UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Gunay Abdullayeva

Application and Evaluation of LSTM Architectures for Energy Time-Series Forecasting

Master's Thesis (30 ECTS)

Supervisors: Alan Henry Tkaczyk, Ph.D.
Meelis Kull, Ph.D.
Nicolas Kuhaupt, M.Sc.

Tartu 2019

Application and Evaluation of LSTM Architectures for Energy Time-Series Forecasting

Abstract: Accurate energy forecasting is a very active research field as reliable information about future electricity generation allows for the safe operation of the power grid and helps to minimize excessive electricity production. As Recurrent Neural Networks outperform most machine learning approaches in time series forecasting, they became widely used models for energy forecasting problems. In this work, the Persistence forecast and ARIMA model as baseline methods and the long short-term memory (LSTM)-based neural networks with various configurations are constructed to implement multi-step energy forecasting. The presented work investigates three LSTM based architectures: i) Standard LSTM, ii) Stack LSTM and iii) Sequence to Sequence LSTM architecture. Univariate and multivariate learning problems are investigated with each of these LSTM architectures. The LSTM models are implemented on six different time series which are taken from publicly available data. Overall, six LSTM models are trained for each time series. The performance of the LSTM models is measured by five different evaluation metrics. Considering the results of all the evaluation metrics, the robustness of the LSTM models is estimated over six time series.

Keywords:
Neural Networks, ARIMA, Persistence forecast, long short-term memory, Standard LSTM, Stack LSTM, Sequence to Sequence LSTM, univariate time series forecasting, multivariate time series forecasting, energy forecasting.

CERCS: P170 Computer science, numerical analysis, systems, control

LSTM-arhitektuuride rakendamine ja hindamine energia aegridade prognoosimiseks

Lühikokkuvõte: Täpsete prognooside koostamine on energiavaldkonnas väga aktiivne uurimisvaldkond, kuna usaldusväärne teave tulevase elektritootmise kohta on oluline elektrivõrgu ohutuse tagamisel ning aitab minimeerida liigset elektrienergia tootmist. Kuna rekurrentsed tehisnärvivõrgud ületavad aegridade prognoosimise täpsuses enamikke muid masinõppe meetodeid, siis on need võetud ka energia prognoosimisel laialdaselt kasutusele. Käesolevas töös on energiaprognooside tegemiseks rakendatud algoritme Persistence ja ARIMA baasmeetoditena ning pika lühiajalise mäluga (LSTM) tehisnärvivõrke erinevates konfiguratsioonides. Töö uurib kolme LSTM-põhist arhitektuuri: i) standardne LSTM, ii) kahekihiline (*stacked*) LSTM ja iii) jadast-jadasse (*sequence to sequence*) LSTM. Kõigi nende LSTM-arhitektuuridega uuritakse nii ühemõõtmelisi kui ka mitmemõõtmelisi õpiülesandeid. LSTM-mudeleid treenitakse kuue erineva avalikult kättesaadava aegrea ennustamiseks, kusjuures iga aegrea jaoks treenitakse kuus erinevat LSTM mudelit. LSTM-mudelite poolt tehtud ennustusi mõõdetakse viie erineva hindamismõõdikuga. Lähtuvalt hindamise tulemustest neil kuuel aegreal hinnatakse

LSTM-mudelite arhitektuuride robustsust.

Võtmesõnad:
Tehisnärvivõrgud, ARIMA, Persistence, pika lühiajalise mäluga võrgud (LSTM), standardne LSTM, kihiline LSTM, jadast-jadasse LSTM, ühemõõtmeline aegrea prognoosimine, mitmemõõtmeline aegrea prognoosimine, energia prognoosimine.

CERCS:P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

List of Abbreviation

RES	Renewable Energy Sources
ARIMA	Auto-Regressive Integrated Moving Average
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
RMSE	Root mean squared error
MAE	Mean absolute error
SMAPE	Symmetric mean absolute percentage error
S2S	Sequence to Sequence
GA	Genetic Algorithm
BPTT	Back Propagation Through Time

Contents

1 Introduction **6**
- 1.1 Problem Statement . . . 7
- 1.2 Study Outline . . . 8

2 Related Work **9**

3 Methodology **10**
- 3.1 Forecasting Models . . . 10
  - 3.1.1 Multi-step Forecasting Model . . . 10
  - 3.1.2 Multi-step Univariate Forecasting Model . . . 10
  - 3.1.3 Multi-step Multivariate Forecasting Model . . . 11
- 3.2 Baseline methods . . . 11
  - 3.2.1 Modelling Forecast Using Persistence . . . 11
  - 3.2.2 ARIMA Based Time Series Forecasting . . . 11
- 3.3 Time series forecasting using LSTM architectures . . . 13
  - 3.3.1 Recurrent Neural Network . . . 13
  - 3.3.2 General overview of LSTM unit . . . 14
  - 3.3.3 LSTM Model Architectures . . . 15
- 3.4 Evaluation metrics . . . 17

4 Experiments and Results **19**
- 4.1 Datasets . . . 19
- 4.2 Dataset preparation . . . 26
  - 4.2.1 Dataset preparation for ARIMA . . . 27
  - 4.2.2 Dataset preparation for LSTM model . . . 27
  - 4.2.3 Dataset Preparation for Persistence . . . 29
- 4.3 Experiments . . . 29
  - 4.3.1 ARIMA Model Formulation . . . 29
  - 4.3.2 LSTM Models Formulation . . . 30
  - 4.3.3 First Experimental Results . . . 33
  - 4.3.4 Experimental results with Early Stopping . . . 34
  - 4.3.5 Experimental results with Parameter tuning . . . 35
- 4.4 Prediction results for time series . . . 41
  - 4.4.1 Predictions for Open System Load Consumption . . . 42
  - 4.4.2 Predictions for Beijing PM2.5 . . . 44
  - 4.4.3 Predictions for Driven Hourly and 15 minute Energy Consumption 46
  - 4.4.4 Predictions for Energy Consumption of Appliances and Light . 50

5 Conclusion **54**

1 Introduction

Over the last decade, different projects have been undertaken in the field of energy forecasting. The European Union Horizon 2020 EU-SysFlex project is one such effort to identify issues and solutions associated with integrating large-scale renewable energy and create a plan to provide practical assistance to power system operators across Europe [1]. To provide an accurate plan for power system operators, an important aspect to consider is accurate energy forecasting. Accurate energy forecasting enables efficient operation of power systems, preservation of the balance between supply and demand, reduction of production cost, and management of future capacity planning [2]. Forecasting of load and price of electricity, fossil fuels (natural gas, oil, coal) and renewable energy sources (RES; hydro, wind, solar) are included in the energy forecasting [3]. Energy forecasting is grouped into three categories depending on the forecast duration [4]: short-term, medium term and long-term. Typical definitions are as follows: the short-term forecast ranges between one hour and one week, medium-term forecast ranges between one week and one year, and long-term forecasts span a time of more than a year. Short-term forecasts serve for deciding on the use of power plants, optimization of the scheduling of power systems, economic dispatch and electricity market [3]. Medium and long-term forecasting is important for planning of building future sites or determining fuel sources of power plants [3].

Forecasting problems are divided into a single-step and multi-step forecasting depending on the future forecast steps [2]. In single-step and multi-step forecasting problems, one step and multi-step ahead predictions are solved, respectively. The current work is focused on multi-step short-term energy forecasting. Two learning problems are investigated to implement multi-step forecasting: univariate and multivariate. Univariate time-series forecasting is a problem comprised of one single series where the model learns from the past values to predict the next values of the sequence. The difference between univariate and multivariate forecasting problem is that multivariate models use multiple input series for prediction.

Various machine learning methods have been considered for forecasting; these are divided into traditional statistical techniques and deep learning based approaches. In this work, the statistical model Auto-Regressive Integrated Moving Average (ARIMA), Persistence forecast and deep learning methods with different configurations are implemented for time-series forecasting. The ARIMA model and the Persistence forecast results are considered as a baseline.

ARIMA is a statistical model used for analyzing and forecasting time series data. An ARIMA model considers the dependent relationship between an observation and past

values along with the error in the forecasting. In previous work, it was found that the ARIMA model works better for linear and stationary time series data [5]. For short-term forecasting, the ARIMA model has been applied by several researchers [6, 7].

The Persistence forecast is used to generate baseline results for time series forecasting problems [8]. In the multi-step time series forecasting problem, the Persistence forecast uses the previous time steps to predict an expected outcome for the next time steps.

Recently, considering complex non-linear patterns and large amounts of data, different deep learning techniques have been applied to time series forecasting problems due to their ability to capture data behavior. [9, 10]. Recurrent Neural Network is one type of deep learning allows the use of multiple layers and helps to learn different feature representations in data. Recurrent Neural Networks (RNN) allow learning patterns in sequential data such as video, speech and time series. In this work, long short-term memory (LSTM)-based neural network is used, which is a variation of RNN and performs considerable results for time series forecasting problems. Different LSTM architectures are implemented and evaluated on publicly available data.

1.1 Problem Statement

The aim of this study is to explore different LSTM architectures over six different time series and determine robust LSTM architectures for energy time series forecasting problem. The LSTM architecture is called robust when the model does not necessarily always have the best results for each time series, but it should not be much worse than the best. The results of robust LSTM architectures should perform always better than baseline results. The presented work investigates three variations of LSTM: i) Standard LSTM, ii) Stacked LSTM and iii) LSTM based Sequence to Sequence (S2S) architecture. Both univariate and multivariate forecasting problems are explored for each time series and LSTM architecture. For baseline methods, only the univariate forecasting problem has been learned.

The performances of baseline methods and LSTM models are measured by using five evaluation metrics: root mean squared error (RMSE), mean absolute error (MAE), symmetric mean absolute percentage error (SMAPE), bias, and correlation function. These metrics help to explore the errors from different aspects. While deciding robust LSTM architectures, all these evaluation metrics should be considered over all time series.

1.2 Study Outline

This thesis consists of five chapters in total. The structure of this thesis is given below:

– Section 2 highlights related work in the area of energy forecasting.
– Section 3 describes the forecasting methods, baseline approaches and LSTM architectures, and the chosen evaluation metrics.
– Section 4 introduces the datasets, explains dataset preprocessing steps and discusses the results from the experiments.
– Section 5 summarizes the key findings of this work.

2 Related Work

The availability of a relatively large amount of energy data allows using different Artificial Intelligence (AI) methods. A lot of work has been done in the area of short-term energy forecasting [2]-[4]. Noticeably, Recurrent Neural Network (RNN) was widely used in this research field as it is able to capture model nonlinearity. As RNN models outperformed statistical machine learning models, autoregressive and moving-average models have remained the baseline methods [12].

Daniel et al. [4] presented two univariate models namely, ARIMA and a Standard LSTM for energy load forecasting. The results showed that LSTM model outperformed ARIMA model in multi-step short-term load forecasting.

Various LSTM approaches have been implemented for univariate one-step ahead PV forecasting [11]. Four different Standard LSTM and Stack LSTM models were applied for PV forecasting using two various datasets. Standard LSTM models differ in using various lags of previous time steps (one time step, three time steps, time steps as features) and memory between batches. Stack LSTM model was built using two LSTM hidden layer. The comparative analysis revealed that the Standard LSTM model with used the lag three time steps had the best results for both datasets.

Shamsul et al. [2] presented the work for multivariate energy load forecasting. In this work, Standard LSTM and Sequence to Sequence LSTM models results were investigated for one-minute and one-hour time step resolution data. The results showed that the standard LSTM architecture failed on load forecasting using one-minute resolution, the S2S LSTM architecture performed well in both datasets.

Different machine learning approaches covering linear regression, k-nearest neighbors, random forest, gradient boosting, ANN and extra tree regressor were also applied for short-term electric load forecasting [12]. Salah et al. [12] presented a load forecasting methodology using classical machine learning methods and LSTM network. The classical machine learning models were trained for multivariate load forecasting, in turn, LSTM network was trained for univariate load forecasting. Genetic algorithm (GA) was used to find out optimal hyper-parameters such as the length of window size, the number of hidden units and the number of hidden layers. The results showed that LSTM network with optimal hyper-parameters performed better than classical machine learning models and it had stable results for both short and medium-term load forecasting.

3 Methodology

In the first section, we describe the general forecasting approach for multi-step univariate and multivariate forecasting models. In Subsection 3.2, we introduce the baseline methods and explain why we selected them. Subsection 3.3.1 explains the LSTM architectures which were chosen for this study. Lastly, the evaluation metrics are presented and explained how they evaluated the predictions differently.

3.1 Forecasting Models

In this study, we consider both multi-step univariate and multi-step multivariate forecasting techniques for energy forecasting. In this section, we discuss what is the multi-step forecasting model, and how to implement multi-step forecasting for univariate and multivariate forecasting problems. We also give a general understanding of univariate and multivariate forecasting models, explain the difference between them.

3.1.1 Multi-step Forecasting Model

For real forecasting problems, the main objective is not only to predict a value ahead in time but also a certain time forecast horizon k. The forecast horizon is the span of time into the future for which forecasts should be prepared. If the forecast horizon k is bigger than one, this kind of forecasting is called multi-step forecasting and can be implemented using two strategies [14]: i) the direct strategy - by explicitly training a model to predict several steps ahead, or ii) the iterative method - by doing repeated one-step ahead predictions up to the desired horizon. In this study, the forecast horizon is defined as future 36 hours time steps and the direct strategy is applied for this multi-step ahead prediction.

3.1.2 Multi-step Univariate Forecasting Model

In "classical time series", it is assumed that the following series members depend only on a certain amount of its direct predecessors [14]. In this case, the forecasting problem is comprised of one single series and called univariate forecasting problem. Suppose we have historical data for some time series given like x_1, ... , x_{n-1}, x_n. As there exist some functional dependency between historical and future time series data points, the forecast values x'_{n+1}, x'_{n+1}, ... , x'_{n+k} for the k forecast horizon are a function of the preceding n data points. This dependency is described in Eq. 1.

$$x'_{n+1}, x'_{n+2}, ..., x'_{n+k-1}, x'_{n+k}, = f(x_1, ..., x_{n-1}, x_n) \tag{1}$$

Here f might be any machine learning method. As a machine learning tool, the baseline approaches and LSTM models are applied for the current forecasting problem.

3.1.3 Multi-step Multivariate Forecasting Model

The multivariate forecasting model is an extended version of the univariate forecasting model where the only difference is that future time series values not only depend on the preceding values of the same series, but also the values of another time series. Suppose we have the historical data for the time series as x_1, ... , x_{n-1}, x_n and another time series as y_1, ... , y_{n-1}, y_n and there is a functional dependency among their members. The task is to predict the future k values of the time series, which is x'_{n+1}, x'_{n+1}, ... , x'_{n+k}. According to the multi-step multivariate forecast model, these future values are predicted using the Eq. 2. Similar to the univariate model, f could be any machine learning method.

$$x'_{n+1}, x'_{n+2}, ..., x'_{n+k-1}, x'_{n+k}, = f(x_1, y_1, ..., x_{n-1}, y_{n-1}, x_n, y_n) \tag{2}$$

3.2 Baseline methods

In this study, due to their simplicity, the Persistence forecast and the ARIMA statistical model are used as baseline approaches which provide a point of comparison with LSTM architectures. The only univariate forecasting problem is explored with baseline methods.

3.2.1 Modelling Forecast Using Persistence

The Persistence forecast is a common reference model for the time series forecasting as it provides a computationally inexpensive forecast [8]. Persistence introduces the concept of "memory". The algorithm uses the value at the previous time step t to predict the expected outcome at the next time step $t + 1$. That is why this model gives better results for the stationary time series. The performance of the persistence model depends on the forecast horizon. The uncertainty for the future time steps is getting bigger when the large forecast horizon is used.
The forecast technique of the Persistence forecast is described in Figure 1. To forecast the next 36 hours, the last 24 hours of the historical data are used. Firstly, the next 24 hours are forecasted using the last 24 hours of the available data. Second, the data between 12h and 24h are used to fulfill the next 12 hours forecast points. The forecasting strategy is interpreted in Eq. 3. In the equation, $x'_{n+1}, x'_{n+2}, ..., x'_{n+36}$ are the forecast and $x_{n-24}, x_{n-23}, ..., x_{n-1}$ are input data points.

$$x'_{n+1}, x'_{n+2}, ..., x'_{n+36} = (x_{n-24}, x_{n-23}, ..., x_{n-1}, x_{n-24}, x_{n-23}, ..., x_{n-13}) \tag{3}$$

3.2.2 ARIMA Based Time Series Forecasting

ARIMA is the acronym for Auto Regressive Integrated Moving Average where each component has a key characteristic [2]: AR (Autoregression), relying on a dependent

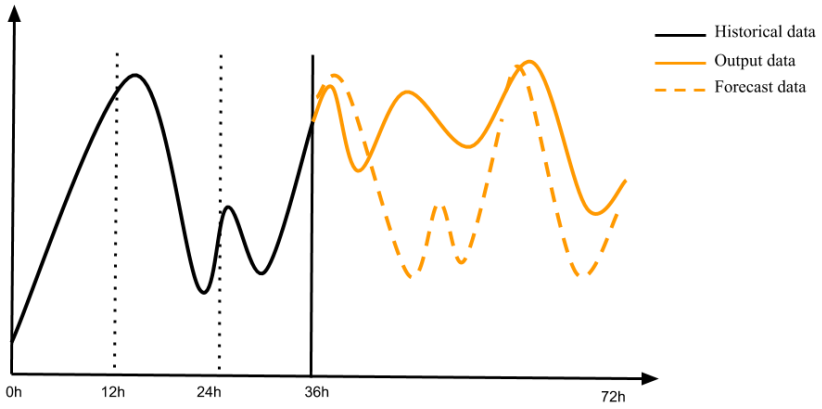


Figure 1. The forecast strategy of the Persistence forecast for the next 36 hours.

relationship between an observation and some number of lagged observations; I (Integrated), the number of differences of actual observations, needed to make the time series stationarity; and MA (Moving Average), the number of lagged forecast errors in the prediction equation.
These components are introduced in an ARIMA model as a set of parameters given as ARIMA*(p,d,q)*: p is the number of lag observations, d is the number of times that the actual observations are differenced and q is the size of the moving average window.
The Auto Regressive model is shown in Eq. 4 where y_t depends only on its own lags [2].

$$y_t = \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + ... + \beta_p y_{t-p} \tag{4}$$

y_t is the current measured values at time t; α and β_i are coefficients; and p is the autoregressive component .
In the Moving Average (Eq. 5), y_t depends only on its lagged forecast errors [2].

$$y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_p \epsilon_{t-p} \tag{5}$$

y_t is the current measured values at time t; ϵ_t is the forecast error at time t, θ_i are coefficients; and q is the moving average component.
As the ARMA model is the combination of the AR and MA terms, it is represented as a formula in Eq. 6.

$$y_t = \alpha + \beta_{1t-1} + \beta_2 y_{t-2} + ... + \beta_p y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_p \epsilon_{t-p} \tag{6}$$

In the case of non-stationary time series, a transformation of the series is presented by Box and Jenkins to make it stationary and it results the ARIMA model [13]. The

measured values y_t are replaced with the results of a recursive differencing process. The first order differencing can be described as Eq. 7 [13].

$$y_t = y_t - y_{t-1} \tag{7}$$

3.3 Time series forecasting using LSTM architectures

In this work, three different LSTM architectures are studied for the multi-step univariate and multivariate time series forecasting. In the subsections below, we briefly describe the simple Recurrent Neural Network (RNN) architecture, the LSTM unit as a variation of RNN and the proposed LSTM architectures.

