Augmented Out-of-sample Comparison Method for Time Series Forecasting Techniques

Igor Ilic[1], Berk Gorgulu[2], and Mucahit Cevik[1]

[1] Ryerson University {iilic,mcevik}@ryerson.ca
[2] University Of Toronto bgorgulu@mie.utoronto.ca

Abstract. Time series data consists of high dimensional sets of observations with strong spatio-temporal relations. Accordingly, conventional methods for comparing different regression methods, such as random train-test splits, do not sufficiently evaluate time series forecasting tasks. In this work, we introduce a robust technique for out-of-sample forecasting that takes the spatio-temporal nature of time series into account. We compare well-known auto-regressive integrated moving average (ARIMA) models with recurrent neural network (RNN) based models using hourly Turkish electricity data. We observe that RNN-based models outperform ARIMA models. Moreover, as the length of forecast interval increases, the performance gap widens between these two approaches.

Keywords: Time series prediction · Performance measures · Deep learning · ARIMA

1 Introduction

Humans have been trying to accurately predict the future for as long as history remembers. From historic models like Aristotle's Meteorologica [7] all the way to contemporary models like GluonTS [2], the ways to predict the future are plentiful.

Time series, $\{y_t\}$, are high dimensional sets of observations with strong spatio-temporal relations. They can be broken down into time-dependent deterministic component, Ω_t, and random components, ϵ_t, and are represented as follows:

$$y_t = \Omega_t + \epsilon_t$$

The goal of prediction models is to identify as much of the deterministic component, Ω_t, without overfitting and capturing random noise.

As the real-world continues to evolve and data continues to become available, finding the line between the two continues to be a challenge. With the amount of available modern tools, there is not enough analysis on the accuracy of a given prediction model. Due to strong spatio-temporal relations between observations, time series forecasting can not be treated as a regular regression problem. Learning the underlying deterministic component (Ω_t) requires the capabilities of modeling these spatio-temporal relations. Therefore, evaluation of

different time series forecasting algorithms requires special attention. The de facto evaluation methods in regular regression tasks, such as random train/test splits, are no longer suitable for time series forecasting. In this work, we introduce a new approach to compare models, known as *an augmented out-of-sample model* comparison method. This method is able to accurately compare different models on a dataset, with a higher degree of certainty.

Current comparison techniques for time series models include one shot comparison, random interval testing, and multiple dataset comparison [11,20]. These methods lack robustness, or the testing does not correctly reflect real-world situations. The augmented out-of-sample model comparison method alleviates these issues by providing a more flexible and robust technique. To demonstrate the effectiveness of the proposed approach, we compared numerous models using highly seasonal Turkish electricity consumption data. We find that recurrent neural network (RNN) architectures outperform classical algorithms in the associated forecasting task. Moreover, the gap widens as the forecast interval grows.

2 Background

2.1 Related Works

Forecasting methods in time series are designed to take spatio-temporal relations into account. Earlier studies on time series forecasting focus on linear prediction models which predict future values based on a linear function of past observations. Commonly used linear models include auto regressive (AR), moving average (MA) and auto-regressive integrated moving average (ARIMA) models [3]. Recent advances in artificial neural networks and deep learning allow researchers to utilize deep structures in time series forecasting. We refer the reader to recent review papers on deep learning methods in time series prediction [6,8] and focus our literature review on recent papers in electricity demand forecasting.

Electricity demand is a form of time series data with important applications. The most famous and widely known models in electricity demand forecasting are based on statistical models such as moving-average, exponential-smoothing and variants such as double exponential smoothing [18], double seasonal ARMA, and Holt–Winters methods [19]. Several recent studies focus on Long Short Term Memory Neural Networks (LSTM) for electricity demand forecasting [1,4,14,23]. In addition, some other studies focus on Gated Reccurent Units (GRU) architecture and compare it with the other RNNs [12]. Desirable characteristics of electricity datasets (especially the inherent high seasonality) make them suitable for benchmarking purposes, with recent deep learning based time series prediction studies reporting highly accurate forecasts over generic electricity datasets [15,16].

2.2 Time-series Comparison Techniques

In modern time series forecasting, there are plenty of ways to test how well an algorithm fits to a dataset. The most common methods include single forecast

testing, multiple dataset testing and random test interval sampling. These methods are a starting point to compare models against each other, but they lack rigor.

