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 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 · Performance measures · Deep learning · ARIMA

1 Introduction

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. 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 such as one shot comparison and random interval testing lack robustness and the testing usually does

not correctly reflect real-world situations. Our augmented out-of-sample model comparison approach 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 Time-series prediction

Earlier studies on time series forecasting focus on linear prediction models such as auto regressive (AR), moving average (MA) and auto-regressive integrated moving average (ARIMA) which predict future values based on a linear function of past observations [2]. Recent advances in artificial neural networks and deep learning allowed researchers to utilize deep structures in time series forecasting. Several recent studies focus on Long Short Term Memory Neural Networks (LSTM) and Gated Reccurent Units (GRU) architecture for forecasting problems in various fields including electricity demand prediction [3, 8]. Further, a recently developed GluonTS package provides a comprehensive deep-learning-based time series modeling environment [1].

2.2 Time-series Model Comparison Techniques

Commonly used methods for comparing time-series prediction models include single forecast testing [7], multiple dataset testing [9] and random test interval sampling [6]. The strengths to single interval tests is the simplicity of implementation as well as the translation to real-world tests. However, this approach allows for lucky one-shot tests to determine the most accurate method. This issue might be averted by expanding the algorithm to predict the future (data points) for multiple datasets. One pitfall to multiple dataset testing is that it is computationally expensive because of the effort to find numerous datasets and training each model on each dataset. As an alternative, random test interval sampling allows all tests to be done on the same dataset, hence there is no need for multiple datasets to have multiple tests. However, disadvantage is that we are no longer predicting the immediate future, as required in most practical applications. As well, we lose the ability to compare against algorithms that are data dependent such as ARIMA, which can not make a prediction at some arbitrary point in the future without the preceding residuals.

To remedy aforementioned issues, Tashman et al. [10] proposed an augmented training approach. The idea is to first train over the dataset, $train_1$, then test the trained model on a varying interval length of the next time period $test_1$. Next, 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. 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. Figure 1 provides a visualization of the augmented training approach.

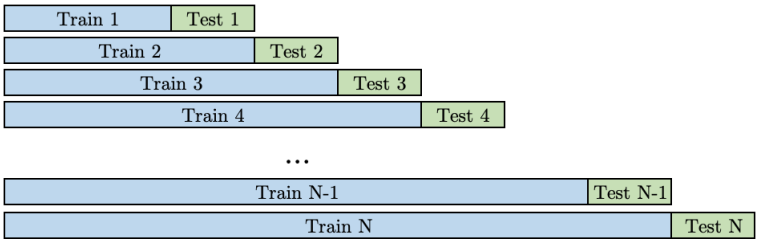

Fig. 1. Train-Test Rolling Visualization

3 Augmented Out-of-sample Comparison Technique

Augmented training method can be used to evaluate numerous models on test datasets. Specifically, the entire 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. In cases where old data should be discarded, a sliding window approach should be used. This would require model retraining, which is a laborious task. Using the proposed expanding horizon approach, the model is instead updated on a small portion of data which prevents full retraining. The proposed approach is summarized in Algorithm 1.

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

$\mathcal{T}, \mathcal{U} \leftarrow$ **TrainTestSplit**$(\mathcal{D})$;
$model \leftarrow$ **TrainUsing**$(f, \mathcal{T})$;
$\{C_j\}_{j=1}^{n} \leftarrow$ **Split**$(\mathcal{U}, n)$;
$results \leftarrow \emptyset$;
for $i = 1 \ldots n$ **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 $results$;

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 and mean squared error. Comparison plots can be made to ensure that outlier data points do not impact the results. If the dataset has 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 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 [5] and GRU [4]. In addition, we implement a variant of the ARIMA model, called SARIMAX (Seasonal ARIMA with Regressors), and a naive baseline model which uses last week's data to predict the current week.

The electricity dataset is broken down into five years worth of hourly data. 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.