3.3.1 Recurrent Neural Network

In a traditional neural network, inputs and outputs are considered as independent of each other. As the sequential pattern exists in time series data, such a neural network does not give efficient results for the time series forecasting. As an alternative network, RNN is more effective to learn the dependency between observations. It has been proved that RNN shows considerable results for time series forecasting [19]. The simple architecture and the unrolled version of RNN is shown in Figure 2 [18].

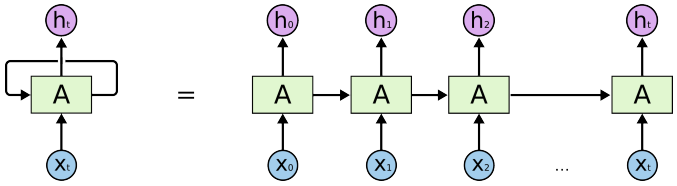


Figure 2. An unrolled recurrent neural network [18].

The simple RNN is a network with loops which allows persisting information to be passed from one step of the network to the next. This looping process can be unrolled as described in Figure 2. The process is illustrated for the time steps from $0, 1, 2$ up to time t: $x_0, x_1, x_2, ..., x_t$ are the inputs, A is the hidden state, and $h_0, h_1, h_2, ..., h_t$ are the outputs. A_t hidden state is an activation function (normally tanh) which takes its input from the hidden state of the previous step A_{t-1} and the output of the current step x_t . This process is described in Eq. 8.

$$A_t = f(A_{t-1}, x_t) \tag{8}$$

RNNs use backpropagation through time (BPTT) to optimize weights during training. BPTT uses the chain rule to go back from the latest time step to the previous steps and

the gradients tend to get smaller and smaller while moving backward in the network. That is why RNN has a vanishing gradient issue and it leads to the problem of learning the long-term dependencies. To solve this issue, as a variation of RNN, LSTM network was introduced by Hochreiter & Schmidhuber [15].

3.3.2 General overview of LSTM unit

LSTM networks are specially designed to learn long term dependency problems. The traditional neural networks have neurons, in turn, LSTMs have memory cells that are connected through layers. Each memory cell contains gates which handle information flow into and out of the cell. There are three types of gates in the LSTM unit [2]: forget, input and output. The task of each gate is listed as follows:

• **Forget gate** forgets the irrelevant parts from the previous state.

• **Input gate** selectively updates the cell state values.

• **Output gate** outputs the certain part of cell state.

The structure of the LSTM unit is shown in Figure 3.

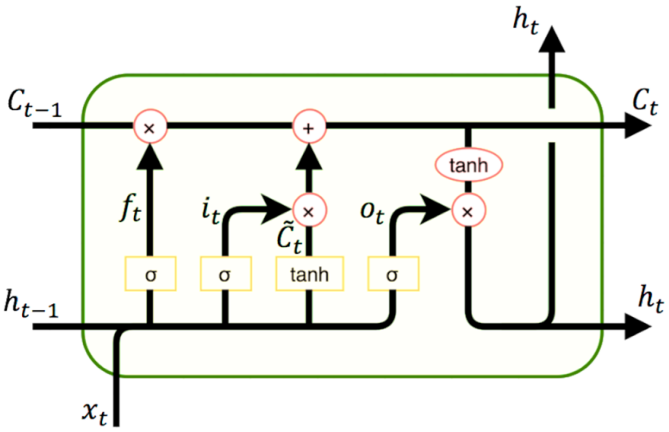


Figure 3. LSTM unit [17].

As seen from Figure 3 and Eq. 9 - 12, the LSTM unit gets the information from the previous state h_{t-1} and input x_t, and uses the activation functions to decide which part of the information to pass to the output and next LSTM unit.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{9}$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{10}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_0) \tag{11}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{12}$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \tag{13}$$

$$h_t = o_t \otimes tanh(C_t) \tag{14}$$

Eq. 9 - 11 describes three sigmoid functions ($\sigma(x) = \frac{1}{1+e^{-x}}$) where W's and b's are the parameters (weights and biases) for input, forget and output gates. f_t, i_t and o_t are input, forget and output gates respectively. In Eq. 13, the tanh layer creates the vector of new candidate value $\tilde{C}_t$ which is added to the cell state.
LSTM unit has two kinds of hidden states: "slow" state C_t and a "fast" state h_t. The slow state C_t is updated by summing the multiplication the forget gate f_t by the previous cell state C_{t-1} and the multiplication the input gate i_t by the new candidate value $\tilde{C}_t$. The h_t state is updated using the hyperbolic tangent function (tanh) of C_t state and o_t output gate.
The main preference of LSTM unit is that its cell state accumulates activities over time. As derivatives of the error are summed over time, they do not vanish quickly [18]. In this way, LSTMs can implement tasks over long sequences.

3.3.3 LSTM Model Architectures

In this work, we investigates three kinds of LSTM architectures: i) Standard LSTM, ii) Stack LSTM and iii) Sequence to Sequence (S2S) LSTM. Both Univariate and Multivariate forecasting problems are explored for each architecture. Each LSTM architecture is explained as follows:

Standard LSTM Architecture The network has one input layer, one hidden LSTM layer and an output layer. The architecture of the LSTM model is shown in Figure 4. $x_{t+1}, x_{t+2}, ..., x_{t+n}$ are the inputs. n defines the window size which determines how many previous values of the time series will be used during the training. Depending on the forecasting problem (univariate or multivariate), there might be one or multiple inputs for each LSTM cell. LSTM cells share the same amount of LSTM units. In this architecture, many to one LSTM model [16] is applied where the output is generated from the last LSTM cell. The output of the hidden LSTM layer is fully connected to the last layer which generates the next 36 hours forecast measures.

Stacked LSTM architecture This LSTM architecture makes a difference from the previous model using one more LSTM hidden layer (Figure 5). The configurations for the input and first hidden layer are the same as in the Standard LSTM architecture. However, in this architecture, each LSTM cell in the first hidden layer has its own output to pass

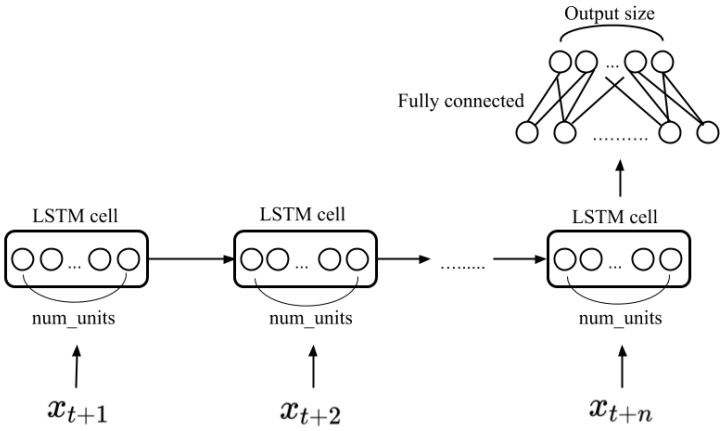


Figure 4. Standard LSTM architecture.

the information to the second hidden layer. The output of the second hidden LSTM layer is fully connected to the last layer which generates the next 36 hours forecast measures.

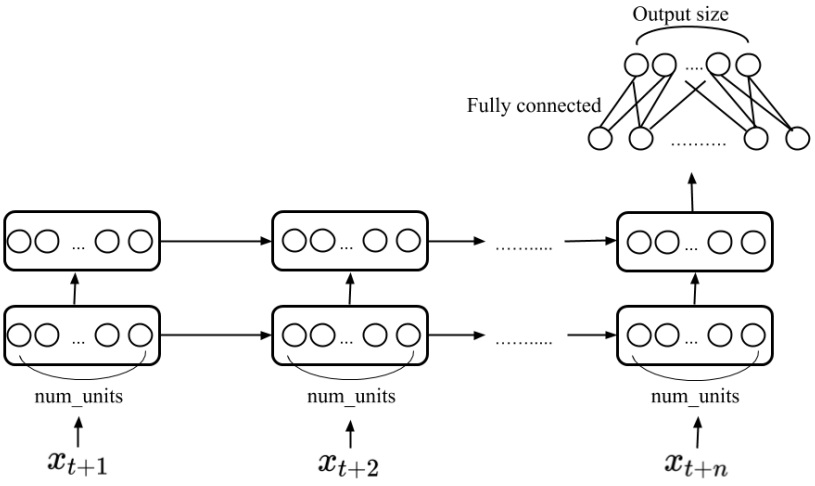


Figure 5. Stack LSTM architecture.

Sequence to Sequence LSTM architecture This architecture consists of two LSTM networks: encoder and decoder. The encoder holds the input series and encodes them in a fixed length vector, which is used as the hidden input state for the decoder (Figure

6). The decoder LSTM cell inputs are set to zero. The output is generated from decoder LSTM cell for each future time step.

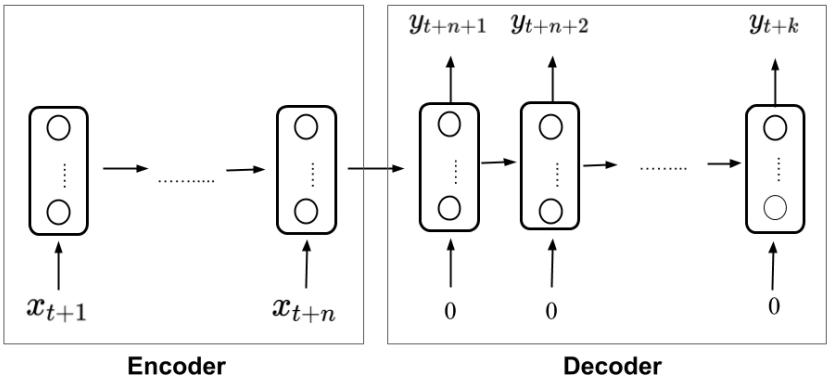


Figure 6. Sequence to Sequence LSTM architecture.

3.4 Evaluation metrics

Five evaluation metrics are used to measure the performance of the models (Eq. 15 - 19): the root-mean-square error (RMSE), the mean absolute error (MAE), the symmetric mean absolute percentage error (SMAPE), the BIAS, and the correlation function between the forecast and measured time series. The equations use x as a value of the measured and x' as a value of the forecast time series. Both time series have N samples.

$$RMSE(x', x) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x'_n - x_n)^2} \tag{15}$$

$$MAE(x', x) = \frac{1}{N}\sum_{n=1}^{N}|x'_n - x_n| \tag{16}$$

$$SMAPE(x', x) = \frac{100}{N}\sum_{n=1}^{N}\frac{|x'_n - x_n|}{|x'_n| + |x_n|} \tag{17}$$

$$BIAS(x', x) = \frac{1}{N}\sum_{n=1}^{N}(x'_n - x_n) \tag{18}$$

$$Correlation(x', x) = \frac{\sum_{n=1}^{N}(x'_n - \overline{x}'_n) * \sum_{n=1}^{N}(x_n - \overline{x}_n)}{\sqrt{\sum_{n=1}^{N}(x'_n - \overline{x}'_n)^2 * \sum_{n=1}^{N}(x_n - \overline{x}_n)^2}} \tag{19}$$

RMSE and MAE are one of the common metrics to measure the average error between forecast and actual values. The RMSE is more sensitive to the outliers in the data as it calculates the average of the squared errors. SMAPE interprets an average percentage error between 0% and 100%. The BIAS allows assessing whether the forecast is predicting higher or lower values than the actual value on average. Lastly, the Correlation measures the similarity of the behavior of the forecast and actual values.

4 Experiments and Results

In this section, we present the datasets and discuss the results of the experiments for the multi-step short-term energy forecasting. The planned path for this study is described in Figure 7. The proposed process can be seen as a framework of four processing components, namely, data preparation and pre-processing, the baseline models training, the LSTM models training and analyzing results. All these processing components are explained in the their own sections.

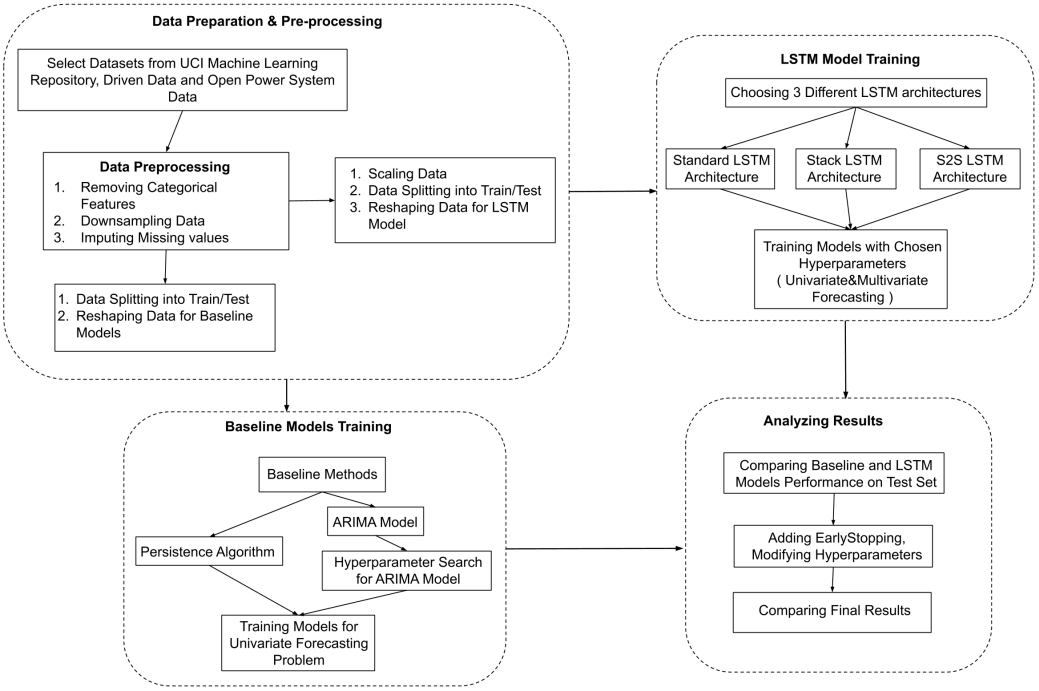


Figure 7. The planned path for the multi-step short-term energy forecasting.

4.1 Datasets

The datasets for this study were chosen from three different data sources: UCI Machine Learning Repository [20], Driven Data [21], and Open Power System Data [22]. We selected those datasets because they cover electricity and weather data, they had appropriate time resolution and multiple time series to consider for multivariate forecasting problem. In total, we worked with four different datasets. These datasets have different sampling rates (ex: one-minute, ten-minute, one-hour). For simplicity, we downsampled the time series with small frequency to fifteen-minute. As a result, we worked with fifteen minute and hourly sampled datasets.

The datasets from UCI Machine Learning Repository The two datasets were chosen from UCI Machine Learning Repository: i) Beijing PM2.5 dataset [23], ii) Appliances Energy Prediction dataset [24].

Beijing PM2.5 dataset This hourly dataset contains the PM2.5 (particle that affects air pollution) data of the US Embassy in Beijing and weather data from Beijing Capital International Airport. The dataset does not cover the electricity measurements, however is worked to investigate the performances of the models. In this study, PM2.5 data is used as a time series to forecast future values. The weather measurements (dew point, temperature, pressure, wind speed, cumulated hours of snow and rain) are used in multivariate forecasting. The dataset contains 2067 missing values for PM2.5. As PM2.5 starts with the missing value, these values are imputed using the next valid observation. The dataset ranges from 2010-01-01 to 2014-12-31. The first six months of PM2.5 as shown in Figure 8.

