Machine learning methods do have a place in univariate time series forecasting

Evans M. Riba[1], Katherine M. Malan[2][0000−0002−6070−2632], and Edinah Mudimu[2][0000−0002−5479−9255]

[1] University of South Africa, South Africa
evansr@bankseta.org.za
[2] University of South Africa, South Africa
{malankm,mudime}@unisa.ac.za

Abstract. In the time series forecasting community, many believe that machine learning methods perform poorly compared to more traditional statistical techniques. This has been confirmed in the results of the M4 Competition – an open competition comparing the performance of new and established univariate forecasting methods. A problem with these competitions is that the results are aggregated over many datasets (100,000 in the case of the M4 Competition). This fails to recognise that each approach may perform the best on some datasets, but very badly on others. In this paper, two groups of the best-known methods, one for statistical and the other for machine learning, are compared by analysing their performance on a set of 300 time series datasets. Results show that although statistical methods performed the best on 52% of the time series, machine learning methods achieved lower error rates on average over all datasets. When zooming into the performance of methods by frequency of the time series, the machine learning methods performed better than the statistical methods on yearly and quarterly time series. The findings show that there are a significant number of time series datasets where machine learning methods dominate the statistical methods, and more research is needed to understand the features of time series datasets that each method is most suited to solving. These findings also show that there is clear performance complementarity between the different algorithmic approaches, motivating for the future use of metalearning to implement automated algorithm selection for univariate time series forecasting.

Keywords: Univariate time series · Forecasting methods · Machine learning · Statistical methods.

1 Introduction

Time series forecasting, where historical data over time is used to predict future events, is an important part of business planning processes. Examples of time series data include demand for a particular product, or the number of passengers passing through an airport – both examples of univariate time series due to the data consisting of single observations recorded sequentially over time. With advances in hardware and software technology, the quantities of time series data produced have increased significantly. Time series data can occur at different frequencies, for example, electricity consumption at minute intervals, daily closing stock prices, monthly hotel occupation totals, quarterly crime figures, and so on. Regardless of the frequency of the data, the basic principle of time series forecasting is the same: analysing historical data and building some form of model to make the required prediction.

Research on time series analysis and forecasting has been an active area for many decades [7] and continues to attract interest from academics and practitioners from different disciplines. In the early years, simple statistical techniques dominated the field, but more recently, machine learning algorithms, such as artificial neural networks, have also found applicability in time series forecasting [22]. This has led to an increase in the number of time series forecasting methods available to forecasters.

The increased pool of possible forecasting methods to choose from creates a challenge for forecasters. This is because according to the "no-free-lunch theorem" for supervised learning [26], no single method can perform better than all the other methods all the time. The implication is that for any given time series forecasting problem, there are many possible candidate methods to choose from and it is not clear which method will be the best for the problem upfront. The problem is compounded when there are a number of different forecasting problems to solve and each problem could have its own best algorithm. Although machine learning methods have found applicability in time series forecasting, their competitiveness against statistical methods is contested [14, 15].

In this paper, we perform an empirical analysis of the performance of five statistical and five machine learning time series prediction methods on a sample of 300 datasets with three different frequencies. We show that there is performance complementarity between the methods, meaning that each method outperformed all other methods on a subset of the datasets. We also show that there are particular subsets of time series data where machine learning methods perform better than statistical methods. The main contributions of this study are:

1. We demonstrate the existence of a subset of time series datasets for which ML methods are superior to the statistical methods and thus,
2. we show that there is performance complementarity between the ML and statistical methods in univariate time series forecasting.

This paper has four remaining sections. The next section describes the background of the study and also covers the necessary literature underpinning the study. In Section 3, the research methods and tools are presented and results are discussed in Section 4. Lastly, conclusions are discussed in Section 5 including possible future studies.

2 Background

Forecasting methods are generally compared using different metrics as explained by Armstrong [2]. The most common comparison method is based on performance against other methods under consideration. There are a number of research studies that compare the performance of different forecasting methods [1, 11, 17, 25]. These studies compared how well the methods were able to forecast data either from a specific domain [11, 17, 21] or for a specific time series [1].

Studies were also conducted to compare a variety of machine learning and statistical methods [15, 11, 24, 19]. These methods were Compared using different measures of accuracy. In general, machine learning methods were found to be inferior to their statistical counterparts. In their study, Makridakis [15] used the 1,045 monthly time series datasets from the M3 Competition to compare these categories of time series forecasting methods. They recommended that more research is needed in order to understand and improve the performance of machine learning methods.

Using the same dataset as in [15], Cerqueira et al. [5] later found that the machine learning methods performed better than statistical methods when longer time series with at least 1,000 observations were used. Other studies where ML methods were found to be superior to statistical methods include [18, 27, 23, 12].

