in such situation would be CNN-LSTM model with good predictions up to 6-12h in advance.
If, however, we were in a situation where building went through a retrofit, and we wanted to understand if investing in a building's retrofit was the right choice, we would want to compare what is retrofitted building's consumption relative to our expectation of the old building's consumption for the same time period. In this case, one could assume that weather data would be available and we would probably want to compare consumption of two buildings over a year or several years time period, and in this case, random forest regression model would be a better model to use to make predictions.
Another interesting question that comes to mind is, if an unexpected electricity consumption is about to happen in the future, could any of the models used in this report be used to make this prediction. If for example, an unexpected consumption is closely related to the input features, such as weather data, then it might be possible that all the models used in this report would be able to make unusually different consumption prediction in the future given that weather data will also have unusually different input at that timestamp. If, however, we either don't have the climate data that will have very unexpected behaviour in the future or an input feature not considered in the model was to dramatically change, in that case, I would believe neither of the models would be able to do good

predictions of the future consumption. Potentially, a CNN-LSTM model could be used, once unusual consumption values are recorded for couple of timestamps, in order to make near-term consumption prediction on couple of hours in advance timescale.

4.3 How good of predictions could non-expert make?

Autosklearn regression model was used to understand what kind of predictions would a non-expert in the field make. As discussed in Auto-ML results section, this toolkit seems to provide comparable results to the best predictions we get with random forest model and CNN-LSTM model only when we do input feature encoding prior to sending this data to the learning algorithm. When sending data to learning algorithm with no prior feature encoding we get 3-4$\times$ worse RMSE results compared to best predictions obtained in this report and the more involved feature encoding used for this project gave 10% better results compared to simple encoding. These results suggests that a non-expert wanting to make building predictions with autosklearn regression model would need to make some transformation on the input data to make them look "more linear" for this model to provide good results. This suggests that, at least for the data used in this project, a non-expert might not be able to make good predictions with autosklearn regression model without any knowledge about what the data looks like and without pre-processing this data prior to learning.

5 Conclusion

In conclusion, the best predictions of the future data in this report was obtained with CNN-LSTM model, giving best predictions of the data 1h in advance with acceptable predictions of the data up to 6h in advance. For long term predictions, random forest regression model gave the best results, with RMSE metric being less than 1.5$\times$ larger than the best RMSE result obtained with CNN-LSTM model for 1h in advance predictions. In this report we also addressed a question of whether it is possible for non-expert in the field to make good predictions of the data using automated machine learning toolkit autosklearn. The conclusion is that autosklearn could do well with automation of finding the right learning algorithm to fit the data, however, automation of feature processing is something that seems not to be optimized for this data.

References

[1] URL: `https://www.kaggle.com/c/ashrae-energy-prediction`.

[2] Saleh Seyedzadeh et al. “Machine learning for estimation of building energy consumption and performance: a review”. In: *Visualization in Engineering* 6 (2018).

[3] Mohammad Azhar Mat Daut et al. “Building electrical energy consumption forecasting analysis using conventional and artificial intelligence methods: A review”. In: *Renewable and Sustainable Energy Reviews* 70 (2017), pp. 1108–1118.

[4] Daniel L. Marino, K. Amarasinghe, and M. Manic. “Building energy load forecasting using Deep Neural Networks”. In: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society* (2016), pp. 7046–7051.

[5] Tae-Young Kim and Sung-Bae Cho. “Predicting residential energy consumption using CNN-LSTM neural networks”. In: *Energy* 182 (2019), pp. 72–81. ISSN: 0360-5442.

[6] URL: `https://github.com/TamaraSumarac/6.862_MachineLearningProject`.

[7] Jihoon Moon et al. “A comparative analysis of artificial neural network architectures for building energy consumption forecasting”. In: *International Journal of Distributed Sensor Networks* 15.9 (2019).

[8] Choi, Cho, and Kim. “Power Demand Forecasting using Long Short-Term Memory (LSTM) Deep-Learning Model for Monitoring Energy Sustainability”. In: *Sustainability* 12 (Feb. 2020), p. 1109.

A Data processing details

As described in the main text, I did two types of data processing and selection: one for the features data and one for the target data.