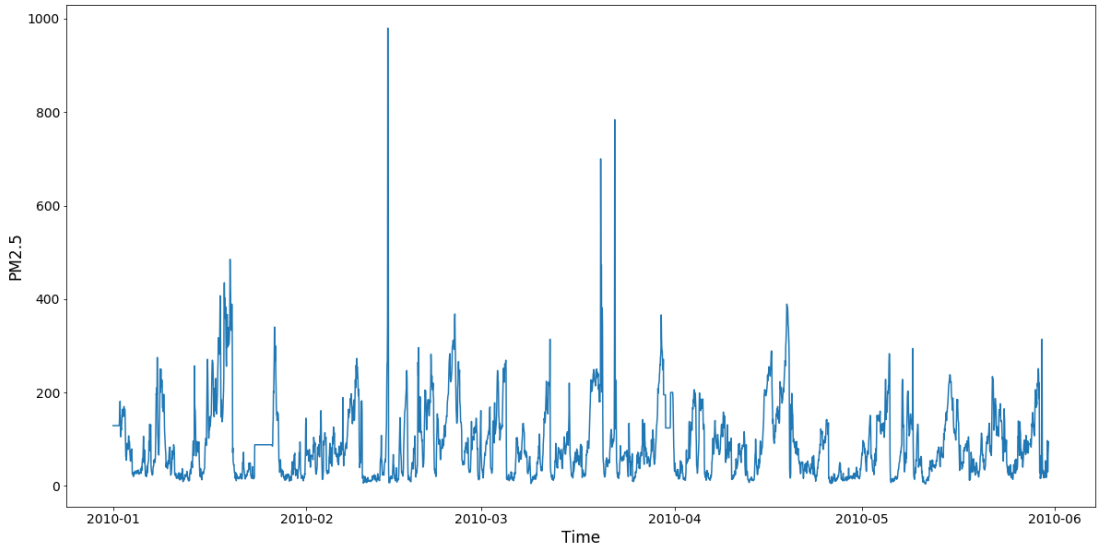


Figure 8. `PM2.5` for the first six months.

In Figure 9, the monthly, weekly and hourly behavior of the time series is described. Box plots of PM2.5 reveal that the average PM2.5 is almost constant across months (Figure 9a), weeks (Figure 9b) and quarters of days (Figure 9c). The quarters of days observe PM2.5 for the 04.00-10:00, 10:00-16:00, 16:00-22:00 and 22.00-04.00 time ranges.

Appliances Energy Prediction dataset This data set is at 10 min resolution for about 4.5 months from 2016-01-11 to 2016-05-27. The dataset includes the Energy Consumption of Appliances and Light fixtures in a house and weather information (humidity and

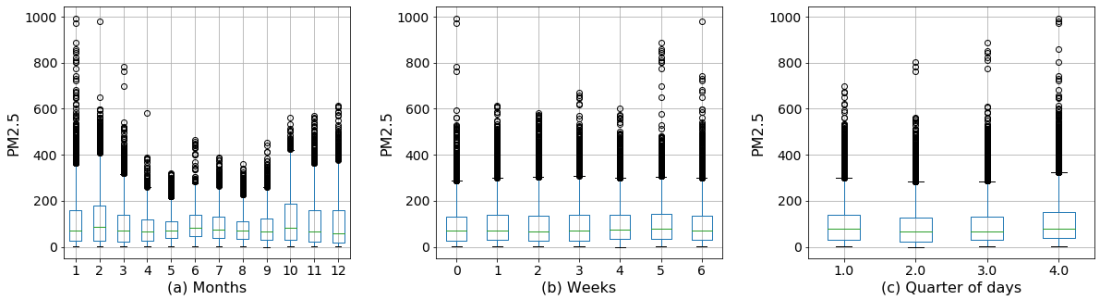


Figure 9. Monthly (a), Weekly (b), Quarterly (c) behavior of `PM2.5`.

temperature inside a house and temperature, pressure, humidity, wind speed, dew point outside). The frequency of the dataset is downsampled from 10 min to 15 minute. The forecast measurements are the Energy Consumption of Appliances and Light fixtures and the weather information is considered in the multivariate forecasting. The first thousand instances of Energy Consumption of Appliances and Light are shown in Figure 10 and Figure 11 respectively. As seen from the figures, both time series have drastic changing characteristics.

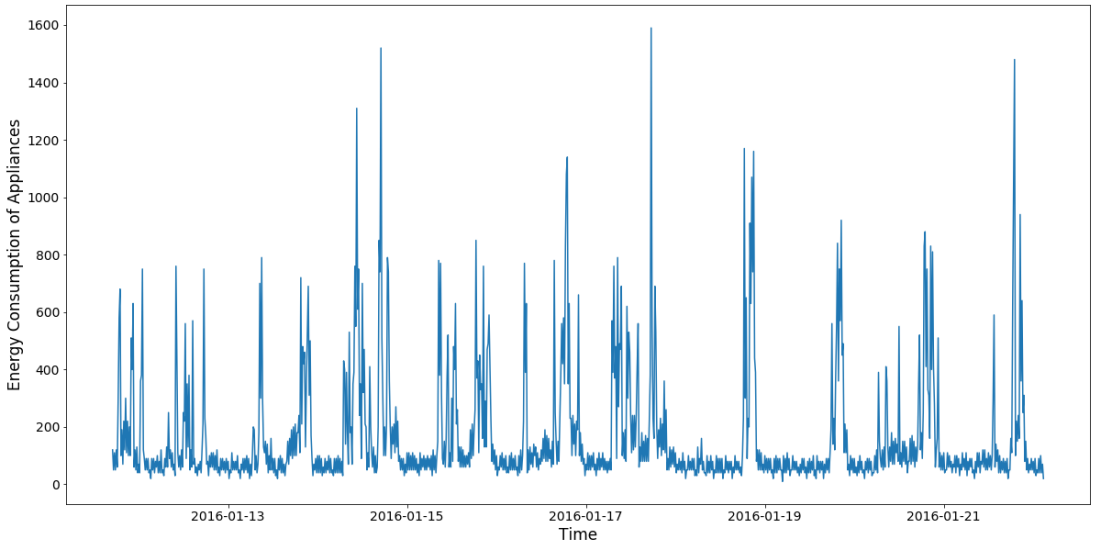


Figure 10. The first thousand instances of `Energy Consumption of Appliances`.

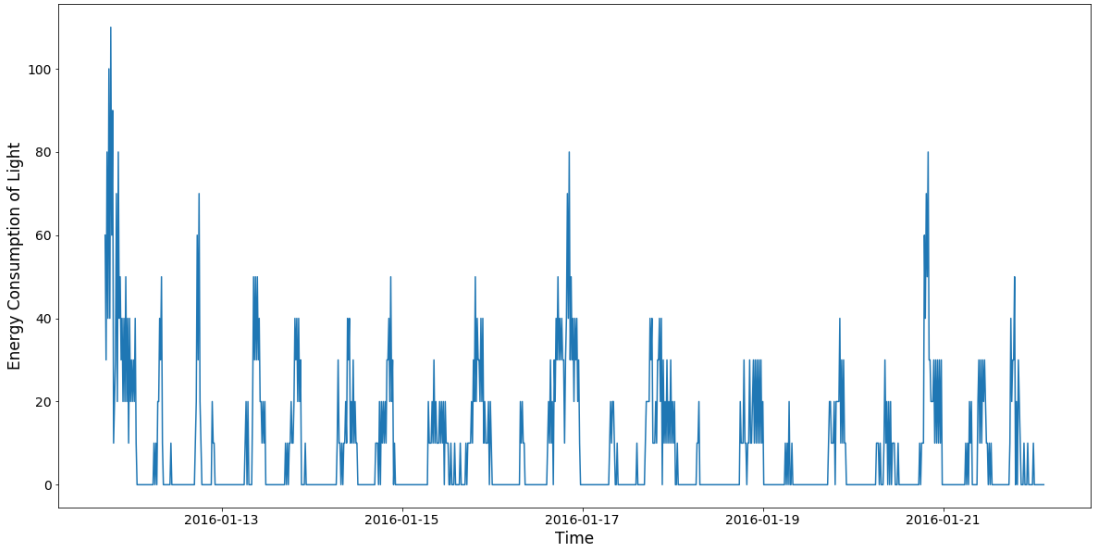


Figure 11. The first thousand instances of `Energy Consumption of Light`.

For both time series, the monthly, weekly and hourly behaviors are presented in Figure 12-13 respectively. Figure 12 depicts that the average Energy Consumption of Appliances are almost constant across months (Figure 12 (a)) and weeks (Figure 12 (b)), while quarterly plot (Figure 12 (c)) shows lower consumption in the first and last quarters.

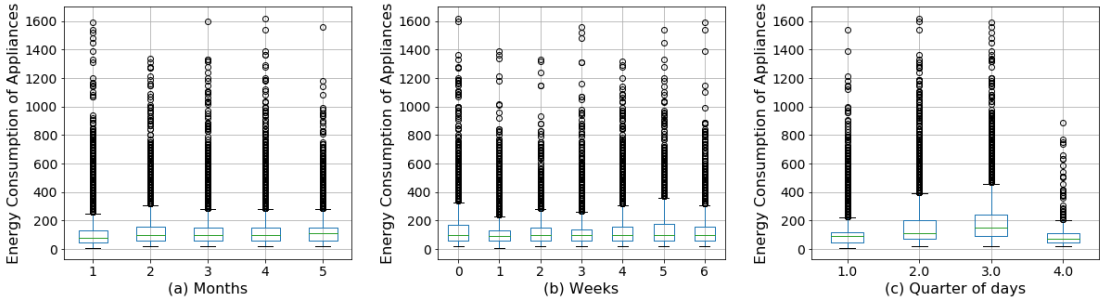


Figure 12. Monthly (a), Weekly (b), Quarterly (c) behavior of `Energy Consumption of Appliances`.

Figure 13 (a) reveals that the average Energy Consumption of Light is around zero and last two months have less consumption than first three months. Weekly Energy Consumption of Light (Figure 13 (b)) shows that there is less consumption on Friday and Saturday. Energy Consumption of Light increases in the third quarter due to night time (Figure 13 (c)).

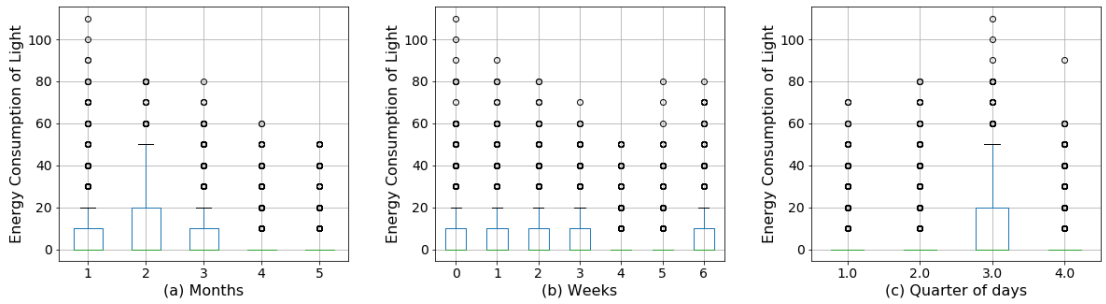


Figure 13. Monthly (a), Weekly (b), Quarterly (c) behavior of `Energy Consumption of Light`.

The dataset from Driven Data Four datasets are presented: train data - the historical data about the energy consumption for building sites, metadata - additional information such as surface area, the base temperature of the buildings, weather data - the temperature data from several stations near each site, holidays data - public holidays name and date at the sites. We only use energy consumption and weather data (temperature) for our study. Both energy consumption and temperature data include missing values. These values are imputed using the next valid observation in the series. The data is presented for 267 building sites. From 267 sites, 89 sites have daily, 89 sites have hourly and the last 89 sites have quarterly sampling rates. As we are interested in hourly and 15 minute time series, we work with two sites which have hourly and 15 minute frequencies. Figure 14 and Figure 15 described hourly and 15 minute Energy Consumption for the first month which have rise and falls periodically.

Box plot (Figure 16 (a)) shows that the average hourly Energy Consumption is less in the first and last months. Weekly energy consumption (Figure 16 (b)) is getting decrease to the end of the week. Figure 16 (c) reveals that more energy is consumed in the first half of the day.

In contrast to hourly Energy Consumption, box plot (Figure 17 (a)) shows that the average 15 minute Energy Consumption is more in the first and last months. Weekly energy consumption (Figure 17 (b)) is getting decrease to the end of the week. Figure 17 (c) reveals that less energy is consumed in the third quarter of the day.

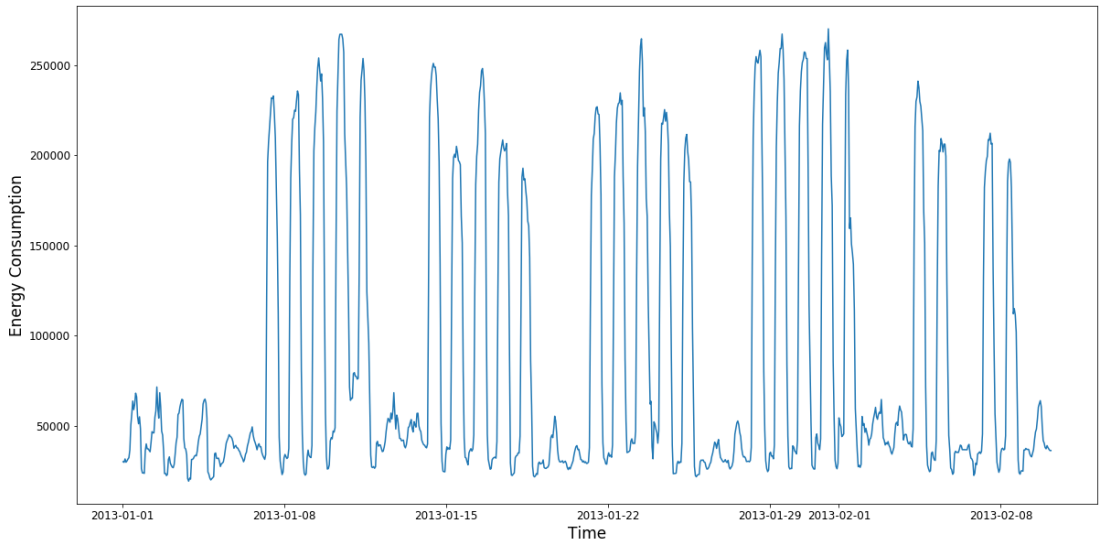


Figure 14. `Hourly Energy Consumption` of the building site for the first month.

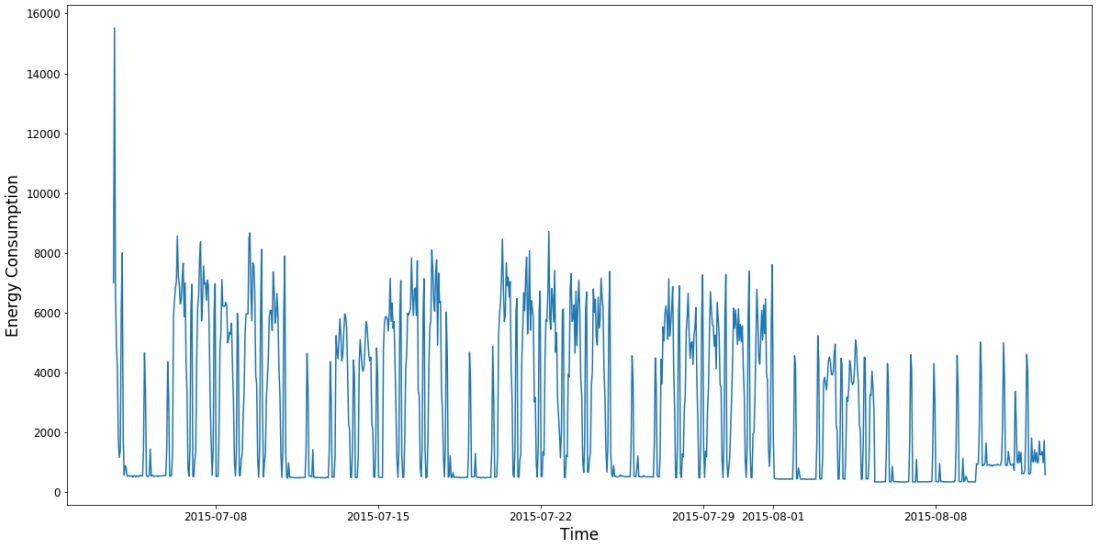


Figure 15. `15 minute Energy Consumption` of the building site for the first month.

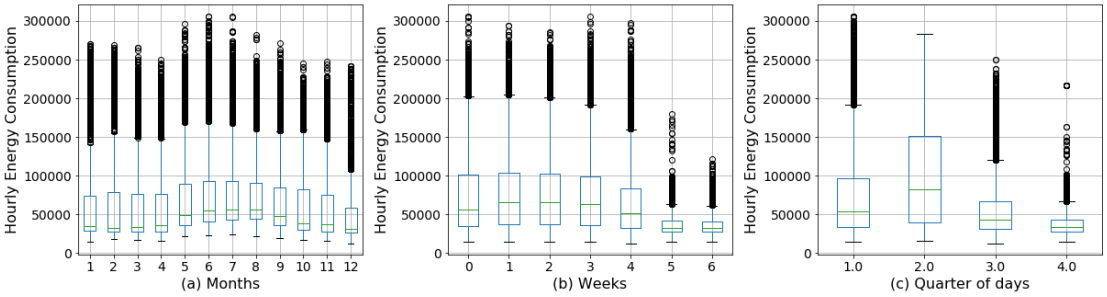


Figure 16. Monthly (a), Weekly (b), Quarterly (c) behavior of `hourly Energy Consumption`.