The debate about which category of methods performs better than the other remains. The proponents of statistical methods argue that machine learning methods perform poorly compared to statistical methods. They argue that studies that suggest the superiority of machine learning are biased and designed to suit them [15]. The other argument is that machine learning methods are rarely compared with statistical or benchmark methods. This has led to an increase in methods which are not necessarily suitable for time series forecasting. The results of the M4 Competition have also stressed the findings made by Makridakis et al. [15] where the machine learning methods did not perform better than the statistical methods [14].

On the other hand, studies like Cerqueira et al. [5] that favour machine learning methods have also pointed out the weaknesses in studies that try to highlight the lack of accuracy of machine learning methods. In the M4 Competition results, Makridakis et al. [16] highlighted the quantities of the data as a possible reason for the poor performance of the machine learning methods. This is due to these methods requiring more computation power than statistical methods.

What is emerging from the differing views is that ML methods cannot be ignored when candidates for forecasting a problem are considered. They have presented themselves as strong contenders for time series prediction applications. In this paper, we compare the two groups of time series forecasting methods by comparing stronger ML methods for time series forecasting with the traditional statistical methods. The stronger ML methods are based on the performance of 10 ML methods in Makridakis et al. [15].

3 Experimental Methodology

Different time series forecasting methods were selected to represent the two categories of forecasting methods. The top four ML methods were selected based on findings by Makridakis et al. [15]. These methods are multi-layer perceptron (MLP), support vector regression (SVR), Gaussian processes (GP) and Bayesian neural network (BNN). In addition, the neural network autoregression (NNERTAR) method was also included as part of the ML methods. The statistical methods selected were autoregressive integrated moving averages (ARIMA), exponential smoothing (ETS), the theta forecasting method (THETAF) [3], TBATS (T:trigonometric seasonality, B: Box-Cox transformation, A: ARIMA errors, T: trend, S: seasonal components) [8], and random walk forecasting with drift (RWF). The statistical methods used were selected based on their popularity and availability in the *forecast* package in R.

3.1 Data and Sampling

The datasets used for this study come from the Makridakis M4 Competition [16]. The M4 Competition is one of a series of competitions organised since 1982 by the researcher Spyros Makridakis[3]. The competitions are intended to gauge the advances in time series forecasting methods. The M4 Competition contained 100,000 time series datasets from different backgrounds and time horizons; in particular 48,000 monthly time series datasets, 24,000 quarterly time series datasets, 23,000 yearly time series datasets, and smaller numbers of weekly, daily and hourly datasets [16].

For this study, a random sample of 300 time series was selected with 100 series from each of the monthly, quarterly and yearly categories. The time series in the sample had different numbers of observations; the shortest consisted of 19 data points, the longest had a length of 918 and the average length of all the time series in the training set was 166. Table 1 shows a summary of the datasets by the frequency of the time series and highlights the differences between the time series of the three frequencies. The monthly time series had many more data points that could be used for training (median of 365) than the quarterly and yearly time series with medians of 59 and 37, respectively.

Table 1. Summary of the 300 time series datasets used in the study.

Frequency	Number of Observations			Number of
	Minimum	Median	Maximum	Test Observations
Monthly	82	365	918	18
Quarterly	25	59	271	8
Yearly	19	37	84	6

[3] https://mofc.unic.ac.cy/history-of-competitions/

The M4 Competition dataset was partitioned into a training set and a test set by the competition organisers. The test set was initially hidden from the public and became available only after the competition had closed. For the monthly time series, the next 18 observations following on from the training series were set aside for testing. The next eight observations of the quarterly series were kept for testing, while only six were used for testing in the yearly series. For this study, the training and test sets of the sub-sample of 300 time series datasets were used exactly as they were used in the competition.

3.2 Software and Tools

This study used the open source software, R for statistical computing [20]. R is widely used for statistical analysis purposes and includes many useful analysis extensions shared by users. The R packages that were used to accomplish the objectives of this research include the *forecast* package written by Hyndman et al. [9] that implements different statistical forecasting methods in R.

Table 2 shows a summary of the different methods used in this paper including the R packages and functions used to implement the methods.

Table 2. Summary of the five machine learning and five statistical methods used in this study with the relevant R package and function used for implementation.

