Automating Energy Demand Modeling and Forecasting Using Smart Meter Data

Poojitha Amin
Arm Research
San Jose, CA 95134, USA
poojitha.amin@arm.com

Ludmila Cherkasova
Arm Research
San Jose, CA 95134, USA
lucy.cherkasova@arm.com

Rob Aitken
Arm Research
San Jose, CA 95134, USA
rob.aitken@arm.com

Vikas Kache
Arm Inc, ADIM Dept
San Jose, CA 95134, USA
vikas.kache@arm.com

***Abstract*—The Internet of Things (IoT) technology with a variety of smart devices, communication networks, and software systems for data processing is critical for optimizing Smart Grid operations. While energy demand forecasting is an integral part of Smart Grid management, there is no a general forecasting method which works best across different situations and scenarios. In this paper, we design, build, and compare three forecasting methods to show how these approaches can be trained on time-series data with limited intervention from users, i.e., enabling automated model selection. Using smart meters measurements collected from 114 residential apartments over two years, and weather information for the same period, we build the following three models: (i) a *piecewise linear regression* model, (ii) the *univariate seasonal ARIMA* model, and (iii) the novel *multivariate LSTM* model (a special form of Recurrent Neural Network). While the designed linear regression model could be used for longer-term planning, the constructed LSTM model significantly improves the accuracy of short term (24 hours) demand prediction compared to both the linear regression and ARIMA models. However, training LSTM and modeling with ARIMA require higher computing resources and longer runtimes (45 min and 125 min respectively) compared to 2 min for building and predicting with linear regression model. Therefore, the automated *best* model selection can be driven by the user-defined metrics of interest, such as model accuracy, required computing resources, model usability, and model construction/execution time.**

***Index Terms*—Smart Grid, smart meters, data analytics, demand forecasting, automated model selection.**

I. Introduction

Utility companies are looking to leverage IoT to improve the management and operation of Smart Grids [1]. One of the enabling components of Smart Grid is the Advanced Metering Infrastructure (AMI) [2]. AMI is an integrated system of smart meters [3], communication networks, and data management systems, that supports two-way communication between utilities and customers. It enables new useful functions such as the ability to automatically and remotely measure electricity use, connect and disconnect service, identify outages, etc.

Understanding patterns and trends of energy consumption in the residential sector is crucial for utility companies to properly support and provision their current and future services. In this paper, we consider the issues of workload analysis, performance modeling, and demand forecasting based on collected historical data. We analyze the *UMass Apartments dataset* – smart meters data measurements and weather temperature reports available over 2 years at a fine time granularity [8].

First, we build a *piecewise* linear regression model to forecast future energy demands. This model could be used as an effective tool for analyzing a variety of *"what-if"* scenarios, e.g., impact of colder/warmer weather for optimizing the future energy distribution across the utility grid.

Using the same training data, we design the *univariate seasonal ARIMA* model and a novel *multivariate LSTM* forecasting model, which is a special form of RNN. Our results show that the designed LSTM model improves the accuracy of short term (24 hours) demand prediction by 31% and 19% compared to the piecewise linear regression model and ARIMA results respectively. However, constructing LSTM and predicting with ARIMA require higher computing resources and longer runtimes (45 min and 125 min respectively) compared to 2 min for building the linear regression model.

We introduce elements of automation in each of these models to take away the burden of manual and tedious modeling tasks. In building an ARIMA model, searching over all possible models with different hyper-parameters is time consuming. We describe a series of tests on the time-series data to establish ARIMA hyper-parameter configurations, so the model's effectiveness is more readily attained in practice. We implement a break point detection technique to determine the number of breaks in the piecewise linear regression model. The simulations show that our procedure works well in identifying the segments for energy usage in residential buildings. We demonstrate the automated feature selection in the data to aid accurate predictive model.

Once the forecasting problem is formulated and modeled, the users can choose the right forecasting technique based on the metrics results, run time and computing resource requirements. This way, the user's choice of the model can be driven by important considerations, such as modeling intentions (i.e., short vs long term forecast, or the analysis of possible hypothetical scenarios), performance accuracy, required computing resources as well as the model construction, training, and the execution time.

The remainder of the paper presents our results in more detail.