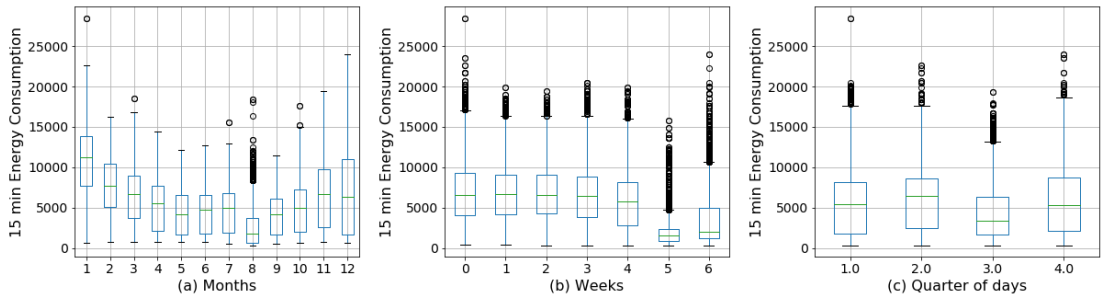


Figure 17. Monthly (a), Weekly (b), Quarterly (c) behavior of `15 minute Energy Consumption`.

The dataset from Open Power System Datasets This dataset is at an hourly resolution for about 2 years from 01-12-2014 to 31-12-2016. The dataset includes the time series of Load (Electricity) Consumption and weather information. The weather data contains the weather measurements (wind speed, temperature, direct horizontal radiation, diffuse horizontal radiation) from multiple weather stations. Load Consumption is used as a forecast measurement and all weather information is considered as features in multivariate forecast problem. Load Consumption for the first six months is described in Figure 18.

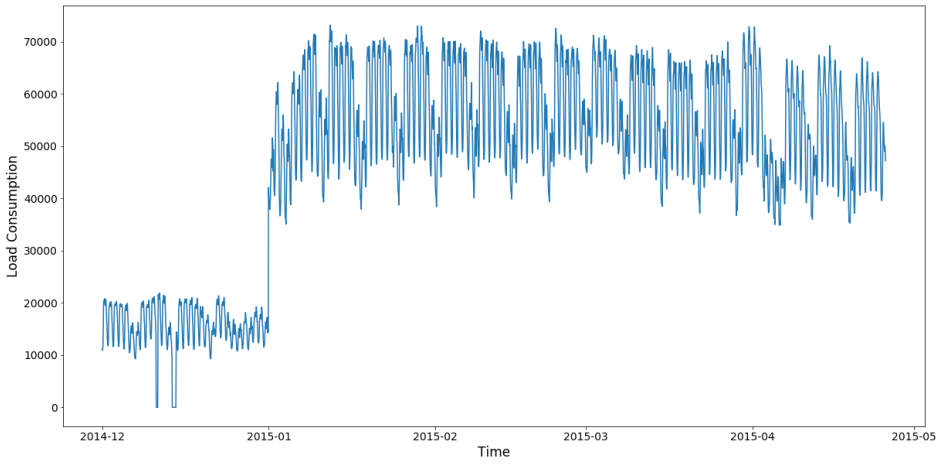


Figure 18. `Load Consumption` for the first six months.

Figure 19 (a) shows that there is especially less Load Consumption in December. Weekly average Load Consumption (Figure 19 (b)) is getting decrease to the end of the week. Figure 19 (c) reveals that less electricity is consumed in night time of the day.

4.2 Dataset preparation

All selected time series need to be transformed to the input and output series before fitting the model. The length of output series is defined as future 36 hours. As we work with time series with 15 minute and hourly resolution, the size of the output equals to 144 or 36 time steps. As the learning techniques are different for the baseline methods and LSTM models, two different data transformations are applied for the datasets.

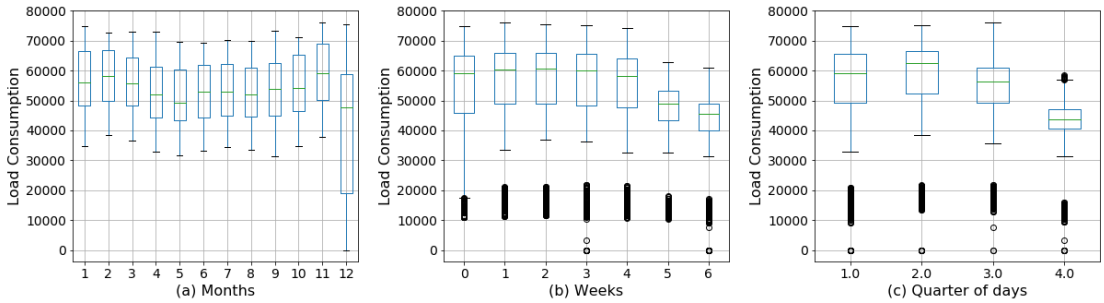


Figure 19. Monthly(a), Weekly(b), Quarterly(c) behavior of `Load Consumption`.

4.2.1 Dataset preparation for ARIMA

The transformation of the time series has been done in two steps. Firstly, the time series was split to the training and testing parts using the *train_test_split* method [26] with *test_size* fraction 0.2, setting *shuffle* parameter to *False*. As a result, the first 80% of the time series as training and the last 20% is used as testing data. In the second step, the training split is updated to implement 36 hours forecasting for all testing data. The first training set is used to forecast the first 36 hours in the testing data. To implement further 36 hours forecasting for all testing data, one past value is added to the training data iterating through the testing dataset and the next 36 hours are forecasted in each iteration. The generation process of the training and output series is described in Figure 20. The gray and orange areas correspond to the training and forecast data respectively. A training window (in grey) expands over the entire history of a time series and is repeatedly tested against forecasting window (in orange).

4.2.2 Dataset preparation for LSTM model

The transformation of the datasets for LSTM model has been done in three steps. In the first step, the time series were scaled between the range 0 and 1 using *MinMaxScaler* class [27]. *MinMaxScaler* is the first setting for scaling. For comparison purposes, another method like $\frac{x - mean}{std}$ could be investigated in the future work. In the data preprocessing, scaling is the important step to achieve the fast learning and convergence of the network [2]. In the second step, the dataset was split to the training and testing parts using the same methodology in the dataset preparation for ARIMA. In the last step, training and testing splits were divided to the input and output components using the sliding window technique described in Figure 21. In the sliding window technique, the previous n (window size) time steps are used as an input and the next k (forecast horizon) time steps are used as an output variable. Sliding window technique is implemented to enable supervised learning for LSTM network. After conversion the time series to the

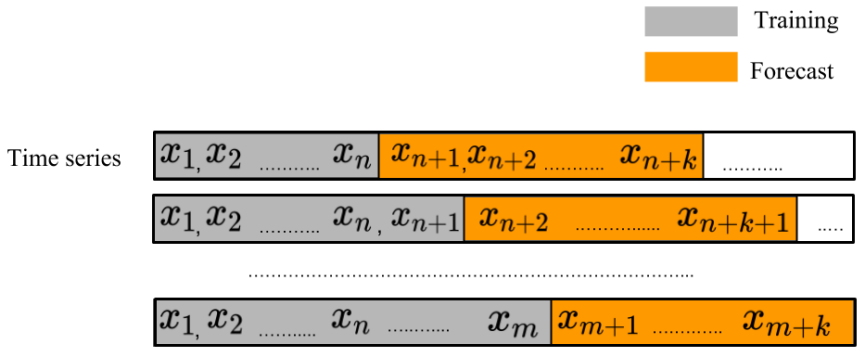


Figure 20. The time series transformation for ARIMA model.

supervised learning problem, we got the input shape in *[samples, window size, features]* format, which is the required input format for the LSTM network. While implementing the univariate forecasting, as we observe just one time series, the number of features equals to one. In the multivariate time series, as we also consider some other time series values, the size of features is more than one. All these data preprocessing steps were applied for each time series.

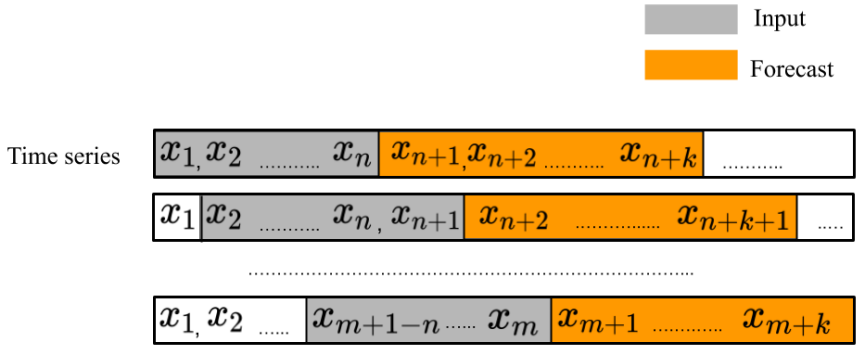


Figure 21. The time series transformation for LSTM model.

4.2.3 Dataset Preparation for Persistence

The time series is split to the training and testing parts are implemented as the same way in ARIMA (Section 4.2.1). The training and testing splits are divided to the input and output components using sliding window technique shown as implemented in Seection 4.2.2. To implement 36 hours forecasting, last 24 hours from the input data are used.

4.3 Experiments

4.3.1 ARIMA Model Formulation

In ARIMA model, the parameters *p*, *d* and *q* should be defined properly. In this work, the optimal value of *p*, *d* and *q* parameters was explored in the range of $[0, 1, 2, 4, 6]$, $[0, 1, 2]$ and $[0, 1, 2, 4, 6]$ respectively. The smaller range was chosen for d parameter, as some of time series are already stationary or there is a small trend pattern. The train split of each time series was used to search for the optimal combination of these parameters. This operation involves following steps:

- Split the train set to 50/50, 60/40, 70/30, 80/20 and 90/10 training/validation splits.
- Train ARIMA model for each training split considering all possible combination of *p*, *d* and *q* parameters and make 36 hours time step forecasting.
- Calculate the average RMSE error from validation splits for each combination.
- Choose the combination which has minimum RMSE score.

Using this method, the optimal *p*, *d* and *q* parameters were calculated for each time series as shown in Table 1:

Table 1. Chosen Optimal Hyper-parameters for each time series.

Time Series	**AR**	**I**	**MA**
Beijing PM2.5	p=0	d=1	q=6
Energy Consumption of Appliances	p=2	d=0	q=2
Energy Consumption of Light	p=4	d=0	q=2
Driven data Energy Consumption(hourly)	p=2	d=0	q=6
Driven data Energy Consumption(15 minute)	p=2	d=1	q=2
Open System Datasets Load Consumption	p=2	d=1	q=4

In conclusion, ARIMA(0,1,6) for Beijing PM2.5, ARIMA(2,0,2) for Energy Consumption Appliances, ARIMA(4,0,2) for Energy Consumption Light, ARIMA(2,0,6) for Driven hourly Energy Consumption, ARIMA(2,1,2) for Driven 15 minute Energy Consumption, and ARIMA(2,1,4) is trained for Load Consumption.

4.3.2 LSTM Models Formulation

Keras Deep Neural Network API [25] was used to build the LSTM models. The hyper-parameters were chosen based on literature review as shown in Table 2.

Table 2. Chosen Hyper-parameters for LSTM models.

Hyperparameters	**Values**
Activation Function	RELU
Optimizer	ADAM
Learning rate	0.005
Dropout rate	0.2
Batch size	64
Hidden units size	50
Epoch	150
Loss	MSE

To increase the stability of the networks, *Batch Normalization* layer was applied for hidden layer outputs after *RELU* activation. To overcome the overfitting issue, *Dropout* layer was added with *dropout fraction* 0.2 after batch normalization. As a dropout technique, the regular dropout function [28] is used which drops the linear transformation of the inputs. In the future work, for comparison purposes, the recurrent dropout function [28] could be applied which drops the linear transformation of the recurrent state. Both batch normalization and dropout techniques have regularization effects. In the Standard LSTM model, batch normalization and dropout layers were added after LSTM hidden layer sequentially (Figure 22). In the Stack LSTM model, batch normalization and dropout layers were applied after each LSTM hidden layer (Figure 24). In the S2S LSTM model, batch normalization and dropout layers were applied after decoder LSTM hidden layer (Figure 23).

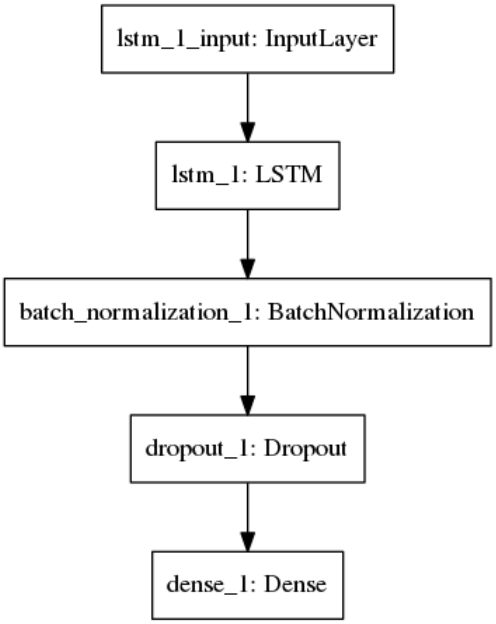


Figure 22. Standard LSTM model.

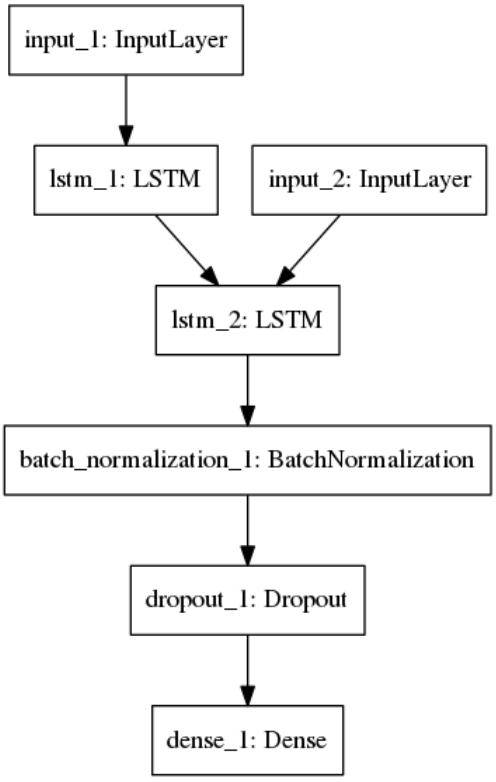


Figure 23. Encoder Decoder LSTM model.

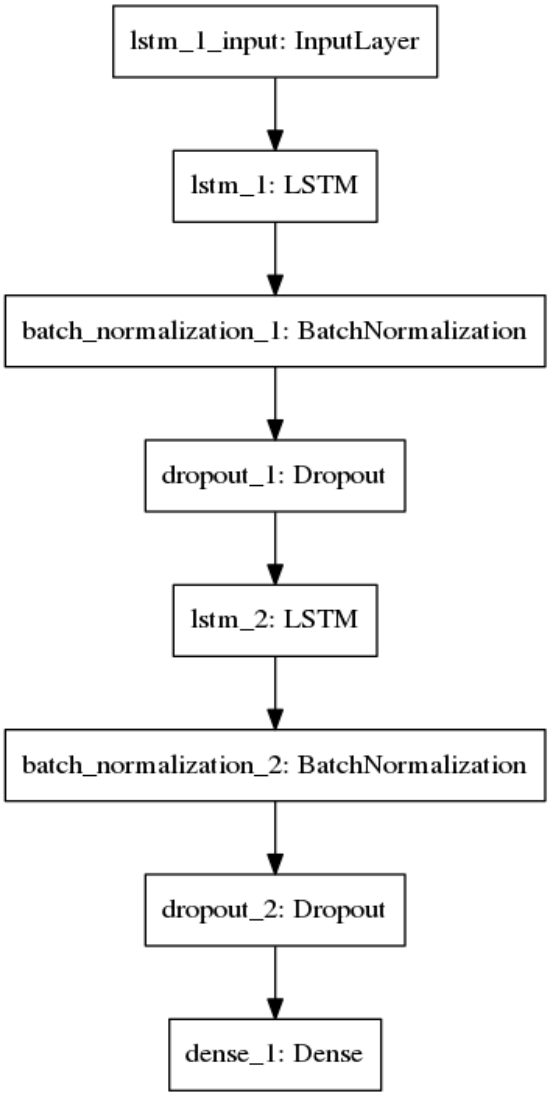


Figure 24. Stack LSTM model.