Method Category	Method	Description	R Package	Function
Machine Learning	MLP	Multi-layer perceptron	*neuralnet*	*neuralnet()*
	SVR	Support vector regression	*e1071*	*tune.svm()*
	GP	Gaussian processes	*kernlab*	*gausspr()*
	BNN	Bayesian neural network	*brnn*	*brnn()*
	NNETAR	Neural network autoregression	*forecast*	*nnetar()*
Statistical	ARIMA	Autoregressive integrated moving averages	*forecast*	*auto.arima()*
	ETS	Exponential smoothing	*forecast*	*ets()*
	THETAF	Theta forecasting	*forecast*	*thetaf()*
	TBATS	Trigonometric, Box-Cox, ARMA, trend & seasonality	*forecast*	*tbats()*
	RWF	Random walk forecasting with drift	*forecast*	*rwf()*

3.3 Data Preprocessing

Preprocessing was done for all the 300 time series datasets based on some of the main findings and recommendations by Makridakis et al. [15]. The time series datasets were first transformed using the Box-Cox transformation [4] to remove non-stationarity in the variance of the series.

The monthly and quarterly time series data with seasonality were further deseasonalised through the application of the STL (seasonal and trend decomposition using loess) method proposed by Cleveland et al. [6]. All the time series were further differenced to remove trend components in the series.

Lastly, the observation points y of each time series were normalised using min-max scaling to prepare them for application in the ML methods, using the formula:

$$y' = \frac{y - y_{min}}{y_{max} - y_{min}}, \tag{1}$$

where y_{min} is the minimum and y_{max} is the maximum data point in the time series.

In order to determine the number of input nodes for ML methods, a neural network autoregression model was fitted to each time series. The method is implemented in the *forecast* package in R. The number of input nodes determined by the autoregression model was then used to create an $(n - p) \times p$ matrix of lagged variables where n is the length of the series and p is the number of input nodes. These lagged variables were then used as inputs to the ML methods. The other parameters of the ML methods were left at their default values and allowed the functions of the different R packages to optimise the parameters. The only exception was with the support vector regression where the hyperparameter γ was chosen from a grid of values ranging from 0 to 10 at a rate 0.01. This was done to reduce the tuning time for the method.

3.4 Data Analysis

The different methods were divided into two groups. The first group represents the ML methods while the second group is made up of statistical methods as shown in Table 2. A naïve method, random walk with drift (RWF) was also included in the statistical methods.

All the methods were trained on each time series dataset in the sample using the predetermined training set from the competition. After training the method on the n training data points, the test data was then used to calculate the performance of the method. The one-step-ahead forecast f_{n+1} for each time series dataset was obtained using the training set. In order to obtain the multi-step-ahead forecasts $(\mathrm{f}_{n+2}, \mathrm{f}_{n+3}, \ldots)$, the previous periods' forecasts were added and used. Figure 1 demonstrates this process. Since some of the ML methods are stochastic in nature, multiple forecasts were generated (30 runs) for these methods and the results averaged to arrive at the performance measure obtained. In this case, the stochastic ML methods are MLP, GP, BNN and NNETAR.

The performance measures used in this study are the symmetric mean absolute percentage error (sMAPE) [13] and the mean absolute scaled error (MASE).

Measure sMAPE, was proposed by Makridakis [13] to overcome some of the drawbacks encountered with mean absolute percentage error (MAPE), such as the measure not being defined for data points that can take on zero values. sMAPE is expressed as a percentage and is calculated as:

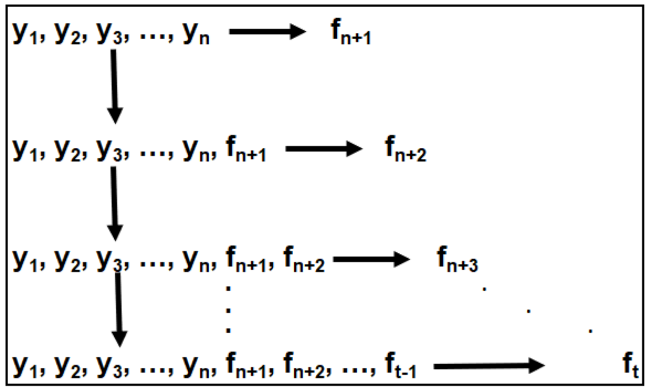

Fig. 1. Multi-step-ahead forecasts generation process for the forecasting methods under comparison. The first forecast in the test set is based on the data points of the training set. To predict the next forecast, the first forecast is added to the end of the training series, and so on.

$$\text{sMAPE} = \frac{200}{n} \sum_{t=1}^{n} \frac{|e_t|}{|y_t| + |f_t|}, \tag{2}$$

where e_t is the error at time t (the difference between the actual and predicted observation), y_t is the actual observation, and f_t is the forecast at time t.