4.1 Experimental Settings

In our analysis, each model was fit using an 80-20 train-test split, with 25 individual tests embedded in the test data. Then, three forecast intervals have been chosen: 6 hours, 24 hours, and 48 hours, which allows for 150, 600, and 1200 test points, respectively. We only considered multi-step forecasts, 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. Both LSTM and GRU models contain two layers with 32 and 16 hidden units, which is useful for making a multi-step forecast. In addition, the final 15% of the training data was placed into a validation set. This was done in order to perform cross validation during training. Once the loss in the validation plateaued, training is stopped 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, which can be defined as $\text{MAPE} = 100 \times \frac{1}{N} \sum_{t \in \mathcal{T}} \frac{|\hat{y}_t - y_t|}{y_t}$ 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.

Additional regressors considered were cyclic day-of week and hour-of-day regressors, which are incorporated using sinusoidal transformation. Due to the inherent seasonality, instead of letting the model pick up this information, we

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

pass it to the model explicitly. We observed that addition of these regressors improved the MAPE value by 10-30% depending on the model. Several other potential regressors such as holidays and weather information were not included.

4.2 Benefits of Augmented Out-Of-Sample Testing

Since the subtests were evaluated on many different points, our testing captured many cases. In Figure 2, 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, subtest 3 ranks algorithms from best to worst as SARIMAX, Baseline, LSTM, and GRU, which is different from the actual raking of GRU, LSTM, Baseline, SARIMAX. This comparison would have taken 25 times as long using a sliding window approach. This highlights the benefits of updating models instead of retraining.

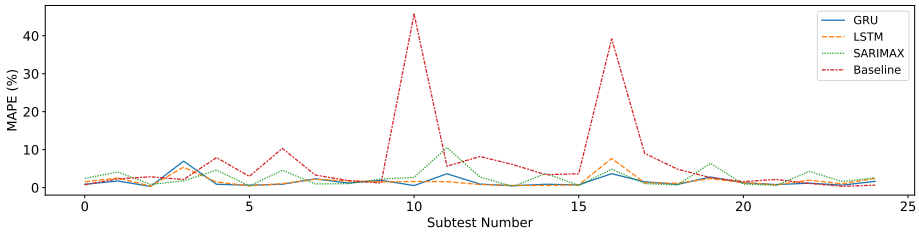

Fig. 2. Subtest MAPEs for 24-hour Forecast

4.3 Results

Table 1 provides a numerical comparison between different models. LSTM and GRU models produce similar results for a 6-hour and a 48-hour forecast, and the GRU performs best for the 24-hour forecast. Both RNN models perform fairly consistently across all the tests with low deviations. In fact, while our objective was not to develop the best possible prediction model, we note that predictive performance of GRU is better than recently published studies with the same dataset (e.g. see [11]). 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 since the SARIMAX model is not well-suited to predict 48 intervals into the future with its relatively low number of parameters. The SARIMAX model error fluctuates a lot more and it misses many segments in the testing, which can be seen by the larger error bounds on its predictions. If augmented out-of-sample testing had not been used, this pattern in the predictive nature of SARIMAX would not have been evident.

5 Conclusion

The augmented out-of-sample method alleviates many shortcomings of the standard approaches that are currently used to compare different time series models.

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

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. In our numerical analysis, we found neural networks to outperform classical models to predict electricity consumption rates in Turkey using the augmented out-of-sample model comparison. Beyond the scope of the electricity dataset, this comparison method is flexible to be used in comparing many different time series models.

References

1. Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., et al.: Gluonts: Probabilistic time series models in python. arXiv preprint arXiv:1906.05264 (2019)
2. Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time series analysis: forecasting and control. John Wiley & Sons (2015)
3. 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)
4. 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)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
6. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
7. 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)
8. 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)
9. 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)
10. Tashman, L.: Out-of sample tests of forecasting accuracy: a tutorial and review. Int J Forecasting **16** (01 2000)
11. Yukseltan, E., Yucekaya, A., Bilge, A.H.: Forecasting electricity demand for turkey: Modeling periodic variations and demand segregation. Applied Energy **193**, 287–296 (2017)