4.3.3 First Experimental Results

In total, six LSTM models were trained for each time series considering both univariate and multivariate problems. Each model was trained for 150 epochs and saved by 50 epochs. As a result, the performances of the models were evaluated after 50, 100 and 150 epochs. To compare the results, the average RMSE errors of 36 hours forecast points are used for each time series. As each time series had a different value range, the average RMSE errors of the models were not comparable at the same plot. That is why the RMSE scores were calculated based on scaled actual values and predictions. As there was still a challenge to compare the results, the scaled average RMSE errors of the models are divided to the scaled average RMSE errors of Persistence. In Figure 25, the average scaled RMSE errors of models' predictions are presented relative to Persistence. The average scaled RMSE errors of Persistence predictions are set to 1 for each time series. The best LSTM models were chosen based on the minimum average scaled RMSE errors by epochs. In Table 3, the best LSTM models by epochs are described for each time series.

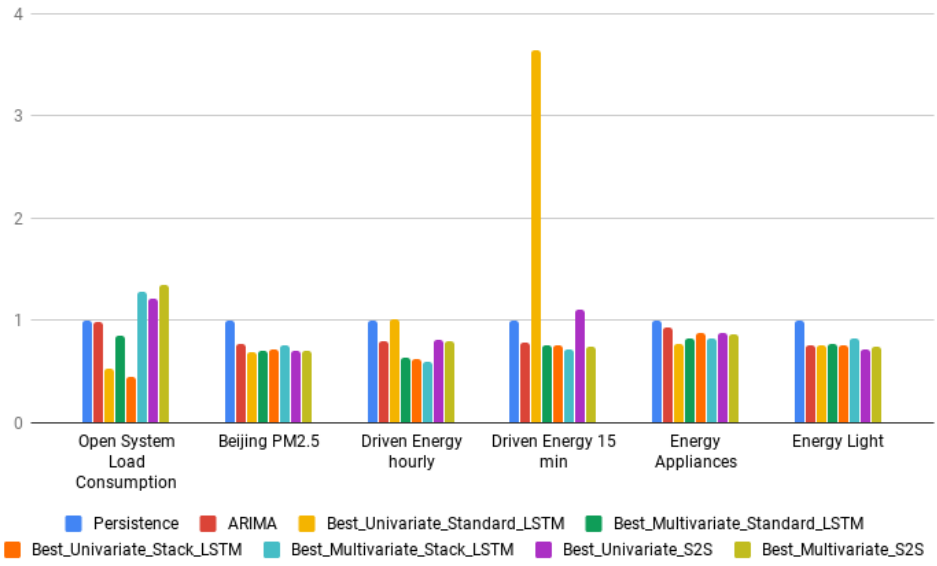


Figure 25. In the result of the first experiment, the average scaled RMSE error relative to Persistence.

Figure 25 shows that the ARIMA model outperforms the Persistence for each time series. It can be seen that the LSTM models are not always better than baseline methods. To improve the LSTM models performance, the early stopping technique was implemented which is explained in the Subsection 4.3.4.

Table 3. The best LSTM models by epochs for each time series.

	Best Univariate Standard LSTM	Best Multivariate Standard LSTM	Best Univariate Stack LSTM	Best Multivariate Stack LSTM	Best Univariate S2S LSTM	Best Multivariate S2S LSTM
Open System Data Load Consumption	epochs=100	epochs=150	epochs=150	epochs=100	epochs=100	epochs=150
Beijing PM2.5	epochs=100	epochs=50	epochs=50	epochs=50	epochs=150	epochs=50
Driven Energy Consumption (hourly)	epochs=150	epochs=50	epochs=50	epochs=100	epochs=150	epochs=150
Driven Energy Consumption (15 min)	epochs=50	epochs=150	epochs=150	epochs=100	epochs=150	epochs=50
Energy Appliances	epochs=50	epochs=50	epochs=50	epochs=50	epochs=50	epochs=100
Energy Light	epochs=50	epochs=100	epochs=100	epochs=50	epochs=50	epochs=50

4.3.4 Experimental results with Early Stopping

In this experiment, the early stopping technique was implemented to increase the performance of the LSTM models. As early stopping requires validation data, the last 10% of the training data was used as validation data. One more Stack LSTM model was trained which uses the *Batch Normalization* and *Dropout* layers only after the second LSTM hidden layer. In total, four models were trained for each time series considering both univariate and multivariate problem cases for 300 epochs. The best model was saved based on minimum validation loss during the training. The results are shown in Figure 26. We could achieve the improvement in the RMSE error for the models which had worse results than Persistence in the first experiment. Generally, the performance of Stack LSTM with one *Batch Normalization* and *Dropout* layers is about equal or worse than Stack LSTM with two *Batch Normalization* and *Dropout* layers for time series. However, Stack LSTM with one *Batch Normalization* and *Dropout* layers has definitely better results for Driven Energy hourly (both univariate and multivariate problem cases) and Energy Light (just multivariate problem case). As the performances of the LSTM models are still worse than Persistence in some cases, the parameter tuning technique

was applied in the next experiment. The Stack LSTM with one*Batch Normalization* and *Dropout* layers were trained for Driven hourly Energy Consumption and Energy Consumption of Light time series as it performs better than than Stack LSTM with two *Batch Normalization* and *Dropout* layers.

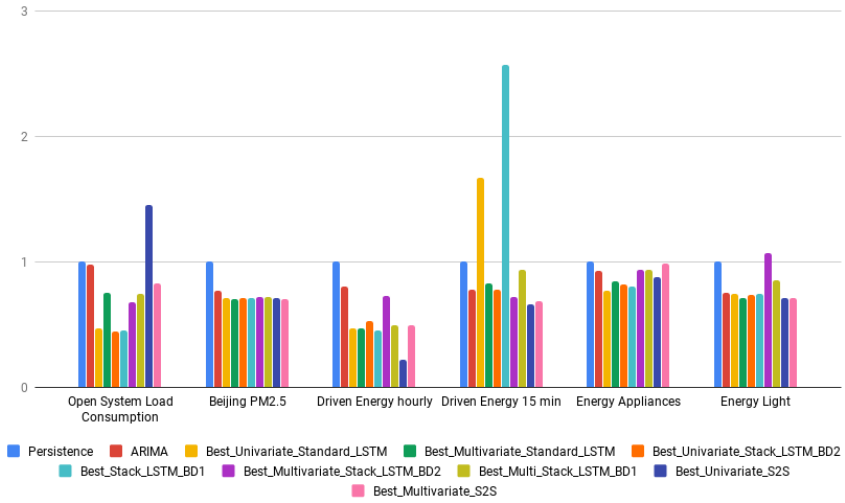


Figure 26. In the results of Early Stopping, the average scaled RMSE error relative to Persistence.

4.3.5 Experimental results with Parameter tuning

In this experiment, the parameter tuning method was applied together with early stopping. As tuning of all hyperparameters demands quite enough time, two important hyperparameters, the window size and the size of hidden units were tuned for each model. As the window size has a role in the definition of the input shape, it is more prioritized parameter in the parameter tuning. That is why, firstly the window size, then, the hidden units size was tuned for each model. In the initial experiments, the previous 36 hours were chosen as a window size to forecast the next 36 hours and the size of the hidden units was defined as 50. In this experiment, the models were trained with 24 and 48 hours window sizes additionally. The size of the hidden units was tuned with additional values 100 and 200.
Firstly, the window size was tuned. The average scaled RMSE results of models are presented for each time series in Table 4. The minimum average scaled RMSE errors by window size are bold for each time series. It can be seen that the optimal window size changes depending on the time series and model. In most of the cases, the Multivariate

LSTM models use the window size smaller than the Univariate LSTM models. It might happen because of the Multivariate LSTM models use additional features during the training. Choosing the best performing LSTM models from Table 4, the average scaled relative RMSE results are shown for each time series in Figure 27. The window size tuning improved the RMSE errors especially for the univariate Standard LSTM in Driven hourly Energy Consumption which had worse results than Persistence in the previous experiments.

Table 4. Average scaled RMSE error for each time series by LSTM models and window sizes.

Models	Window size	Open System Load Consumption	Beijing PM2.5	Driven Energy hourly	Driven Energy 15 min	Energy Appliances	Energy Light
Univariate Standard LSTM	24	0.051195	0.076558	0.003032	**0.000654**	0.072966	0.079712
	36	0.047266	0.077143	0.002861	0.001314	**0.071657**	0.080701
	48	**0.045585**	**0.076452**	**0.002626**	0.000678	0.073478	**0.079603**
Multivariate Standard LSTM	24	0.107526	**0.074569**	0.002962	0.000608	**0.076921**	0.082452
	36	**0.075867**	0.075376	**0.002840**	0.000654	0.078481	**0.077640**
	48	0.077754	0.075970	0.003076	**0.000586**	0.081422	0.080913
Univariate Stack LSTM	24	0.046244	0.078474	0.002908	**0.000572**	0.081791	0.082298
	36	0.045105	**0.076777**	0.002730	0.000616	**0.076951**	0.080453
	48	**0.044174**	0.080379	**0.002655**	0.000581	0.082381	**0.077973**
Multivariate Stack LSTM	24	**0.067629**	0.077796	**0.002781**	0.000560	**0.076281**	**0.080890**
	36	0.075706	0.077519	0.003017	0.000562	0.087148	0.116420
	48	0.133465	**0.076824**	0.002913	**0.000551**	0.092186	0.123611
Univariate S2S LSTM	24	**0.124194**	**0.076861**	0.001955	0.000519	**0.082162**	0.081701
	36	0.147838	0.077184	**0.001317**	0.000522	0.082213	**0.077727**
	48	0.146141	0.076885	0.002835	**0.000518**	0.082520	0.078645
Multivariate S2S LSTM	24	**0.082639**	0.076214	0.003192	0.000542	0.091685	0.077527
	36	0.083930	0.075880	**0.002983**	0.000537	0.092475	**0.077213**
	48	0.146244	**0.075054**	0.003442	**0.000536**	**0.090123**	0.076725

After tuning the window size, the size of the hidden units was tuned with additional values 100 and 200. The average scaled RMSE results of models are presented for each time series in Table 5. While tuning of window size, as the models were trained with 50 hidden units, the optimal hidden units size stays as 50 for most of the models. In Figure

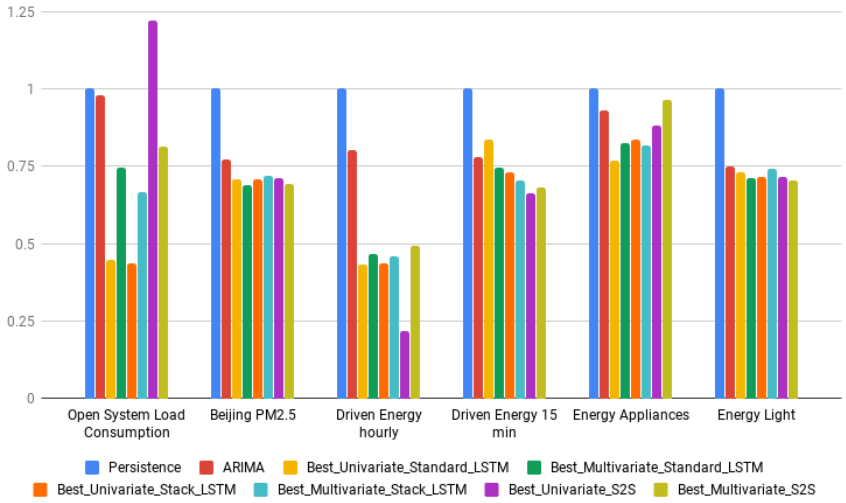


Figure 27. In the result of parameter tuning, the average scaled RMSE error relative to Persistence.

28, the average scaled relative RMSE result of each model are shown for each time series. The best LSTM models were chosen from Table 6 based on the minimum average scaled RMSE errors by the number of hidden units. The results show that the performance of the univariate and multivariate LSTM models depends on the LSTM model architecture and time series. Before it was expected that the multivariate LSTM models would outperform the univariate LSTM models in onerall. However the experiments showed that, the performance of the univariate and multivariate LSTM models is highly dependent on the LSTM architecture, the time series, and hyperparameters.
It can be seen that despite parameter tuning, the Univariate S2S LSTM model can not perform better results than baseline methods for Open System Load Consumption time series. The Univariate Standard LSTM and Multivariate S2S LSTM models have worse results than ARIMA model for the Driven hourly Energy Consumption and Energy Consumption of Appliances respectively. On the other hand, after the window size tuning, Multivariate Standard LSTM, Univariate Stack LSTM and Multivariate Stack LSTM results were already better than baseline methods. The tuning of the hidden units improved the results for these models just a little. It proves that the Multivariate Standard LSTM, Univariate Stack LSTM and Multivariate Stack LSTM are more stable respect to Univariate Standard LSTM, Univariate S2S LSTM and Multivariate S2S LSTM models. To analyze the Multivariate Standard LSTM, Univariate Stack LSTM, and Multivariate Stack LSTM deeply, the other evaluation metrics were calculated for each time series. For each model, the average of the RMSE, MAE, SMAPE, BIAS, and correlation of the

Table 5. Average scaled RMSE error for each time series by LSTM models and sizes of hidden units.

Models	Units size	Open System Load Consumption	Beijing PM2.5	Driven Energy hourly	Driven Energy 15 min	Energy Appliances	Energy Light
Univariate Standard LSTM	50	0.045585	0.076452	0.002626	**0.000654**	**0.071657**	**0.079603**
	100	0.045131	0.076260	0.002638	0.000685	0.072770	0.129262
	200	**0.044021**	**0.076259**	**0.002532**	0.000793	0.072461	0.082231
Multivariate Standard LSTM	50	0.075867	0.074569	**0.002840**	**0.000586**	0.076921	**0.077640**
	100	**0.065650**	**0.073729**	0.003015	0.000599	**0.075656**	0.080929
	200	0.070403	0.076313	0.003250	0.000610	0.084002	0.079810
Univariate Stack LSTM	50	0.044174	0.076777	**0.002655**	0.000572	**0.076951**	**0.077973**
	100	**0.042978**	0.076254	0.002669	0.000689	0.081972	0.079213
	200	0.130023	**0.076154**	0.002803	**0.000550**	0.082010	0.078467
Multivariate Stack LSTM	50	**0.067629**	0.076824	**0.002781**	**0.000551**	**0.076281**	**0.080890**
	100	0.099607	**0.074942**	0.003106	0.000557	0.079587	0.084644
	200	0.094711	0.077017	0.002942	0.000556	0.079265	0.097560
Univariate S2S LSTM	50	**0.124194**	**0.076861**	**0.001317**	0.000518	**0.082162**	**0.077727**
	100	0.136785	0.080200	0.003328	**0.000515**	0.082383	0.078292
	200	0.139985	0.080294	0.001962	0.000530	0.082338	0.079877
Multivariate S2S LSTM	50	0.082639	**0.075054**	**0.002983**	**0.000536**	0.090123	**0.077213**
	100	**0.081423**	0.075850	0.003412	0.000542	**0.088593**	0.077746
	200	0.133489	0.075390	0.003244	0.000564	0.094171	0.077644

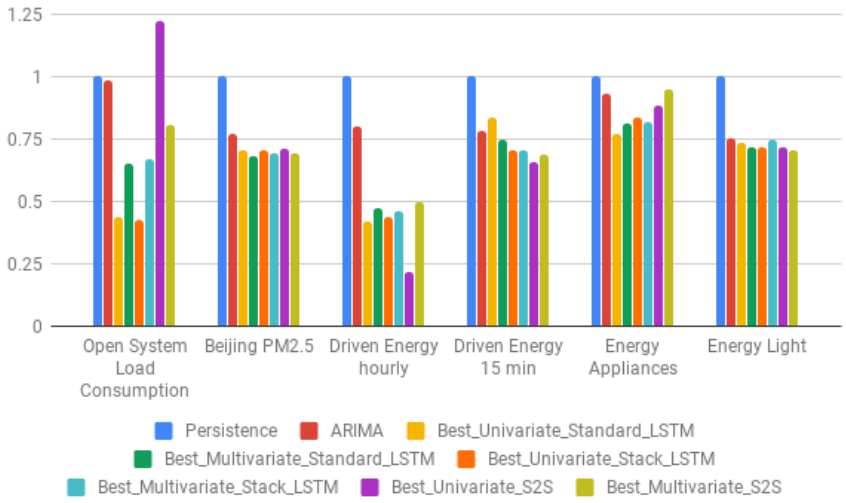


Figure 28. In the result of hidden units size tuning, the average scaled RMSE error relative to Persistence.

36 hours forecast points are shown in Table 6. The best-averaged results are bold for each time series. Generally, it can be seen from the high SMAPE errors that 36 hours forecasting does not work properly for Beijing PM2.5, Driven Energy 15 min, Energy Appliances, and Energy Light time series. According to the BIAS, the LSTM models mostly underestimate the predictions for all time series except for Beijing PM2.5. The high average score of the correlation for Open System Load Consumption and Driven hourly Energy Consumption proves that the predictions for the 36 hours forecast points are correlated to the actual values. The overall scores of the evaluation metrics shows the univariate Stack LSTM model is more robust than the other two LSTM models. In Subsection 4.4, the predictions results of the univariate Stack LSTM are discussed for each time series.

Table 6. Average scaled evaluation metrics results of Multivariate Standard LSTM, Univariate Stack LSTM and Multivariate Stack LSTM for each time series.