II. Energy Consumption Data Analysis

The dataset used in our study is based on the *UMass Apartments dataset*, which is a part of the UMass Trace repository [8]. The created data collection infrastructure (built as part of the Smart* project) records the smart meter data from real homes in Western Massachusetts for 2015 and 2016. For 2015, the measurements of electricity usage from 114 residential apartments are reported at 15 minute granularity. For 2016, the dataset contains one minute granularity electricity usage from all 114 residential apartments. Moreover, the *UMass Apartments dataset* includes the weather station data (reported each hour) with key weather variables such as humidity, pressure, temperature, wind speed, and rainfall.

First, by using 2016 data (collected at 1 min granularity), we aim to analyze the effect of different time scales on average energy consumption. The energy consumption is known to vary significantly from minute to minute, hour to hour, day to day, and month to month. Figure 1 shows the energy consumption of an arbitrarily selected apartment at 1 min granularity during a day. We observe that at a minute granularity the energy usage is very bursty: the proportion of intervals with high and low energy values is very high, making it practically impossible to accurately forecast future energy usage values at this scale.

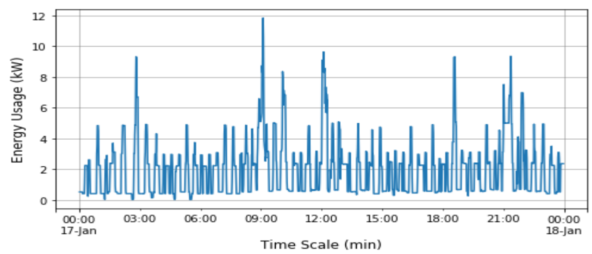

Fig. 1. Energy usage of an apartment over a day at 1-minute scale.

Figure 2 shows the energy consumption aggregated for all 114 apartments at an hourly scale. The aggregate consumption plot is smoother, it has pronounced usage peaks. It suggests that many apartments in the collected dataset have similar daily energy usage patterns.

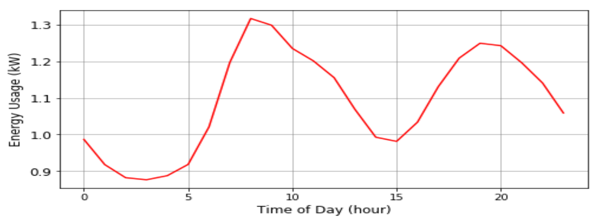

Fig. 2. Mean hourly energy usage for all 114 apartments over a day.

Most people spend active time at home during the mornings (before they leave for work) and in the evenings (after they get back home from work or school). Therefore, they tend to use the electrical appliances more often during the morning/evening ramp resulting in the increase in energy usage between 5:00 am and 8:00 am followed by an early evening time window of 3:00 pm to 8:00 pm. Two peaks shown in Figure 2 demonstrate this pattern.

Additionally, the energy consumption curve has different seasonal usage patterns, i.e., it changes with the season of the year. Figure 3 shows the mean energy consumption across 114 apartments plotted for different months over two years, while Fig. 4 shows the measured temperature for the same period.

The energy usage and weather temperatures over two years are similar (while 2015 year was a slightly colder year with a slightly higher overall energy consumption compared to 2016).

Figure 5 shows the association between the different contributing factors, i.e., weather temperature during the month of January (12 am to 5 am time interval) and the mean hourly energy consumption during this period. The red regression line running via the set of data points shows a good linear relationship between the weather temperature and the energy consumption.

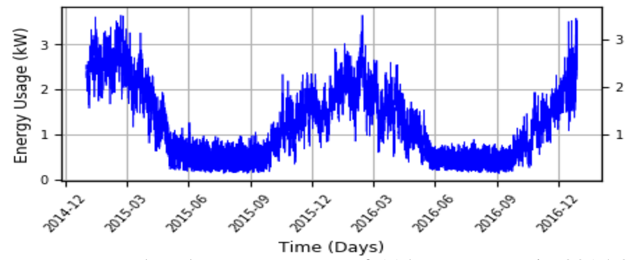

Fig. 3. Mean hourly energy usage of 114 apartments in 2015-2016.