Single Interval Tests A commonly used method for comparing time series prediction models is a single interval test [11, 20]. The strengths to this method of comparison is the simplicity of implementation as well as the translation to real-world tests. Since the method to testing is the same as the one used in forecasting, no extra implementation is required. This makes it easier for practitioners to test and compare models. It is translatable to the real-world as well, since the way to test is exactly the same way to make a future prediction.

However, along with such strengths there exists some major pitfalls for single interval tests. Mainly, this approach allows for lucky one-shot tests to determine which algorithm is the most accurate. An algorithm that could be best suited for a dataset might happen to have a poor performance on a small section that is tested. The way to overcome this problem is to have many tests instead.

Multiple Datasets The first way to fix the lucky-one shot testing is by expanding the algorithm to predict the future (data points) for many datasets [13, 22]. By increasing the number of tests, the chance that an algorithm had a lucky one-shot performance diminishes without losing all of the other benefits from one-shot testing. This makes this approach better compared to single interval tests.

One pitfall to multiple dataset testing is that it is not easily able to identify which algorithm best fits each dataset. Although one algorithm may be better as a general algorithm, it does not identify which algorithm to use for a particular dataset. Moreover, multiple dataset testing is computationally more intensive. A lot of effort needs to be placed into finding numerous datasets, which is a challenging task on its own. Then, each dataset needs to be prepared, and each model needs to be trained on each dataset, which makes the implementation of this approach nontrivial.

Random Test Interval Sampling Another frequently used comparison method is random test interval sampling. This testing technique is commonly used in deep learning based architectures, (e.g. see [10]). Thanks to this feature, they can predict any time interval, as long as they have a certain number of preceding data points. This is different from traditional models, like ARIMA, which are semi-parametric models.

The benefit to this testing approach is that now a model can eliminate the need for multiple datasets to have multiple tests. That is, all tests can be done on the same dataset. Then, all the benefits of multiple dataset testing are also evident. The disadvantage of this method is that it is no longer predicting the immediate future, as is required in most practical applications. Typically, predictions are made for the immediate future, not what will happen at some arbitrary

point in the future. As well, we lose the ability to compare against algorithms that are data dependent, such as ARIMA. Specifically, ARIMA can not make a prediction at some arbitrary point in the future, without the preceding residuals. The residuals are only available for the trained dataset.

Augmented Training Before overcoming above issue, we need to look at a new way to train a time series model, as pioneered by Tashman et al. [17]. What was proposed here is a rolling method for training a dataset. The idea is to train the dataset, $train_1$, then test the trained model on a varying interval length of the next time period $test_1$. After this, the model's hyper parameters are tuned, the entire model is retrained on the initial test dataset $train_1 + test_1 = train_2$, and then the model is tuned on the next test interval $test_2$. This process is repeated until some stopping condition is met. Figure 1 presents a visual representation of this approach.

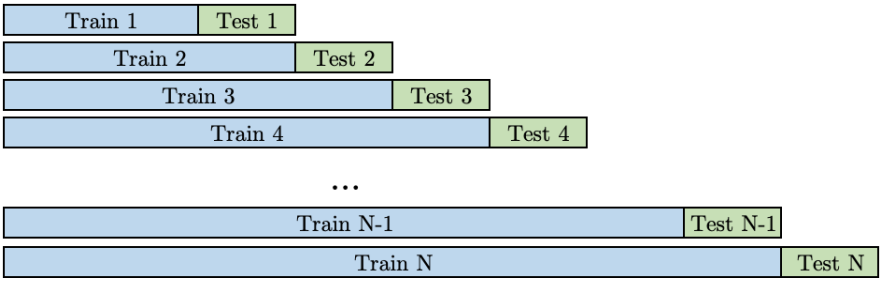

Fig. 1. Train-Test Rolling Visualization

This method has shown to be a fast ad hoc way to train a model with the best hyper parameters, as well as a way to determine the performance of a single model.

3 Augmented Out-of-sample Comparison Technique

By taking the methodologies introduced in the augmented training, we can similarly evaluate numerous models on test datasets. Specifically, the dataset is used to obtain the train-test sets, a forecast interval, and the number of tests. After each test, the model is updated to include the test data. Since the model is merely updated on a small portion of data, we do not have to retrain the entire model as found in rolling horizons, which leads to a significant speedup in testing. The proposed approach is summarized in Algorithm 1.