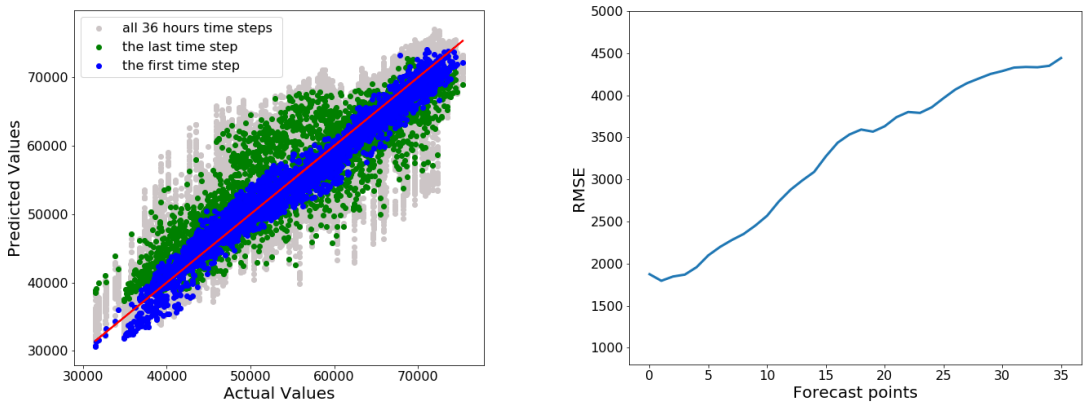
Models	Evaluation metrics	Open System Load Consumption	Beijing PM2.5	Driven Energy hourly	Driven Energy 15 min	Energy Appliances	Energy Light
Multivariate Standard LSTM	Avg. RMSE	0.065650	**0.073729**	0.002840	0.001314	0.075656	**0.077640**
	Avg. MAE	0.049925	**0.051067**	0.002134	0.001210	0.046617	0.047086
	Avg. SMAPE	3.496004	**30.086966**	10.749476	48.071790	**24.891671**	95.017037
	Avg. BIAS	-0.012298	0.009619	-0.000091	-0.000509	-0.010367	-0.005836
	Avg. CORR	0.861872	**0.587559**	0.781927	-0.061754	**0.451007**	0.032446
Univariate Stack LSTM	Avg. RMSE	**0.042978**	0.076154	**0.002654**	**0.000550**	0.0769506	0.077973
	Avg. MAE	**0.031728**	0.054697	**0.001973**	**0.000471**	0.045606	0.047965
	Avg. SMAPE	**2.213380**	31.137350	**9.704834**	27.692919	26.089760	**94.801964**
	Avg. BIAS	**-0.002075**	8.170910	-0.000467	**0.000138**	**-0.003392**	-0.007594
	Avg. CORR	**0.941322**	0.538969	**0.818458**	**0.220415**	0.349457	0.035428
Mutlivariate Stack LSTM	Avg. RMSE	0.067629	0.074942	0.002781	0.000586	**0.076281**	0.080890
	Avg. MAE	0.050789	0.052348	0.002059	0.000484	**0.045893**	**0.047052**
	Avg. SMAPE	3.541214	30.089051	10.259448	**26.106360**	25.723219	95.562714
	Avg. BIAS	0.009303	**0.005010**	**-0.000075**	-0.000202	-0.005385	**0.001445**
	Avg. CORR	0.856286	0.557959	0.770238	0.168665	0.371051	**0.103801**

4.4 Prediction results for time series

For better understanding of the quality of the predictions for each time series, the predictions are presented from the results of the Univariate Stack LSTM model together with Persistence and ARIMA models. The prediction results are described for each time series in their own sections. For each time series, five different figures are presented: i) the 36 hours time steps predictions of the Persistence, ARIMA, and Univariate Stack LSTM models for one particular sample, ii) the first time step predictions of the Univariate Stack LSTM model from 500 samples, iii) the last time step predictions of the Univariate Stack LSTM model from 500 samples, iv) the scatter plot of the actual values and predictions of the Univariate Stack LSTM model for the 36 hours forecast points, v) the RMSE errors from the Univariate Stack LSTM model for the 36 hours forecast points.

4.4.1 Predictions for Open System Load Consumption

In Figure 30 (a), the 36 hours time steps predictions of Persistence, ARIMA and Univariate Stack LSTM models are described for one particular sample. The time series on the left side of the vertical black line is the historical data of the last 48 hours. The predictions of the models and the true values of the time series are depicted with different colors which are labeled on the figure. Generally, it can be seen that the predictions can follow the trends in the time series. As expected, the Univariate Stack LSTM model predictions are more accurate than the baseline methods. In Figure 30 (b) and (c), the first time step and the last time step predictions of the Univariate Stack LSTM model are shown for 500 samples. Figure 30 (b) proves that the model can understand the patterns in the time series, and has accurate results for the first time step. The predictions for the last time step of the samples (Figure 30 (c)) are still meaningful, but they are less accurate especially for the weekends. This issue could be solved by introducing the weekends as additional input features to the model. The scatter plot (Figure 29 (a)) shows the actual values and predictions for all 36 hours forecast points (with light color), only first time step (with blue color), and last time step (with green color) from all samples. It can be seen that there is a linear relationship between the actual and predicted values and it proves that the model can do meaningful predictions for the further time steps. Figure 29 (b) describes that the RMSE scores increase linearly as the model can not predict the results accurate enough for the further time steps.



(a) The scatter plot of the actual values and predictions

(b) The RMSE errors for the 36 hours forecast points

Figure 29. The scatter plot of the actual values and predictions from the Univariate Stack LSTM model (a) and the RMSE errors from the Univariate Stack LSTM model (b) for the 36 hours forecast points of `Open System Load Consumption`.

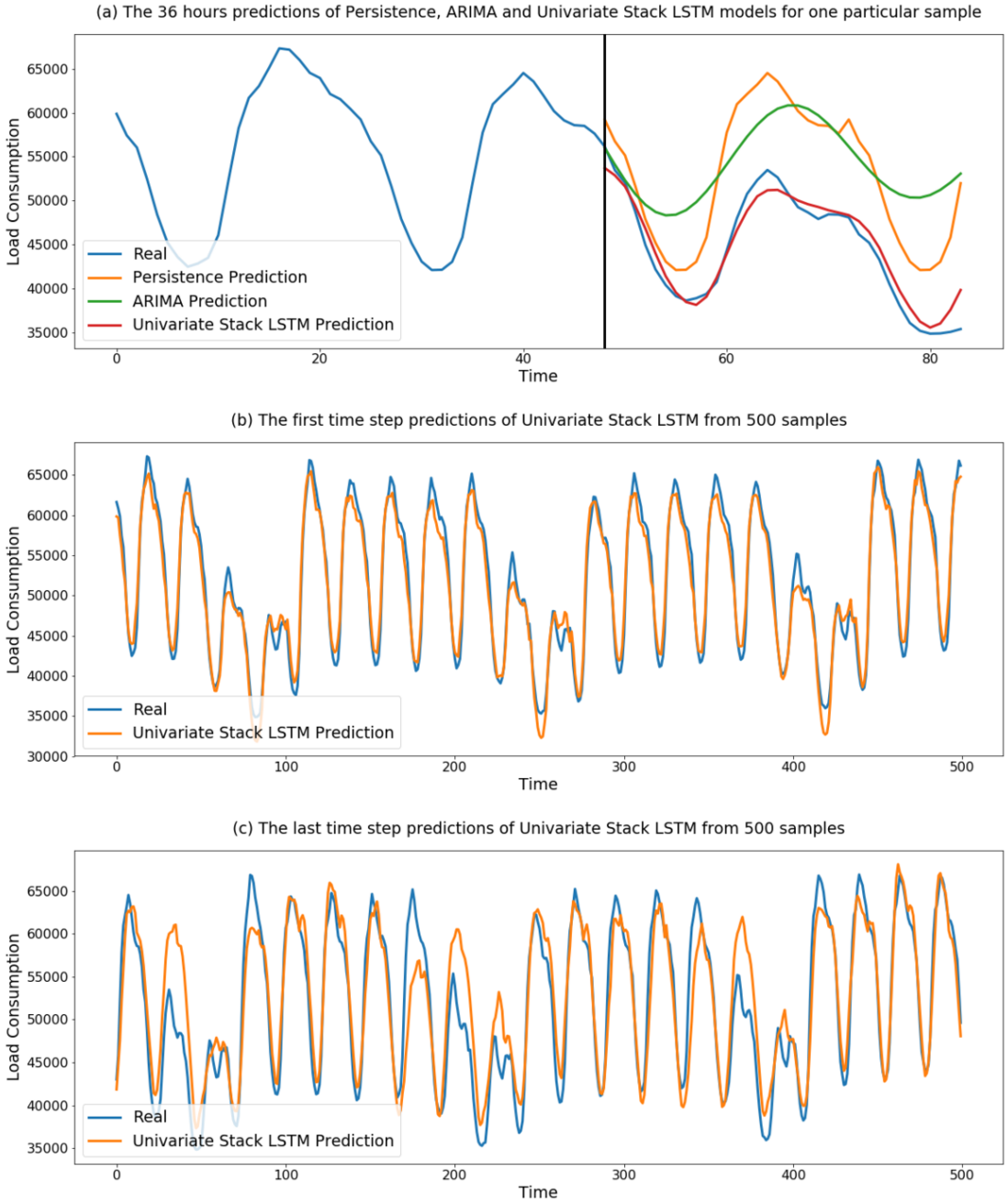
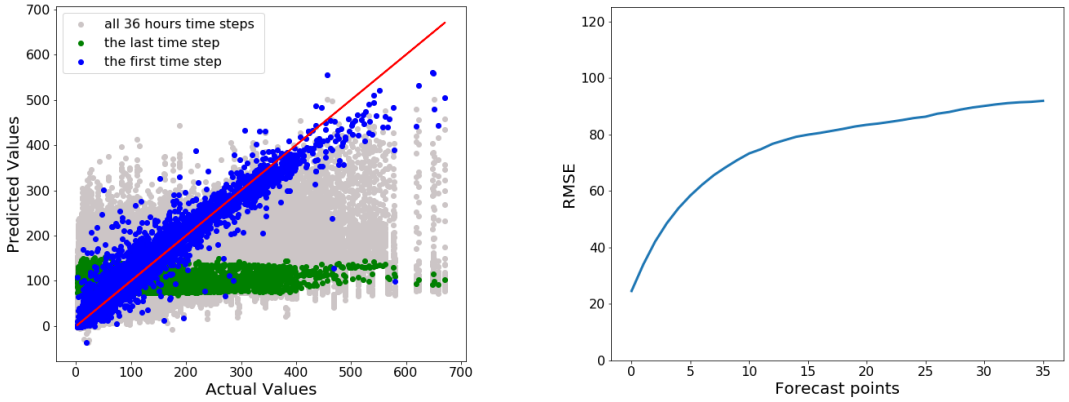


Figure 30. The 36 hours time steps predictions of the Persistence, ARIMA, and Univariate Stack LSTM models for one particular sample (a), the first time step predictions (b) and the last time step predictions (c) of the Univariate Stack LSTM model from 500 samples of `Open System Load Consumption`.

4.4.2 Predictions for Beijing PM2.5

In Figure 32 (a), the 36 hours time steps predictions of the Persistence, ARIMA, and Univariate Stack LSTM models are presented for one particular sample. The figure shows that the Univariate Stack LSTM model predictions can not follow the exact pattern for the whole forecast horizon, but have the near results to the true value at the first and last time steps. As the Persistence forecast use the values from the last 24 hours, the predictions are not appropriate to the next 36 hours. The ARIMA model has constant predictions for the whole forecast horizon. The first time step and the last time predictions of Univariate Stack LSTM from 500 samples are described in Figure 32 (b) and (c) respectively. It can be seen that the model has accurate predictions for the first time step, in turn, for the last time steps, there is no correlation between actual and predicted values of the samples. As the time series has no periodic pattern, it makes difficult for the model to do accurate predictions for the whole forecast horizon. Figure 31 (a) shows the actual values and predictions for all 36 hours forecast points (with light color), only first time step (with blue color), and last time step (with green color) from all samples. It can be seen that the actual values and predictions are in linear behavior for the first time steps from all samples but when the actual values are too high, the model can not predict those points accurately. As we discussed in Figure 32 (c), the scatter plot also shows that there is no correlation between actual and predicted values for the last time steps of the samples. In conclusion, the predictions for the last time steps spread out and in quite less linear behavior with actual values. As seen also from Figure 31 (b), there is a fast increase in the RMSE error for the further time steps.



(a) The scatter plot of the actual values and predictions

(b) The RMSE errors for the 36 hours forecast points

Figure 31. The scatter plot of the actual values and predictions (a) and the RMSE errors from Univariate Stack LSTM (b) for the 36 hours forecast points of `Beijing PM2.5`.

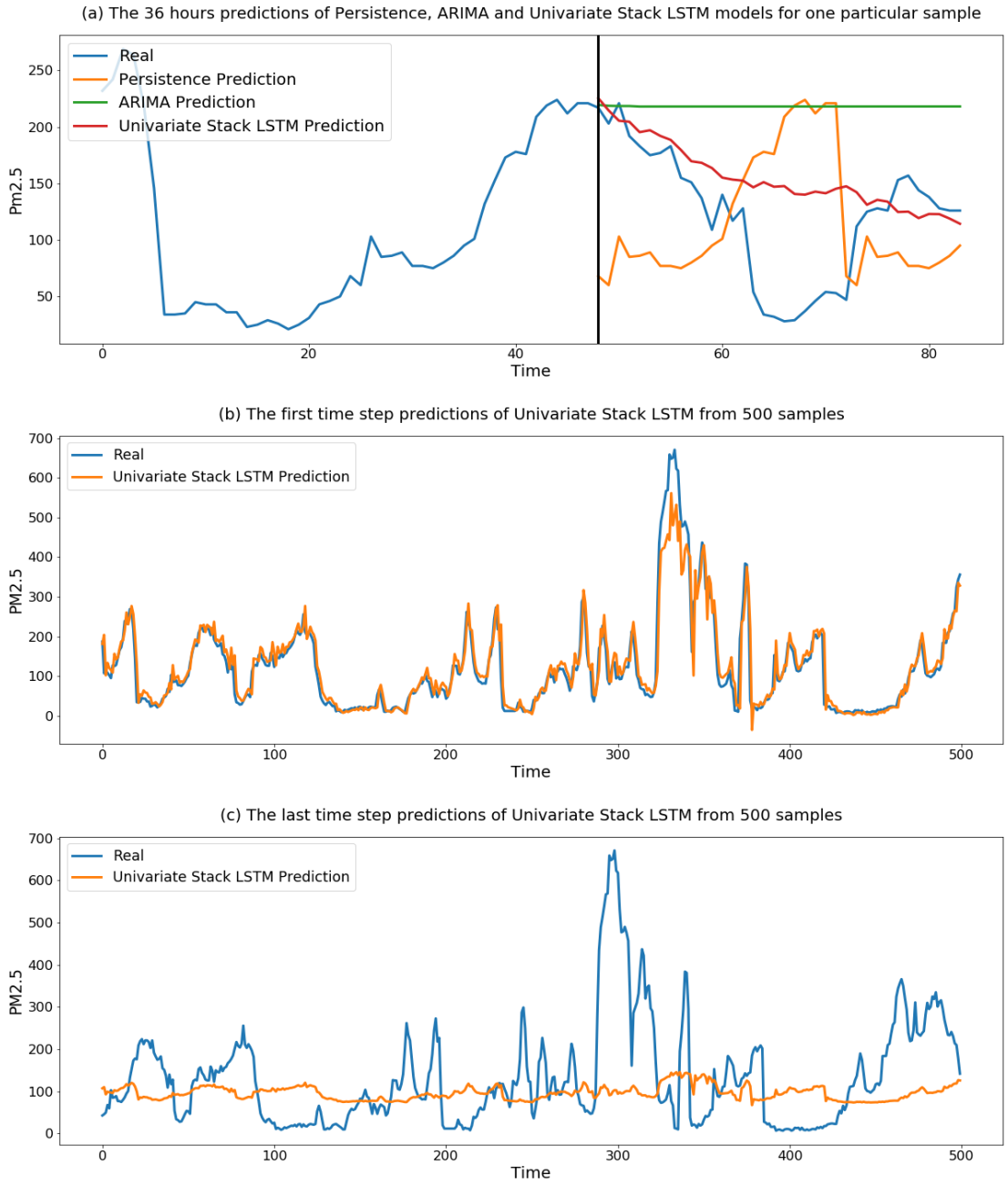
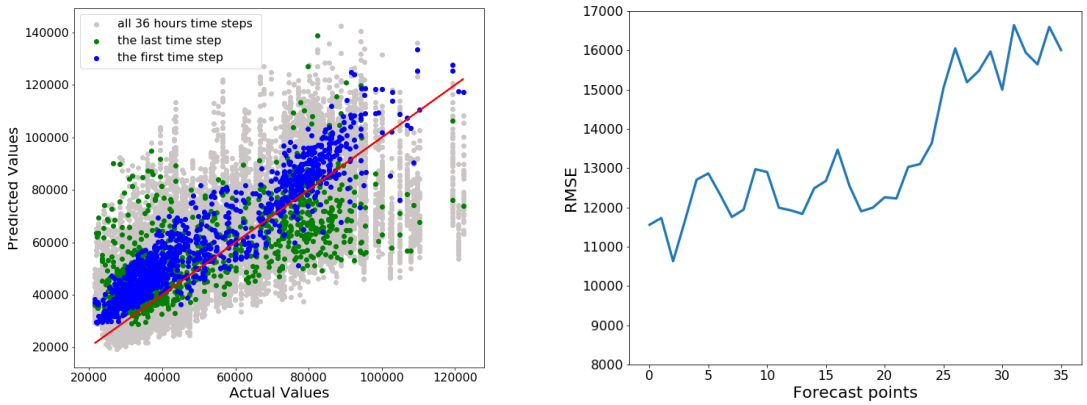


Figure 32. The 36 hours time steps predictions of the Persistence, ARIMA, and Univariate Stack LSTM models for one particular sample (a), the first time step predictions (b) and the last time step predictions (c) of the Univariate Stack LSTM model from 500 sample of `Beijing PM2.5`.