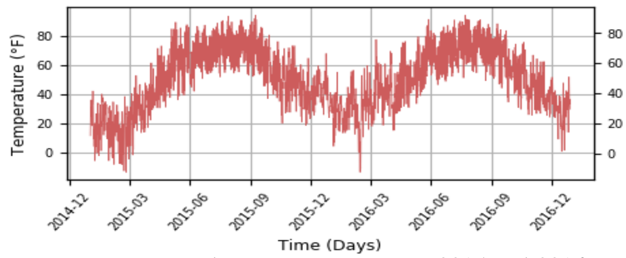

Fig. 4. Hourly temperatures across 2015 and 2016.

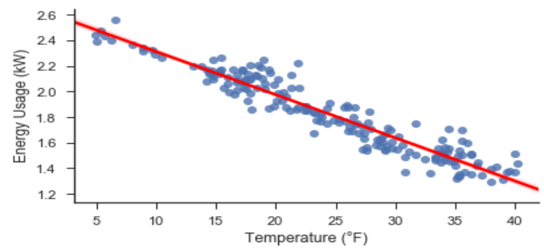

Fig. 5. Linear regression model fit for the mean hourly energy consumption during January, 2015, 12 am to 5 am time interval.

III. A Piecewise Linear Regression Model

Our goal is to predict the aggregate hourly energy demand of a given residential area, which is one of the primary forecasting problems for utility companies. As we have shown in Section II, there is a number of timing parameters of importance for modeling and predicting the energy demands: *season* and *time of the day*. Therefore, we aim to build a piecewise linear regression model characterizing the energy demand as a function of weather temperature in specific time intervals: such as month of a season and time of the day:

1) *Building the season specific models:* Residential energy sector shows a high seasonal variation in energy use, with significant demand spikes during late autumn, winter, early spring. To capture this seasonal variation, we build *12 monthly regression sub-models*, one for each month of the year.

2) *Building the models based on time of the day energy usage patterns:* As shown in Section II, there are specific energy usage patterns (peaks and lows) at a daily scale which require careful modeling. We use an *automated way to detect peaks and lows* of energy consumption over 24 hours of a typical day (obtained by averaging the hourly demands across the same hour in the dataset).

In the UMass apartment dataset, we identified *5 daily usage patterns* according to peaks and lows shown in Figure 2: 1) 12:00 am – 05:00 am, 2) 06:00 am - 08:00 am, 3) 09:00 am - 03:00 pm, 4) 04:00 pm – 08:00 pm, and 5) 09:00 pm – 12:00 am. For each of the five daily intervals, we are creating a linear regression model that reflects the energy usage in the given time interval.

Therefore, we aim to build (in automated way) a piecewise linear regression model[1] for different time segments:

$$M_{i,j}(month_i, hours\_of\_day_j)$$

where $1 \le i \le 12$ and $1 \le j \le 5$ based on the data properties of the UMass apartment dataset. Each model $M_{i,j}$ is built by using the corresponding subset $Data_{i,j}$ of time series data from the original dataset. Therefore, we can form the following set of equations:

$$E_n^{i,j} = c_0^{i,j} + c_1^{i,j} \times T_n^{i,j}, \text{where}$$

- $E_n^{i,j}$ is the energy usage for hour n in $Data_{i,j}$;
- $T_n^{i,j}$ is the weather temperature for the same hour n;
- $c_0^{i,j}$ and $c_1^{i,j}$ are the regression coefficients.

To solve for $(c_0^{i,j}, c_1^{i,j})$, we use a Least Squares Regression.

IV. Seasonal ARIMA Model

ARIMA offers a model-driven approach to forecasting time series. It is a powerful combination of two models: AR (Auto-Regressive) and MA (Moving-Average), i.e., it is a linear function of the past values and the moving average errors. It can be expressed as follows: $y_t = f(y_{t-p}, e_{t-q}) + \mu$, where

- y_t is the predicted energy usage at time t,
- y_{t-p} are the p past energy usage values,
- e_{t-q} is the error from q past moving averages,
- μ is the constant.

ARIMA model, usually denoted as ARIMA(p, d, q), takes three parameters p, d, and q for prediction. A value of p refers to the auto regressive lags (relationship between current and past observations), q stands for the moving average lags (forecast errors of the moving average model), and d is the order of differentiation.