By using fixed forecast intervals, numerous models can all be compared using the same testing points. The test values can be then amalgamated using any preferred metric, e.g. mean absolute error, mean squared error, and root mean

Algorithm 1: Augmented Out-Of-Sample Testing

Input : Dataset sorted by ascending date as $\mathcal{D}$, algorithm as f, test interval length as ℓ, number of tests as n

Output: Array of predicted values and real values

$\mathcal{T} \leftarrow$ **CollectTrainingData**$(\mathcal{D})$;
$\mathcal{U} \leftarrow$ **CollectTestingData**$(\mathcal{D})$;
$model \leftarrow$ **TrainUsing**$(f, \mathcal{T})$;
$\{C_j\}_{j=1}^{n} \leftarrow$ **Split**$(\mathcal{U}, n)$;
$results \leftarrow \emptyset$;
for $i = 1 \dots n$ **do**
    $testingData \leftarrow$ **RetrieveFirst**(ℓ, C_i);
    $testResults \leftarrow$ **TestUsing**$(model, testingData)$;
    $results \leftarrow results \cup testResults$;
    $model \leftarrow$ **UpdateUsing**$(model, C_i)$;
end
return $results$;

percentage error. Comparison plots can be made to ensure that outlier data points are not impacting the results.

If the dataset has some seasonality, s, it is important to check that the test interval length, ℓ, is chosen such that $gcd(s, \ell) = 1$. This ensures that the test intervals capture a wide variety of tests, instead of only a particular segment of the seasonality.

4 Experiments

To demonstrate the power of the proposed technique, we evaluate numerous models on the Turkish electricity dataset[1]. Specifically, we consider two RNN algorithms: LSTM [9] and GRU [5]. In addition, we implement a special variant of the ARIMA model, called SARIMAX (Seasonal AutoRegressive Integrated Moving Average with Regressors), and a naive baseline model which uses the last week's data to predict the current week.

4.1 Exploratory Data Analysis

To get an idea of the dataset, we first explore what the dataset looks like. The dataset is broken down into five years worth of hourly data. A two week sample of the data has been demonstrated in Figure 2.

[1] https://seffaflik.epias.com.tr/transparency/tuketim/gerceklesen-tuketim/gercek-zamanli-tuketim.xhtml

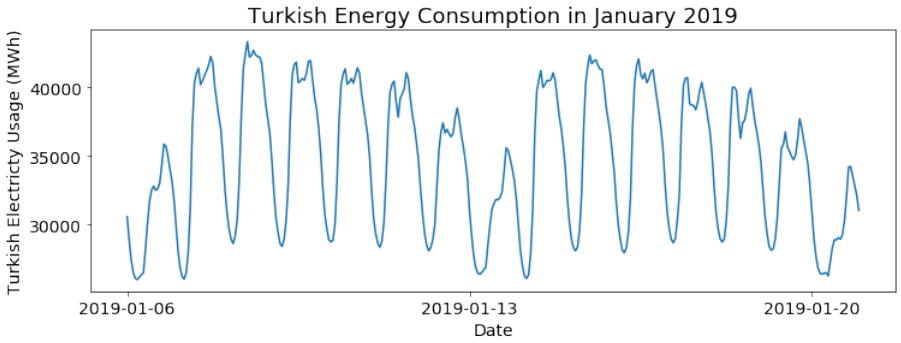

Fig. 2. January 2019 Turkish electricity consumption

Each day, on average, follows a typical trend where the electric usage starts to rise in the morning hours, peaks in the middle of the day, and then tapers off in the evening. There is weekly seasonality as well, with more electricity usage in the weekdays compared to the weekends (see Figure 3).