A significant proportion of features data was missing in the downloaded data files. The breakdown of the missing data for the input features is the following: 53.4% of the year built, 75.5% of the floor count, 9.5% of the cloud coverage, 0.1% of the dew temperature, 36.0% of the precipitation depth, 7.6% of the sea level pressure, 4.5% of the wind speed and 0.2% of the wind direction. In order to have the largest number of data for the training, I filled these missing elements for all the input features. For the weather data features, I filled the missing data by taking the average of the closest non missing values of that feature, whereas for the building data I filled the missing data for the feature with the mean of that feature for that building's primary use group.
For the target data, the breakdown of the missing data is the following: 36% of the buildings had consumption data that was missing for more than 7 days and 83% of the buildings had missing data for more than 1 day. Given that consumption data has large daily and weekly variations, and given that this data should be what we predict, I chose not to include the data of the buildings that have missing values for more than a day. For this report I worked only with buildings whose primary use is educational, which ended up being 118 buildings that I used for learning in total. This project can be further extended to include more data by using one-hot encoding for building's primary use feature.

B LSTM input data structure

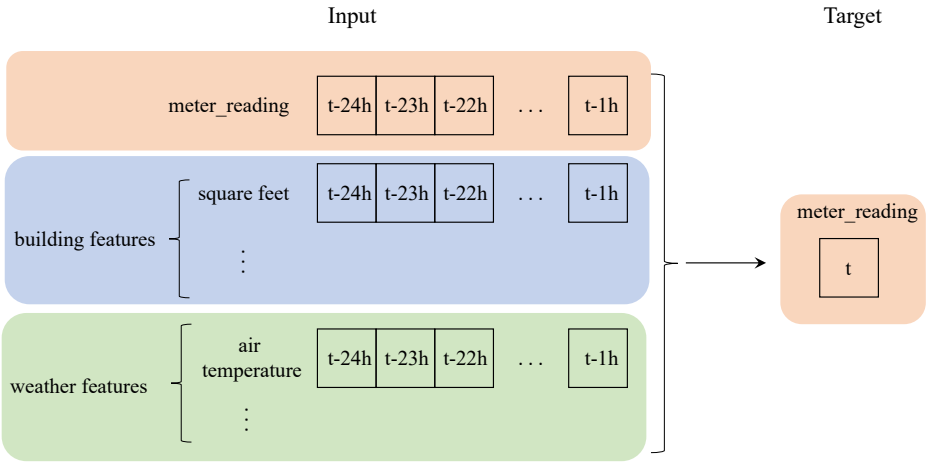

Figure 4: Input data structure sent to LSTM based learning algorithms.

C CNN-LSTM predictions of 12h and 24h in advance

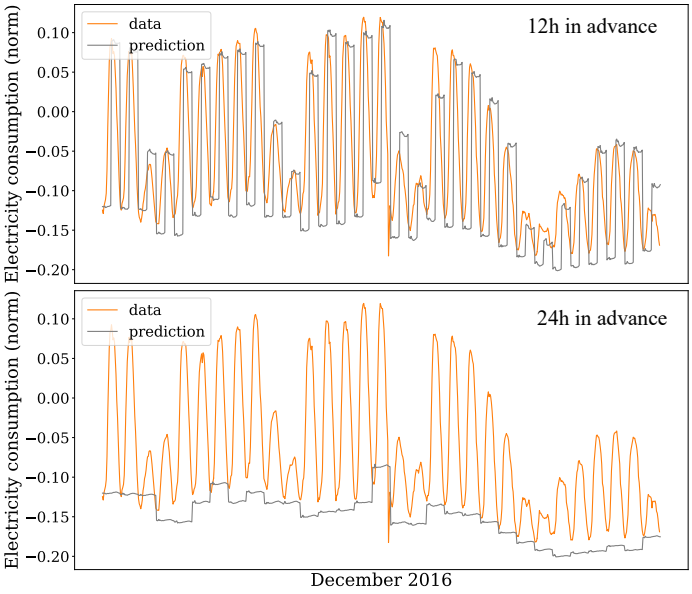

Figure 5: CNN-LSTM model prediction of averaged electricity consumption of 118 educational buildings for the month of December 2016 predicted 12h in advance (top) and 24h in advance (bottom). The values of electricity consumption presented here are normalized w.r.t. training data mean and standard deviation values.