We perform the ARIMA modeling and automated parameters' tuning in following three phases:

• **Checking for data stationarity:** In order to model a time series using ARIMA, the series has to be stationary. A time series is said to be *stationary* if its statistical properties like variance, covariance and mean remain constant over time. From our earlier plots, we could visually see that the data is not stationary (variation in energy consumption across months), and we confirm this by using the *Augmented Dickey-Fuller test*. Therefore this requires *differencing* of the series before proceeding with ARIMA modeling. This is done by setting the differencing parameter d as 1. It takes the time series, and for each value in the series, it subtracts the previous value: $y_{dt} = y_t - y_{t-1}$, where

- y_{dt} is the differenced energy usage value at time t,
- y_t is the actual energy usage at time t,
- y_{t-1} is the energy usage at time $t-1$.

• **Identifying orders of p and q:** After making the data stationary, the next step is to identify the AR and MA terms. It is done by plotting the ACF *(Autocorrelation Function)* and PACF *(Partial Autocorrelation Function)* on the differenced series shown in Figure 6.

[1]See our earlier paper [9] for more details on energy forecasting and regression modeling at multiple time scales, augmented with correlation analysis of weather features and energy usage.

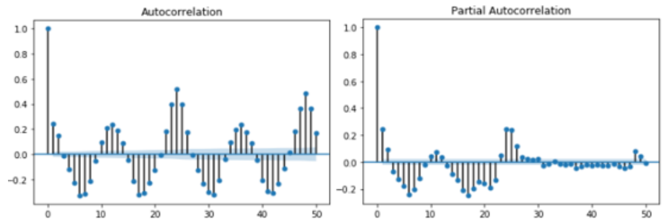

Fig. 6. ACF (left) and PACF (right) for building ARIMA model.

ACF plot gives the correlation between the series and its lags, whereas PACF measures the degree of association between the two observations. The ACF plot (Figure 6, left) shows a recurrent pattern with spikes at every 24^{th} lag (see spikes at 24 and 48 data values), which is a sign of a seasonal pattern involved. Therefore, we make use of a *seasonal ARIMA model*, which is denoted as ARIMA$(p, d, q)(P, D, Q)$, where P, D, and Q identify the seasonal part of the model.

The regularity of the time series is 24, denoting the *daily pattern*. Since the lag at 24 is positive, it suggests $P = 1$ and $Q = 0$. The lags after which ACF and PACF first time get negative values indicate the number of AR and MA terms respectively. As shown in ACF and PACF plots in Figure 6, this happens at $lag = 2$. This suggests using $p = 2$ and $q = 2$ as the orders of ARIMA parameters. All the terms put together gives us ARIMA$(2, 1, 2)(1, 0, 0)$

• **Forecasting**: In this study, we focus on the *ARIMA univariate model*, i.e., the future energy usage forecast is only driven by the *past energy usage data* from the history. We perform a multistep rolling forecast using ARIMA model to determine the energy usage in the next *24 hours*. This was implemented as a sliding window rolling forecast for the entire year of 2016, predicting the usage for the next *24 hours* as a multistep forecast, followed by sliding the window by those many steps and then retraining the model after each forecast.

V. LSTM Model

The *Long Short-Term Memory (LSTM)* is a Recurrent Neural Network (RNN) proposed by Hochreiter and Schmidhuber in [10]. LSTM can seamlessly model *multivariate* time-series data. Additionally, it supports scenarios when predictions depend on the historical context of inputs, rather than the last input. The LSTM architecture has the ability to recognize patterns in the presence of correlation between the time series and its lagged versions. This modeling approach enhances RNN by enabling long term persistence, and unlike other neural networks, it does not suffer from vanishing gradient problem.

We aim to build an LSTM model to provide the energy usage forecast for each hour in the next day (i.e., the next 24 hours), given the energy usage data along with the calendar and temperature details of the last 2 days, i.e., previous 48 hours. We use the entire data from 2015 (i.e., 365*24=8760 hours of data) as a *training set* and the 2016 data as a *testing set*. The steps below explain our approach to building LSTM for the targeted energy demand prediction:

• **Feature Selection:** The LSTM model, designed in this paper, makes use of the *hourly energy usage* data and *weather temperature* information. In addition, it takes into account the attributes like *hour of the day*, *weekday/weekend* indicator, and *month* for more accurate model building.