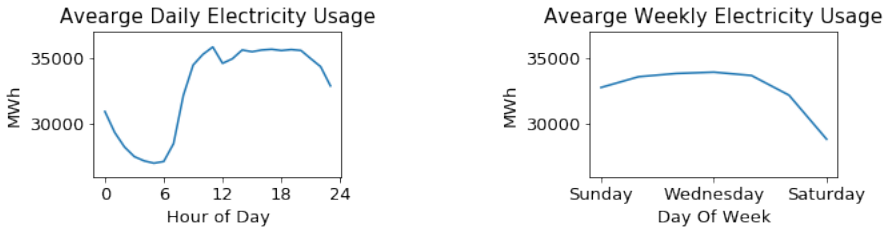

Fig. 3. Turkish electricity average consumption

4.2 Experimental Settings

In our analysis, each model was fit using an 80-20 train-test split, with 25 individual tests. Then, three forecast intervals have been chosen: 6 hours, 24 hours, and 48 hours. This allows for 150, 600, and 1200 test points, respectively. This approach is helpful in seeing which model to use for each forecast interval. Only multi-step forecasts were evaluated, but the same principles translate to single-step forecasting as well.

Parameters of the SARIMAX model were determined by optimization in order to ensure the best performance. The resulting SARIMAX model has an order of $(1, 1, 2)\times(1, 0, 1)24$ for the $(p, d, q)\times(P, D, Q)m$ parameters, respectively. Additional regressors were added on to capture more seasonality, as discussed in Section 4.3.

Both LSTM and GRU models contain two layers, with the first layer containing 32 hidden units and the second containing 16. The two-layer architecture was used in order to deal with the complicated nature of making a multi-step forecast. In addition, 15% of the training data was removed and put into a validation step. This was done in order to perform cross validation during training. Once the loss in the validation plateaued, training ceased and the trained models were saved.

Using a set of test values $\mathcal{T}$ of length N, we use Mean Absolute Percentage Error (MAPE) as our aggregation metric.

$$\text{MAPE} = 100 \times \frac{1}{N} \sum_{t \in \mathcal{T}} \frac{|\hat{y}_t - y_t|}{y_t}$$

where, for a given time $t \in \mathcal{T}$, the predicted value is represented as $\hat{y}_t$ and the observed value is y_t.

4.3 Additional Regressors

Additional regressors considered were cyclic day-of week and hour-of-day regressors. Since there is some inherent seasonality, instead of letting the model pick up this information, we pass it to the model explicitly. We use a sinusoidal transformation on the day of week (dow), as well as hour of the day (hod) to add in four additional regressors.

$$\text{day of week}_{sin} = \sin\left(dow * \frac{2\pi}{7}\right) \qquad \text{day of week}_{cos} = \cos\left(dow * \frac{2\pi}{7}\right)$$

$$\text{hour of day}_{sin} = \sin\left(hod * \frac{2\pi}{24}\right) \qquad \text{hour of day}_{cos} = \cos\left(hod * \frac{2\pi}{24}\right)$$

We observed that addition of these regressors improved the MAPE value by 10-30% depending on the model. Several other potential regressors have not been included in this study. There are religious and national holidays in Turkey which likely have an effect on the electricity consumption. As well, incorporating weather data into the model could yield benefits. For the purpose of initial demonstration of augmented out-of-sample comparison, these additional regressors were purposely omitted as they do not contribute to the understanding of the proposed comparison technique. However, in a follow-up research on the Turkish electricity dataset, it is highly recommended to incorporate these regressors.

4.4 Benefits of Augmented Out-Of-Sample Testing

Since the subtests were evaluated on many different points, our testing captured many cases. In Figure 4, each point on the x-axis is an individual subtest, which could be a lucky-one shot test. We see that the subtest lines overlap, and the performance of the subtest is heavily dependent on the subtest interval. If we had used only one test, the choice could have been unlucky. For example in the 48-hour test, subtest 17 ranks algorithms from best to worst as SARIMAX, LSTM,

GRU, Baseline. This is different from the actual best raking of GRU, LSTM, Baseline, SARIMAX. In addition, six sample forecasts and predictions have been provided in Figure 5, which can be used to visually assess the performances of these models.