4.4.3 Predictions for Driven Hourly and 15 minute Energy Consumption

Predictions for Driven hourly Energy Consumption In Figure 34 (a), the 36 hours time steps predictions of Persistence, ARIMA, and Univariate Stack LSTM models are described for one particular sample. It can be seen that the Univariate Stack LSTM and ARIMA models can follow the trend for the true values. Figure 34 (b) and (c) describes the first and the last time step predictions of Univariate Stack LSTM from 500 samples. Generally, the figures show that both the first and the last time step predictions are correlated with actual values, but the model has a problem to do accurate predictions for the weekends. This issue could be solved by introducing the weekends as additional input features to the model. On the other hand, the model do less accurate predictions for Saturday rather than Sunday. The predictions for Sunday is more accurate as while doing the prediction for Sunday, the model looks back last 48 hours and that includes Saturday. However, just weekdays are used for the predictions for Saturday. Figure 33 (b) shows the actual values and predictions for all 36 hours forecast points (with light color), only first time step (with blue color), and last time step (with green color) from all samples. The scatter plot confirms that there is a correlation between the actual and predicted values but less accurate for the further time steps. Figure 33 (c) shows the RMSE scores from Univariate Stack LSTM for the whole forecast horizon.



(a) The scatter plot of the actual values and predictions

(b) The RMSE errors for the 36 hours forecast points

Figure 33. The scatter plot of the actual values and predictions of the Univariate Stack LSTM model (a) and the RMSE errors from the Univariate Stack LSTM model (b) for the 36 hours forecast points of `Driven hourly Energy Consumption`.

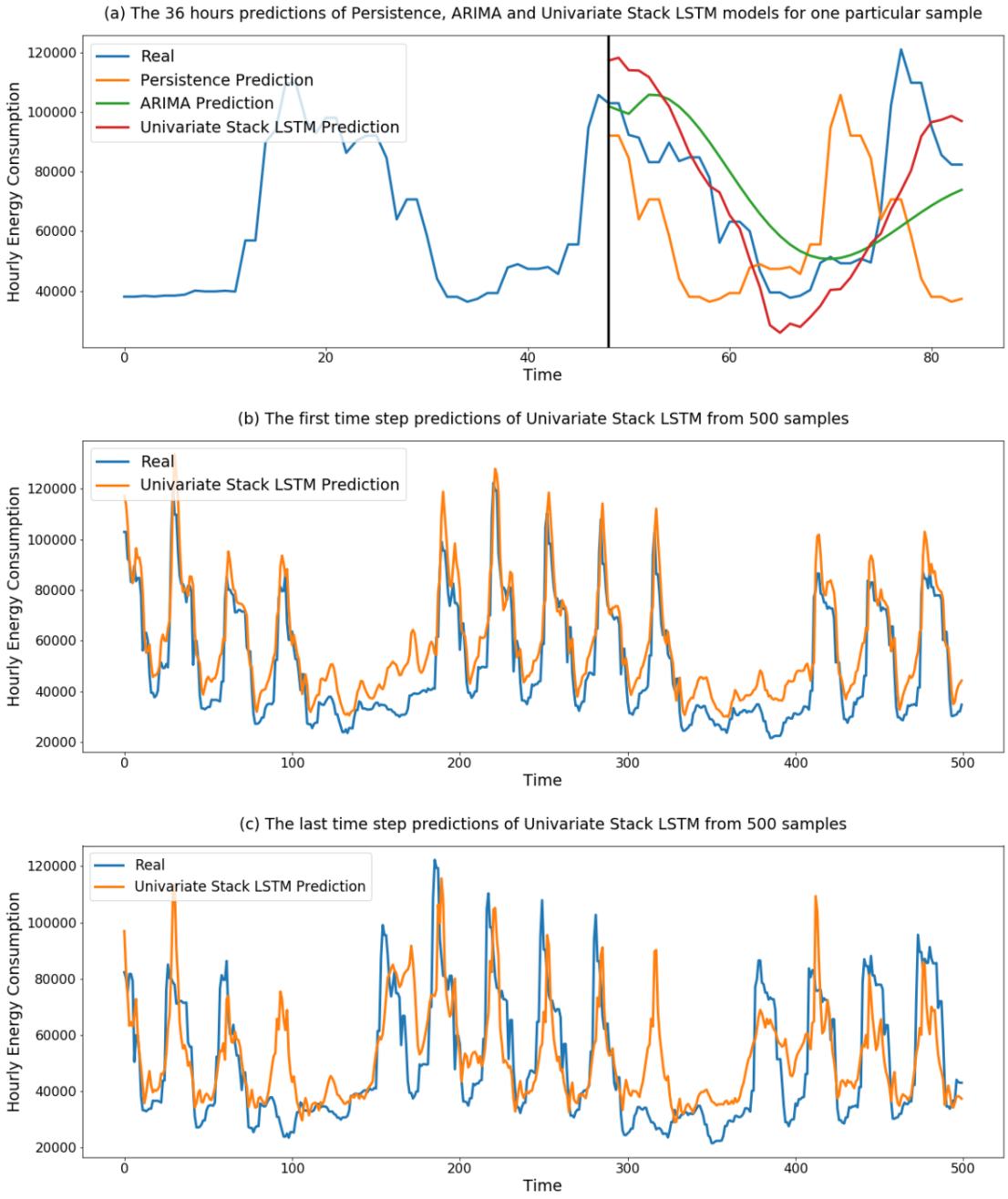
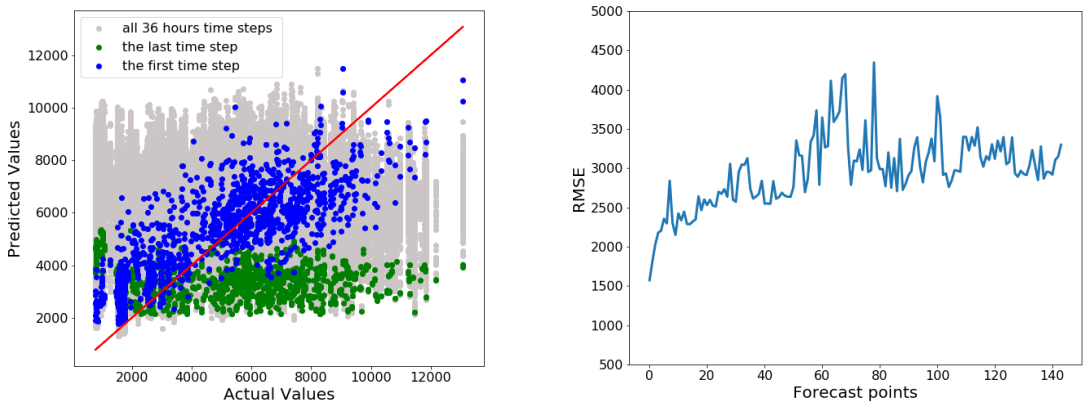


Figure 34. The 36 hours time steps predictions of the Persistence, ARIMA, and Univariate Stack LSTM models for one particular sample (a), the first time step predictions (b) and the last time step predictions (c) of the Univariate Stack LSTM model from 500 samples of `Driven hourly Energy Consumption`.

Predictions for Driven 15 minute Energy Consumption As this time series is in 15 minute resolution, the 36 hours forecasting covers the future 144 time steps. In Figure 36 (a), the 36 hours predictions of Persistence, ARIMA, and Univariate Stack LSTM models are presented for one particular sample. The ARIMA model has constant predictions after one particular time step. The outcome of Persistence is in the same behavior with true values but they do not overlap. The predictions of the Univariate Stack LSTM model tries to follow spikes and downs but they are not accurate enough. This behavior can be seen from Figure 36 (b) and (c) which shows the first and the last time step predictions of Univariate Stack LSTM from 500 samples. Figure (c) shows that the predictions for the last time step are far from the actual values. Figure 35 (b) shows the actual values and predictions for all 36 hours forecast points (with light color), only first time step (with blue color), and last time step (with green color) from all samples. It can be seen that the predictions for the first time steps form all samples are linear behavior with actual values but they are less confident. Generally, the scatter plot confirms the quality of the predictions are low especially for the further time steps. The RMSE errors from Univariate Stack LSTM for each 144 time steps are depicted in Figure 35 (c). As the time series has a pattern of spikes and falls, the fluctuations happen in the RMSE errors during the whole forecast period.



(a) The scatter plot of the actual values and predictions

(b) The RMSE errors for the 36 hours forecast points

Figure 35. The scatter plot of the actual values and predictions of the Univariate Stack LSTM model (a) and the RMSE errors from the Univariate Stack LSTM model (b) for the 36 hours forecast points of `Driven 15 minute Energy Consumption`.

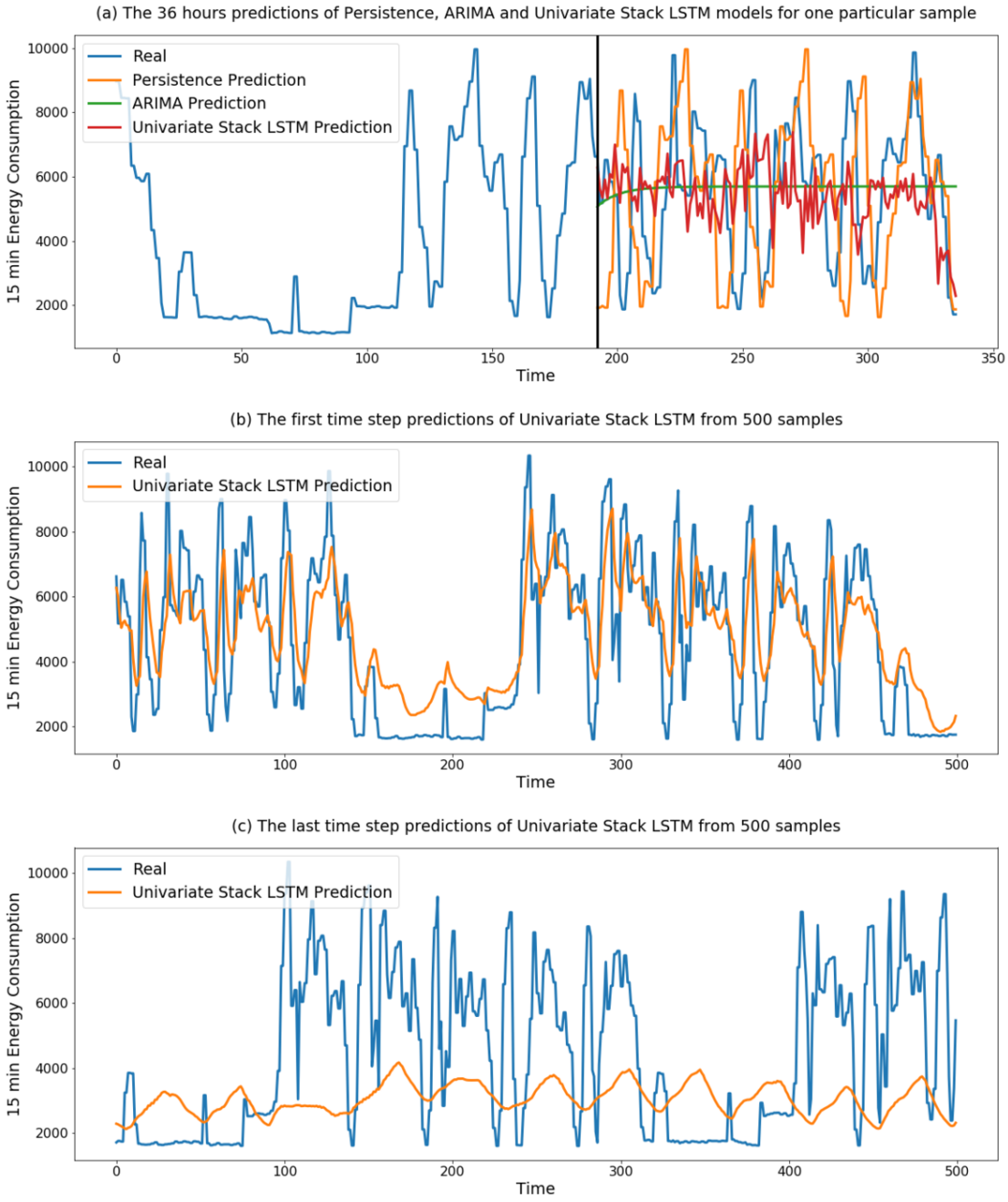
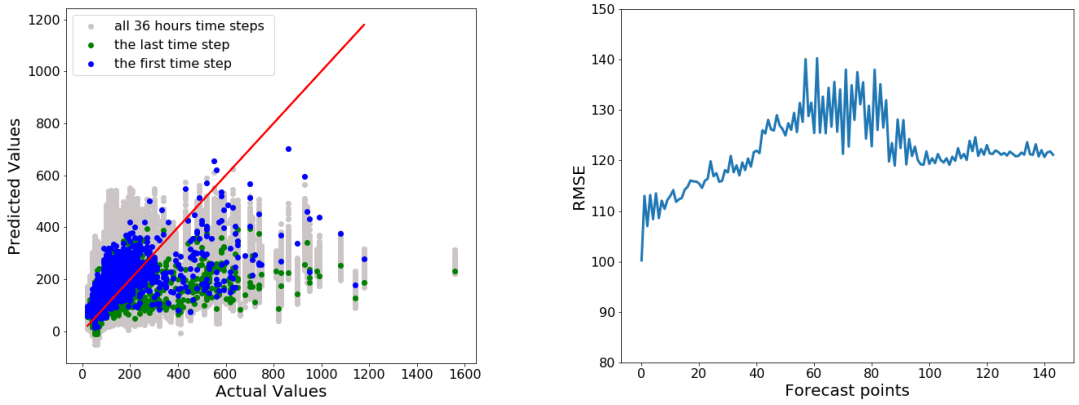


Figure 36. The 36 hours time steps predictions of the Persistence, ARIMA, and Univariate Stack LSTM models for one particular sample (a), the first time step predictions (b) and the last time step predictions (c) of the Univariate Stack LSTM model from 500 samples `Driven 15 minute Energy Consumption`.

4.4.4 Predictions for Energy Consumption of Appliances and Light

Predictions for Energy Consumption of Appliances In Figure 38 (a), the 36 hours predictions of Persistence, ARIMA, and Univariate Stack LSTM models are described for one particular sample. Univariate Stack LSTM model follows the patterns in the time series but can not do accurate predictions when the time series has high spikes. Figure 38 (b) and (c), which shows the first and the last time step predictions of Univariate Stack LSTM from 500 samples, confirms this fact. It can be seen that when the actual values are high, the model's predictions are far from true values. It might be related to the drastic changing characteristics in the training data. Figure 37 (b) shows the actual values and predictions for all 36 hours forecast points (with light color), only first time step (with blue color), and last time step (with green color) from all samples. The scatter plot (Figure 37 (a)) also describes that the model can not do accurate predictions for the high actual values in the time series. Figure 37 (b) presents the RMSE errors from Univariate Stack LSTM for 144 time steps. The fluctuations in the middle time steps might be related to the high changing patterns in the time series.



(a) The scatter plot of the actual values and predictions

(b) The RMSE errors for the 36 hours forecast points

Figure 37. The scatter plot of the actual values and predictions of the Univariate Stack LSTM model (a) and the RMSE errors from the Univariate Stack LSTM model (b) for the 36 hours forecast points `Energy Consumption of Appliances`.