- **Model Architecture:** Our LSTM model architecture comprises of a stack of 2 LSTM layers. Each layer having 8 and 16 units, respectively. The input data is represented as a 3D input vector in the format [samples, timesteps, features]. The N outputs from LSTM layers are the input to a fully connected (dense) layer with linear activation. We insert a dropout layer in between the LSTM layers to randomly drop 20% of the values to avoid overfitting the model with training data. The output of the dense layer becomes the prediction our model generates. The network uses a *Mean Absolute Percentage Error* (MAPE) loss function. It is one of the two parameters required to compile a model. The other parameter is the optimization algorithm, and we use the Adam optimizer (as it achieves better and faster results). It has the combined advantages of both *Adaptive Gradient* algorithm (AdaGrad) and *Root Mean Square Propagation* (RMSProp).

VI. Automated Model Selection Metrics: Accuracy, Compute Requirements, and Runtime

We implemented and evaluated the three designed models using the following computing environments:

- **Lenovo ThinkPad T440S laptop**, with Intel(R) i7-4600U CPU @ 2.10GHz, 2 Core(s), 12 GB RAM (for *Regression* and *ARIMA models*).
- **Dell Precision T1700 workstation**, with Intel(R) Xeon(R) CPU E3-1240 v3 @ 3.40GHz, 4 Core(s), 16 GB of RAM, with NVIDIA GeForce GTX 1080 GPU, 8.4 GB GPU memory (for the *LSTM model*).

To formally evaluate the prediction accuracy of the designed models, we compare the predicted energy usage value E_n^{pred} with the true, measured value E_n^{measrd}.

We compute the following two metrics:

$$prediction\_error = \frac{1}{N}\Sigma_{n=1}^{N}\frac{|E_n^{measrd} - E_n^{pred}|}{E_n^{measrd}}$$

$$RMSE = \sqrt{\frac{1}{N}\Sigma_{n=1}^{N}\left(E_n^{measrd} - E_n^{pred}\right)^2}$$

where N is the total number of observations in a dataset.

We use two sets of metrics for evaluating the constructed piecewise regression model: *(i) the accuracy of the fit*, i.e., the modeling errors on the training dataset (2015 year), and *(ii) the accuracy of the future prediction*, i.e., the prediction errors on the testing dataset (2016 year).

Table I shows the prediction errors and RMSE of the constructed piecewise linear regression model on the training and testing datasets. The modeling results are quite accurate: with mean prediction errors of 12% (for the model fit) and 19% (for the future forecast) and small RMSE values. Note, that the RMSE measure highly depends on the results' numerical values. For months with colder temperature, the energy usage values are higher, resulting in slightly higher RMSE.

For designed ARIMA and LSTM models, we perform a multistep rolling forecast to determine the energy usage in the next *24 hours*. This was implemented as a sliding window rolling forecast for the entire year of 2016, predicting the usage for the next *24 hours* as a *multistep* forecast.

Table II shows the prediction errors seen with different multistep predictions. The 24-hour forecast prediction error indicates the mean error calculated for all the 24-hour forecasts made for an entire year of 2016. Similarly, 1, 2 and 10-hour forecast errors summarizes the respective multistep forecast errors over 2016.

TABLE I
Piecewise Linear Regression Model Accuracy, Prediction Errors, and 95% Confidence Interval

Month	Model Performance Metrics				
	Training Prediction Error	Testing Prediction Error	Training RMSE	Testing RMSE	95% Confidence Interval
Jan	0.05	0.12	0.17	0.26	±0.006
Feb	0.05	0.11	0.18	0.25	±0.007
Mar	0.06	0.16	0.17	0.26	±0.01
Apr	0.13	0.18	0.21	0.24	±0.01
May	0.18	0.23	0.13	0.23	±0.01
Jun	0.19	0.24	0.12	0.12	±0.02
Jul	0.14	0.24	0.08	0.11	±0.01
Aug	0.13	0.25	0.08	0.11	±0.01
Sep	0.18	0.30	0.11	0.13	±0.02
Oct	0.15	0.23	0.18	0.21	±0.02
Nov	0.09	0.12	0.14	0.18	±0.008
Dec	0.07	0.14	0.14	0.44	±0.01
Mean	**0.12**	**0.19**	**0.14**	**0.22**	±0.004