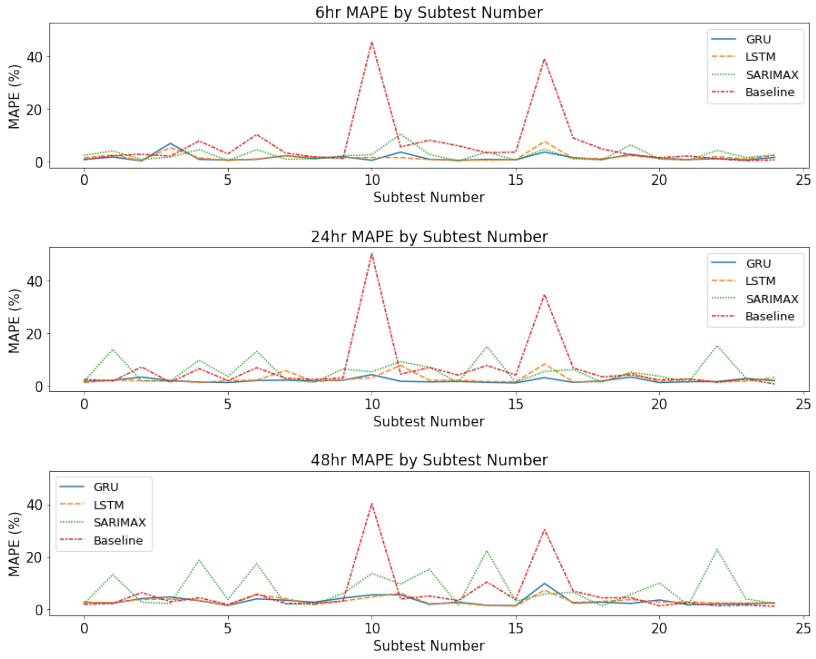

Fig. 4. Subtest MAPEs for Multiple-hour Forecast

4.5 Results

Overall, we found that RNNs perform better than the rest of the models. We see the results in Figure 6, with 95% confidence interval error bars. In addition, Table 1 provides a numerical comparison between different models. In particular, the LSTM and GRU models produce nearly similar results for a 6-hour forecast and a 48-hour forecast. The GRU is the best for the 24-hour forecast. We note that each trained model predicts less accurately as the time horizon increased. Interestingly, SARIMAX perform worse than the naive baseline model for the long forecast. This is because the SARIMAX model is not well-suited to predict 48 intervals into the future with its relatively low number of parameters.

The resulting neural network models perform fairly consistently across all the tests, with a low deviation from the average error. In fact, while our objective was not to develop the best possible prediction model, we note that predictive