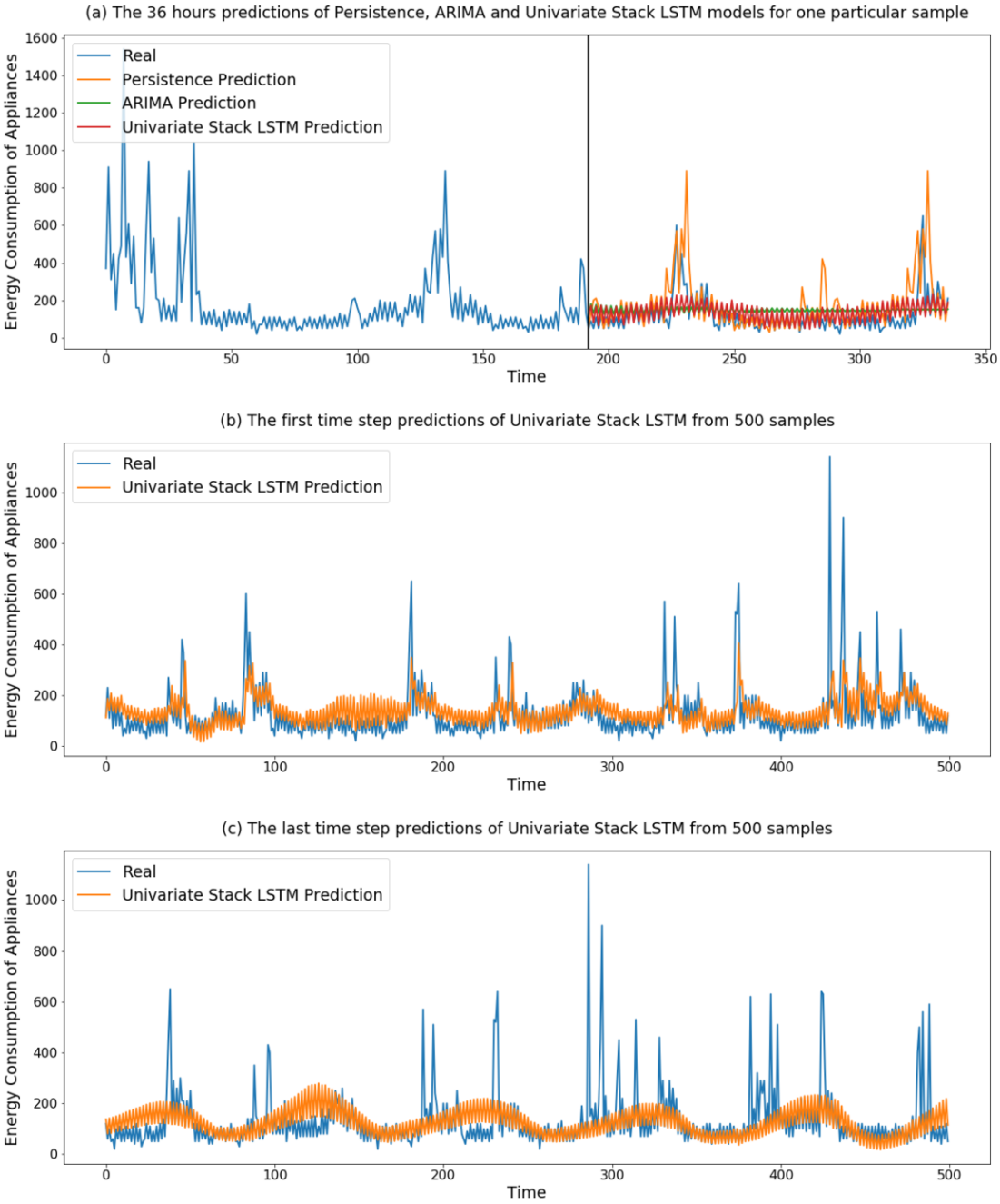
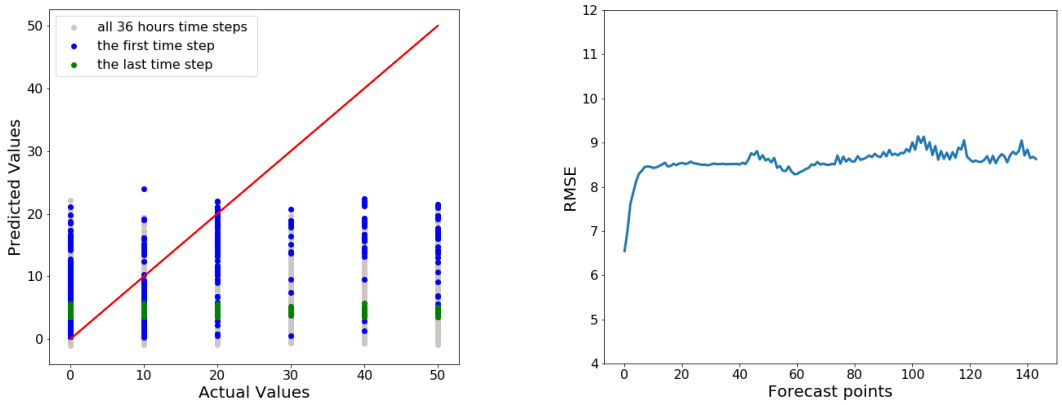


Figure 38. The 36 hours time steps predictions of the Persistence, ARIMA, and Univariate Stack LSTM models for one particular sample (a), the first time step predictions (b) and the last time step predictions (c) of the Univariate Stack LSTM model from 500 samples `Energy Consumption of Appliances`.

Predictions for Energy Consumption of Light This time series has a similar pattern to Energy Consumption of Appliances where time series has a drastic changing characteristic. Differently from the other time series, there are six unique values in this time series. In Figure 40 (a) the 36 hours predictions of Persistence, ARIMA, and Univariate Stack LSTM models are presented for one particular sample. The Persistence and ARIMA model predictions are off from the actual time series. The Univariate Stack LSTM model does not perform well due to the changing characteristic in the time series. Figure 40 (b) and (c) describe the first and the last time step predictions of Univariate Stack LSTM from 500 samples. It can be seen the model tries to follow the trend for the first time steps of the samples but can not predict accurate results. The predictions for the last time step are off from the actual values. Figure 39 (b) shows the actual values and predictions for all 36 hours forecast points (with light color), only first time step (with blue color), and last time step (with green color) from all samples. The poor quality of the predictions can be also seen from the scatter plot. In Figure 39 (b), the RMSE errors from Univariate Stack LSTM are described for each 144 time steps. The results show that after one particular time steps, the RMSE errors do not increase for the further time steps.



(a) The scatter plot of the actual values and predictions

(b) The RMSE errors for the 36 hours forecast points

Figure 39. The scatter plot of the actual values and predictions of the Univariate Stack LSTM model (a) and the RMSE errors from the Univariate Stack LSTM model (b) for the 36 hours forecast points `Energy Consumption of Light`.

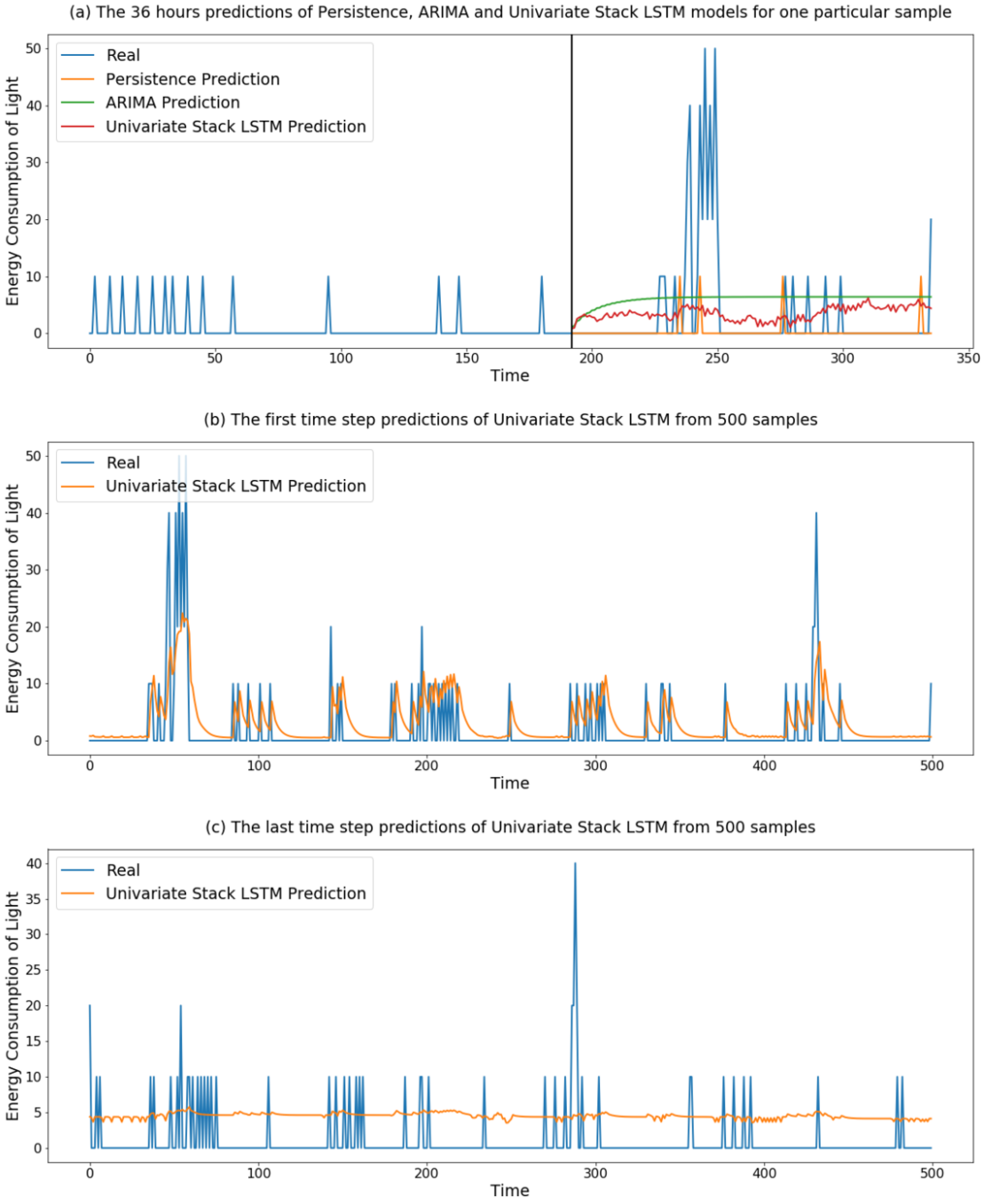


Figure 40. The 36 hours time steps predictions of the Persistence, ARIMA, and Univariate Stack LSTM models for one particular sample (a), the first time step predictions (b) and the last time step predictions (c) of the Univariate Stack LSTM model from 500 samples `Energy Consumption of Light`.

5 Conclusion

The purpose of the presented work was to investigate the effectiveness of LSTM based neural networks for energy time series forecasting. In this work, three different LSTM based univariate and multivariate models were built and optimized for 36 hours energy forecasting. For comparison purposes, the Persistence and ARIMA model were implemented as the baseline methods. Both univariate and multivariate forecasting problems were explored with LSTM models, in turn, the only univariate forecasting problem was considered for baseline methods. All models were trained for the three hourly and 15 minute time-step resolution data. Before the training, the data cleaning and scaling methods were applied for each time series. The time series was split training testing parts to validate the results of the models. The five different evaluation metrics were used to measure the performances of the models from different sides.

The optimal hyperparameters were chosen for the ARIMA model based on training/validation splits. Initially, the LSTM models were trained with hyperparameters according to the literature review. The early stopping and hyperparameter tuning were used to further improve the performances of the LSTM models. To implement the early stopping, the last 10% of the train data was used as validation data. The early stopping technique improved noticeably the performances of the LSTM models which had worse results than baseline methods initially. In the next steps, the window size and the size of the hidden units were tuned sequentially for each LSTM model. The results revealed the Univariate LSTM models perform better with bigger window sizes, in turn, the Multivariate LSTM models use the smaller window size for most of the time series due to usage of additional features. It was explored that the performance of the univariate and multivariate LSTM models depends on the LSTM architecture, hyperparameters, and time series and it is hard to claim that whether the Univariate or Multivariate LSTM model is more stable.

The analysis of the errors revealed that despite the parameter tuning, the performance of the Univariate Standard LSTM, Univariate S2S LSTM, and Multivariate S2S LSTM models are still worse than baseline methods in a few cases. In turn, even after tuning of the window size, the Multivariate Standard LSTM, Univariate Stack LSTM, and Multivariate Stack LSTM models performed better than the baseline methods for all time series. To compare the stability of the models, all evaluation metrics were analyzed for the Multivariate Standard LSTM, Univariate Stack LSTM, and Multivariate Stack LSTM models. The univariate Stack LSTM model shows the best results for three time series over all evaluation metrics and had near results to the best performance for the other time series. In conclusion, the Univariate Stack LSTM model was chosen as a robust model due to the stable results over all time series.

The predictions from the Univariate Stack LSTM model were studied for each time series. The results showed that the model could follow the trends on each hourly and 15 minute sampled time series but gave less accurate results for 15 minute sampled time series. It was assumed that it happened because of the changing characteristic of the time series.

During the analysis of the predictions, we noticed that the days of the weeks are important to do more accurate forecast. As future work, the feature engineering can be applied to consider the time features (months, days, hours) in the Multivariate LSTM models. Another objective could be to implement the parameter tuning methods such as Grid Search or Genetic Algorithm to find out the optimal combination of the hyperparameters for the LSTM models.

Acknowledgement

I am thankful for helpful discussions and guidance from colleagues at the Fraunhofer IEE Institute for Energy Economics and Energy System Technology (Kassel, Germany); in particular, I greatly appreciate the input provided by Katharina Brauns, Christoph Scholz, and Sebastian Wende-von Berg. This work has been supported by the EU-SysFlex project ("Pan-European system with an efficient coordinated use of flexibilities for the integration of a large share of RES") which has received funding from the European Union's Horizon 2020 research and innovation programme under EC-GA No. 773505.

References

[1] EU-SysFlex, Pan-European system with an efficient coordinated use of flexibilities for the integration of a large share of RES.
http://eu-sysflex.com/

[2] Shamsul Masum, Ying Liu and John Chiverton. *Multi-step Time Series Forecasting of Electric Load using Machine Learning Models.* Springer, May 2018.
https://link.springer.com/chapter/10.1007/978-3-319-91253-0_15

[3] Energy Forecasting. In Wikipedia, 18 March 2019
https://en.wikipedia.org/wiki/Energy_forecasting

[4] Daniel L. Marino, and Kasun Amarasinghe and Milos Manic. *Building Energy Load Forecasting using Deep Neural Networks.* IEEE, December 2016.
https://ieeexplore.ieee.org/abstract/document/7793413

[5] Kam, K. M. *Stationary and non-stationary time series prediction using state space model and pattern-based approach.* The University of Texas at Arlington, 2014

[6] Hagan, M. T. and Behr, S. M. *The time series approach to short term load forecastin.* In IEEE Trans. Power Syst., Vol. PWRS-2, No. 3 1987

[7] Espinoza, M., Joye, C., Belmans R., Moor and B. D. *Short-term load forecasting, profile identification, and customer segmentation: a methodology based on periodic time series.* In IEEE Trans. Power Syst., Vol. 20, No. 3 2005

[8] Muhamad Qamar Raza, C. Ekanayake and Mithulananthan Nadarajah. *On recent advances in PV output power forecast.* ResearchGate July 2016.
https://www.researchgate.net/publication/305321699

[9] Mocanu, E., Nguyen, P.H., Gibescu, M. and Kling, W.L. *Deep learning for estimating building energy consumption.* Sustain. Energy Grids Netw. 2016, 6, 91–99.

[10] Hyndman, R.J. and Shu, F *Density forecasting for long-term peak electricity demand.* IEEE Trans. Power Syst. 2010, 25, 1142–1153.

[11] Mohamed Abdel-Nasser, and Karar Mahmoud. *Accurate Photovoltaic power forecasting models using deep LSTM-RNN.* Springer, October 2017.
https://link.springer.com/article/10.1007/s00521-017-3225-z

[12] Salah Bouktif, Ali Fiaz , Ali Ouni and Mohamed Adel Serhani. *Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches.* Department

of Computer Science and Software Engineering, UAE University, 15551 Al Ain, UAE, Department of Software Engineering and IT, Ecole de Technologie Superieure, Montréal, QC H3C 1K3, Canada, January 2019

[13] Ho, S. L., Xie, M., Goh, T. N. *A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction.* Computers Industrial Engineering., vol. 42, pp. 371375 2002

[14] Igor Aizenberg a , Leonid Sheremetov b , Luis Villa-Vargas c , Jorge Martinez-Muñoz. *Multilayer Neural Network with Multi-Valued Neurons in time series forecasting of oil production.* Texas A&M University-Texarkana, Mexican Petroleum Institute, Computer Science Research Center of the IPN. June 2015

[15] Sepp Hochreiter and Jürgen Schmidhuber. *Long Short Term Memory.* Technische Universität München, Germany, IDSIA, Switzerland, 1997 http://www.bioinf.jku.at/publications/older/2604.pdf

[16] Andrej Karpathy. *The Unreasonable Effectiveness of Recurrent Neural Networks.* http://karpathy.github.io/2015/05/21/rnn-effectiveness/ May 21 2015

[17] Salah Bouktif, Ali Fiaz , Ali Ouni and Mohamed Adel Serhani. *Single and Multi-Sequence Deep Learning Models for Short and Medium Term Electric Load Forecasting.* Department of Computer Science and Software Engineering, UAE University, 15551 Al Ain, UAE, Department of Software Engineering and IT, Ecole de Technologie Superieure, Montréal, QC H3C 1K3, Canada, January 2019

[18] Understanding LSTM Networks, August 27 2015.
https://colah.github.io/posts/201508UnderstandingLSTMs/

[19] Medsker, L., Jain, L. *Recurrent neural networks, Design and Applications.* CRC Press LLC (2001)

[20] UCI Machine Learning Repository.
https://archive.ics.uci.edu/ml/index.php

[21] Driven Data. *Daily, hourly, and 15 minute Energy Consumption from 267 building sites.*
https://www.drivendata.org/

[22] Open Power System Data. *Hourly Load Consumption*
https://data.open-power-system-data.org/

[23] Beijing PM2.5 dataset.
https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data

[24] Appliances energy prediction dataset. *15 minute Energy Consumption of Appliances and Light fixtures.*
https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction

[25] F. Chollet, "Keras".
https://github.com/fchollet/keras 2015

[26] Sklearn, Train test split method.
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

[27] Sklearn, MinMaxScaler.
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[28] Keras, Dropout and Recurrent Dropout methods.
https://keras.io/layers/recurrent/

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Gunay Abdullayeva**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Application and Evaluation of LSTM Architectures for Energy Time-Series Forecasting,

   supervised by Alan Henry Tkaczyk, Meelis Kull, and Nicolas Kuhaupt.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Gunay Abdullayeva
16.05.2019