TABLE II
ARIMA and LSTM Models Accuracy, Prediction Errors, and 95% Confidence Interval

Forecast (hours)	Mean Predict. Err.		Mean RMSE		95% Conf. Int.	
	ARIMA	LSTM	ARIMA	LSTM	ARIMA	LSTM
1	0.10	0.081	0.13	0.11	±0.019	±0.002
2	0.103	0.082	0.138	0.12	±0.002	±0.001
10	0.154	0.118	0.207	0.16	±0.003	±0.001
24	0.164	0.134	0.242	0.21	±0.003	±0.001

Both the ARIMA and LSTM models do very well for a short term forecast of 1 or 2 steps (hours) ahead. With the increased number of steps, the errors get higher for both methods. Overall, LSTM has a higher accuracy forecast (19% improvements) compared to ARIMA.

Table III shows the mean prediction error for all three models and selected months in 2016.

TABLE III
Three Models: Forecast Accuracy for Selected Months

Forecasted Month (2016)	Mean Prediction Error		
	Piecewise Regression	ARIMA (24 hours)	LSTM (24 hours)
February	0.11	0.17	0.12
July	0.24	0.16	0.13
November	0.12	0.15	0.12

For February and November (colder months, with high variance in temperature) the linear regression accuracy is similar or slightly better than LSTM results. Both of these models take the weather temperature as a critical feature while ARIMA is missing on this information and performing worse. However, during July, when temperature is not well correlated with energy usage, the linear regression model results are worst out of the three models.

We analyzed the **runtime requirements** of the models. The piecewise regression (being a simpler model) takes only **2 min** to forecast the energy usage for all the hours in 2016. Due to rolling forecast nature of the ARIMA model, we retrain the model after each multistep forecast is made, and hence the time taken for the entire 2016 forecast is **125 min**. For LSTM it takes longer time to train/build the model (depending on the number of epochs), and it requires the server with GPU. In our case, it takes about 10-13 secs per epoch to train the model with the 2015 data (i.e., **45 min** for 200 epochs), and the hourly predictions for the entire 2016 takes **5 sec**.

When approaching a forecasting problem, there are many different algorithms to choose from, and the *best model* selection could be defined by a variety of the user performance objectives and requirements, such as

- Model accuracy,
- Model simplicity,
- Compute requirements,
- Training time,
- Forecasting time period,
- Rolling forecasting period, etc.

Consider the parameters (discussed above) as a guidance for a model selection. If the user is looking for a season specific model, say a model that is simple, explainable, and works well for winter months, a Linear Regression should be used. On the other hand, if the user prefers the model accuracy over compute requirements and complexity, and is looking for a model that works well for all seasons, then LSTM model will be a winning candidate. If the user is looking for a short term forecasting model, that is simple and efficient and works decently well for all the seasons, then ARIMA model will be chosen. Therefore, the user can specify a set of metrics of interest to navigate the automated choice of the best forecasting model over time.

VII. Related Work

Energy demand forecasting has been broadly studied due to the problem importance and its significance for utility companies.

Statistical methods use historical data to predict energy consumption as a function of most significant variables. The detailed survey on regression analysis for prediction of residential energy consumption is offered in [11]. The authors believe that among statistical models, linear regression analysis has shown promising results because of satisfactory accuracy and simpler implementation compared to other methods. In many cases, the choice of the framework and the modeling efforts are driven by the specifics of the problem formulation. In [12], linear regression is used to predict the country annual energy use as a function of GDP, GDP per capita, population, population growth, and industrial growth rate. Our piecewise linear regression approach differs from the described above: we automatically identify the time-related (linear) daily usage patterns as well as apply seasonal sub-modeling.

While different studies have shown that weather variables and electricity demand and can be used in multivariate modeling, the univariate methods like ARMA and ARIMA [4], [5] might be sufficient for short term forecast.