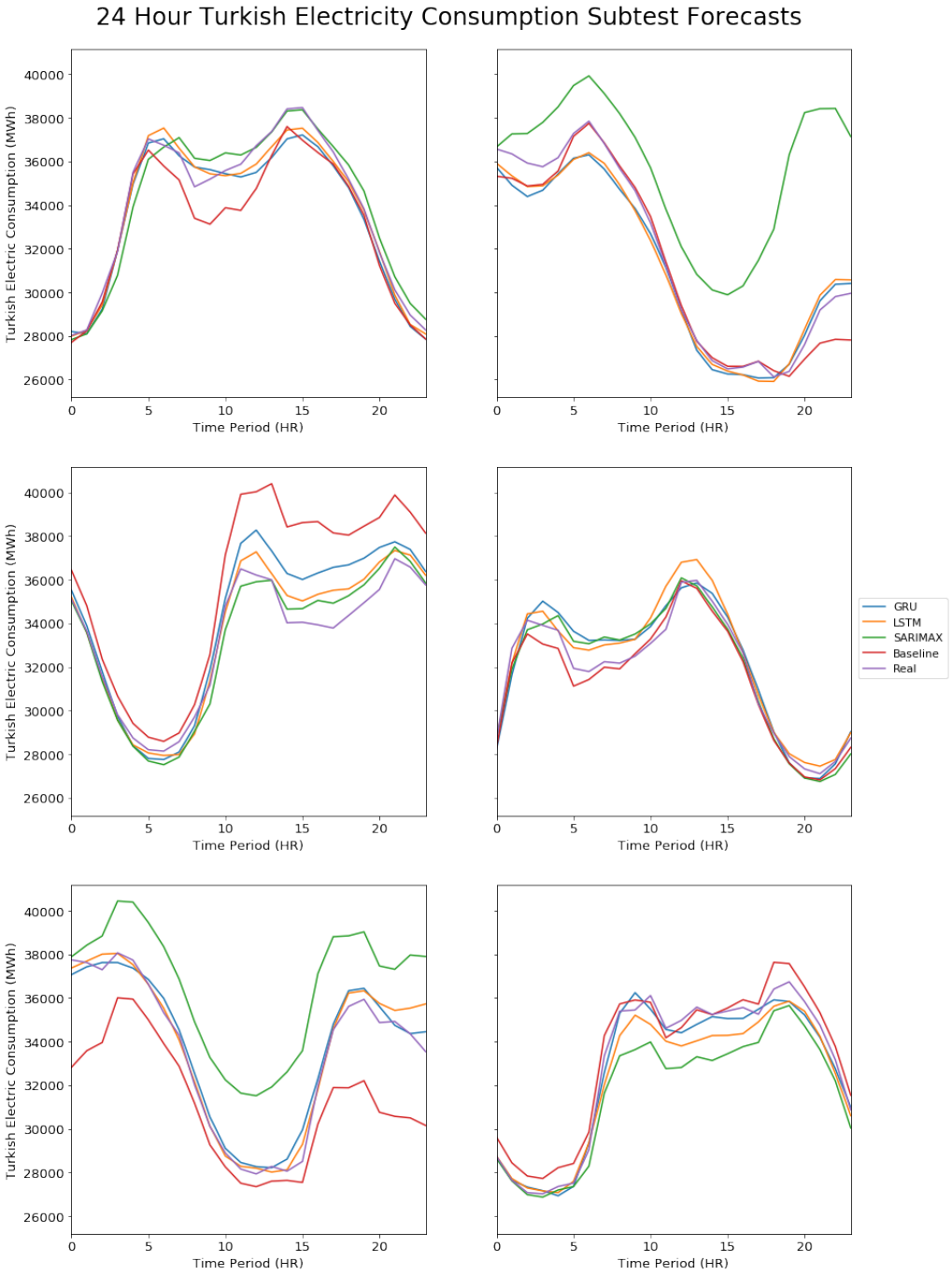

Fig. 5. Six 24-Hour Turkish Electricity Consumption Subtest Forecasts

performance of GRU is better than recently published studies with the same dataset developed for 24-hour forecasts (e.g. see [21]). The SARIMAX model error fluctuates a lot more. Specifically, the SARIMAX model is able to capture some portions of the data very well, but also misses many segments. This can be seen by the larger error bounds on the SARIMAX predictions as shown in Figure 6. If augmented out-of-sample testing had not been used, this pattern in the predictive nature of SARIMAX would not have been evident.

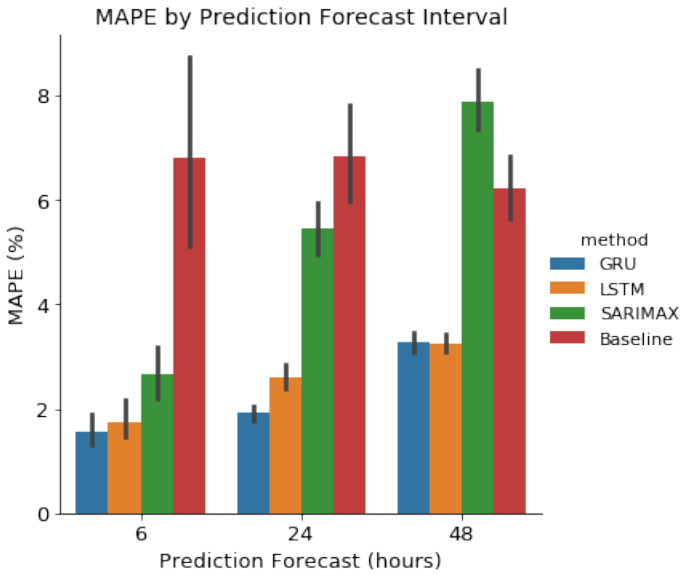

Fig. 6. MAPE by Prediction Forecast Interval

Table 1. MAPE by Prediction Forecast Interval with 95% Error Bounds

Hours	Baseline (%)	SARIMAX (%)	LSTM (%)	GRU (%)
6	6.8 ± 1.8	2.6 ± 0.5	1.8 ± 0.4	1.6 ± 0.3
24	6.8 ± 0.9	5.4 ± 0.5	2.6 ± 0.2	1.9 ± 0.1
48	6.2 ± 0.6	7.9 ± 0.5	3.2 ± 0.2	3.3 ± 0.2

5 Conclusion

By using augmented out-of-sample model comparison, we have found neural networks to outperform classical models to predict electricity consumption rates in Turkey. The augmented out-of-sample method for model comparison alleviates many of the shortcomings that are currently used to compare different time series models. By allowing for more testing on the same dataset, in a realistic manner to real-world training, augmented out-of-sample comparison is able to determine the best algorithm to model a real-world dataset. Beyond the scope of Turkish electricity, this comparison method is flexible to be used in comparing many different time series models. The goal of comparing different models is to see how well a model is able to fit the deterministic components of a specific dataset, without picking up the random components. Due to the empirical nature of this, there is no one-size fits all model. In order to better identify the best model, it is recommended to use augmented out-of-sample forecasting to compare different models.