Machine learning (ML) and *artificial intelligence* (AI) methods based on neural networks [6], [7], support vector machines SVM) [13], and fuzzy logic [14] were applied to capture complex non-linear relationships between inputs and outputs. When comparing ARIMA and artificial neural networks (ANN) modeling, some recent articles provide contradictory results. In [15], ARIMA achieves better results than ANN, while the study [16] claims that ANN perform slightly better than ARIMA methods. In our work, we construct the LSTM-based forecasting model (a special memory-based extension of RNN). The LSTM model works well for a short term (daily) demand prediction, and indeed, automatically captures non-linear daily patterns. In our work, we pursue an additional effort to automate and tune model parameter settings, as well as to navigate the best model selection according to the user requirements.

VIII. Conclusion and Future Work

In this work, driven by a variety of modeling and forecasting needs, we design, build, evaluate, and compare three models: (i) a piecewise linear regression model (ii) the univariate seasonal ARIMA model, and (iii) the multivariate LSTM model. We introduce elements of automation in each of these models to take away the burden of manual and tedious modeling tasks. The three constructed models have different computing resource requirements and training times, which need to be taken into account in guiding the user model choice for energy demand forecasting tasks.

In the future work, we plan to apply our models to more diverse datasets by enhancing the models with a clustering support phase. By clustering the similar usage building/sectors, we will enable the automated parameter tuning and selection of the models applied to the created data clusters.

References

[1] Smart Grid, Wikipedia, https://en.wikipedia.org/wiki/Smart_grid

[2] Advanced metering infrastructure (AMI) https://www.smartgrid.gov/recovery_act/deployment_status/sdgp_ami_systems.html

[3] Smart meter. https://en.wikipedia.org/wiki/Smart_meter

[4] J. Nowicka-Zagrajeka, R. Weron. Modeling electricityloads in California: ARMA models with hyperbolic noise. Signal Processing, vol 82, Elsevier, 2002.

[5] C. Nichiforov, I. Stamatescu, I. Făgărăşan, G. Stamatescu: Energy consumption forecasting using ARIMA and neural network models. 5th Intl. Symp. on Electrical and Electronics Engineering (ISEEE), 2017.

[6] S.M. Sulaiman ; P. Aruna Jeyanthy ; D. Devaraj: Artificial neural network based day ahead load forecasting using Smart Meter data. Biennial Intl. Conf. on Power and Energy Systems: Towards Sustainable Energy (PESTSE), 2016.

[7] Hu Y.-C.: Electricity consumption prediction using a neural-network-based grey forecasting approach. J. of the Operational Research Society, v. 68, 2017.

[8] UMass Trace Repository, Apartment dataset. http://traces.cs.umass.edu/index.php/Smart/Smart.

[9] P. Amin, L.Cherkasova, R. Aitken, V. Kache: Analysis and Demand Forecasting of Residential Energy Consumption at Multiple Time Scales. IFIP/IEEE Integrated Network Management Symposium (IM'2019).

[10] Hochreiter and J. Schmidhuber, Long short-term memory. J. Neural Computation, vol. 9, no. 8, 1997.

[11] N. Fumo, M. A. Rafe Biswas: Regression analysis for prediction of residential energy consumption. J. Renew. Sustain. Energy Rev., vol. 31, Elsevier, 2014.

[12] S. Tabazi, A. Aslani, H. Forotan: Prediction of Energy Consumption by Using Regression Model. Comp. Research Progress in Appl. Science & Eng., vol.2, 2016.

[13] Qiu, Z.: Electricity consumption prediction based on data mining techniques with particle swarm optimization. Intl. J. of Database Theory and Application 6(5), 2013.

[14] K. Song, at al. Short-term load forecasting for the holidays using fuzzy linear regression method. IEEE Transactions on Power Systems, vol.20, no.1, 2005.

[15] Høverstad, B., Tidemann, A., Langseth, H.: Effects of Data Cleansing on Load Prediction Algorithms. IEEE Symp. on Comp. Intelligence 2013.

[16] Gerwig, C.: Short Term Load Forecasting for Residential Buildings - an Extensive Literature Review. Proc.of Intl. Conf on Intelligent DecisionTechnologies, 2017.