References

1. Agrawal, R.K., Muchahary, F., Tripathi, M.M.: Long term load forecasting with hourly predictions based on long-short-term-memory networks. In: 2018 IEEE Texas Power and Energy Conference (TPEC). pp. 1–6. IEEE (2018)
2. Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D.C., Rangapuram, S., Salinas, D., Schulz, J., et al.: Gluonts: Probabilistic time series models in python. arXiv preprint arXiv:1906.05264 (2019)
3. Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time series analysis: forecasting and control. John Wiley & Sons (2015)
4. Cheng, Y., Xu, C., Mashima, D., Thing, V.L., Wu, Y.: Powerlstm: Power demand forecasting using long short-term memory neural network. In: International Conference on Advanced Data Mining and Applications. pp. 727–740. Springer (2017)
5. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
6. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: A review. Data Mining and Knowledge Discovery **33**(4), 917–963 (2019)
7. Frisinger, H.H.: Aristotle and his "meteorologica". Bulletin of the American Meteorological Society **53**(7), 634–638 (1972)
8. Gamboa, J.C.B.: Deep learning for time-series analysis. arXiv preprint arXiv:1701.01887 (2017)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
10. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
11. Kane, M.J., Price, N., Scotch, M., Rabinowitz, P.: Comparison of arima and random forest time series models for prediction of avian influenza h5n1 outbreaks. BMC bioinformatics **15**(1), 276 (2014)

12. Kuan, L., Yan, Z., Xin, W., Yan, C., Xiangkun, P., Wenxue, S., Zhe, J., Yong, Z., Nan, X., Xin, Z.: Short-term electricity load forecasting method based on multilayered self-normalizing gru network. In: 2017 IEEE Conference on Energy Internet and Energy System Integration (EI2). pp. 1–5. IEEE (2017)
13. Merh, N., Saxena, V.P., Pardasani, K.R.: A comparison between hybrid approaches of ann and arima for indian stock trend forecasting. Business Intelligence Journal **3**(2), 23–43 (2010)
14. Narayan, A., Hipel, K.W.: Long short term memory networks for short-term electric load forecasting. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 2573–2578. IEEE (2017)
15. Rangapuram, S.S., Seeger, M.W., Gasthaus, J., Stella, L., Wang, Y., Januschowski, T.: Deep state space models for time series forecasting. In: Advances in Neural Information Processing Systems. pp. 7785–7794 (2018)
16. Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T.: Deepar: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting (2019)
17. Tashman, L.: Out-of sample tests of forecasting accuracy: a tutorial and review. Int J Forecasting **16** (01 2000)
18. Taylor, J.W.: Short-term electricity demand forecasting using double seasonal exponential smoothing. Journal of the Operational Research Society **54**(8), 799–805 (2003)
19. Taylor, J.W.: Triple seasonal methods for short-term electricity demand forecasting. European Journal of Operational Research **204**(1), 139–152 (2010)
20. Valipour, M., Banihabib, M.E., Behbahani, S.M.R.: Comparison of the arma, arima, and the autoregressive artificial neural network models in forecasting the monthly inflow of dez dam reservoir. Journal of hydrology **476**, 433–441 (2013)
21. Yukseltan, E., Yucekaya, A., Bilge, A.H.: Forecasting electricity demand for turkey: Modeling periodic variations and demand segregation. Applied Energy **193**, 287–296 (2017)
22. Zhang, G.P.: Time series forecasting using a hybrid arima and neural network model. Neurocomputing **50**, 159–175 (2003)
23. Zheng, J., Xu, C., Zhang, Z., Li, X.: Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network. In: 2017 51st Annual Conference on Information Sciences and Systems (CISS). pp. 1–6. IEEE (2017)