The other measure used in this study, MASE, was proposed by Hyndman and Koehler [10] as a scale-independent measure that compares the performance of a forecasting method with a naïve benchmark model by scaling the errors of the forecasting method using the one-step ahead mean absolute errors of the benchmark model. In this study, the seasonal naïve method was used as a benchmark. A seasonal naïve model simply uses the previous observation from the corresponding season as the forecast for the current period, that is, $\mathrm{f}_t = y_{t-s}$, where s is the length of the season. This allows the forecaster to determine if the method is worth pursuing. The errors are first scaled based on the mean absolute error (MAE) of a naïve model for non-seasonal time series as follows [10]:

$$q_t = \frac{e_t}{\frac{1}{n-1} \sum_{i=2}^{n} |y_i - y_{i-1}|}. \tag{3}$$

For seasonal time series, the scaled errors are determined by using the following:

$$q_t = \frac{e_t}{\frac{1}{n-s} \sum_{i=s+1}^{n} |y_i - y_{i-s}|}, \tag{4}$$

where e_t is the error calculated as the difference between the actual and the forecast value, s is the length of the season and n is the length of the time series. Then MASE is simply calculated as

$$\text{MASE} = \frac{\sum_{t=1}^{n} |q_t|}{n}. \tag{5}$$

For each time series dataset, the methods were ranked according to their performance. The method with the lowest value is considered the best method compared to the others. The group that contains the method with the lowest value for that time series was taken as the recommended group to forecast that time series.

In comparing the two groups of methods, the following were used:

- Values of sMAPE and MASE were obtained for the individual methods per time series.
- The minimums for each group were taken to represent the performance of the group for that time series.
- An average across all the time series datasets was calculated per group.
- Results were then used to compare the performance of the two groups.

4 Results

Based on the 300 time series datasets selected, the ML methods performed better than the statistical methods on this sample of time series. The ML group achieved a combined sMAPE of 6.88%, whereas the statistical methods achieved a higher sMAPE of 8.21%. The two groups were compared by using the minimum sMAPE for each group of time series. This means for a given time series dataset, the sMAPE of a group is taken as the minimum sMAPE across the different members of the group.

To confirm these results, the alternative metric, MASE was also calculated. Recall that MASE includes a naïve method as a benchmark in the measure. A lower value for MASE is indicative of better performance. Comparing the average minimum MASE values of each group, ML methods achieved MASE of 1.57, compared to 2.24 for statistical methods.

Figure 2 shows the box plots of the performance of these two groups of time series forecasting methods based on sMAPE. The chart shows that the statistical methods have a higher variability than the ML methods. They are also more skewed to the right as the mean has shifted away from the median. Box plots for average performance across the group members were also compared (Average_ML and Average_Stats). The variability of the data increases when using the average across the groups with degrading performance for both groups. In calculating the average across the methods, the random walk forecasts were not considered as the method produced poor performance and this could be observed in Figure 3.

Figure 3 shows the box plot map of the individual methods based on sMAPE. Notably, the random walk method has performed far worse than any method

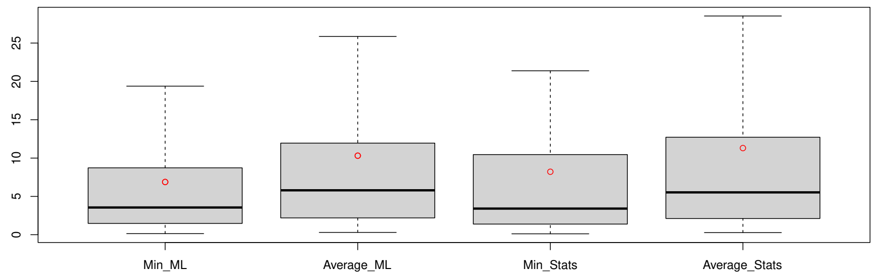

Fig. 2. Boxplots showing the distribution of sMAPE values of the best performing ML and statistical method on each dataset. Each red point corresponds to the mean. This shows that the machine learning algorithms performed slightly better on average and had slightly lower variance as a group than the statistical approaches.

in the chart. The SVR has a smaller range compared to the other methods, indicating consistency in performance across the different time series.

Figure 4 shows the sMAPE values of the individual forecasting methods. The SVR method performed better than all the other methods with sMAPE of 8,99%. This was followed by the GP method with sMAPE of 9,14%. The third best-performing method was the ARIMA method with sMAPE of 9,81%. As expected, the worst-performing method is the random walk method.

Figure 5 shows the sMAPE values of the best method per time series. Each dot in the diagram represents a single time series (numbered from 1 to 300 on the horizontal axis) with the sMAPE value (vertical axis) of the best performing method on that time series. The colour indicates whether the best-performing method was a ML or a statistical approach. The performance of both forecasting method groups shows high variability and deterioration for the last 100 time series (from 200 to 300), corresponding to the yearly time series. The first 200 time series, which are monthly and quarterly time series, respectively, produced better results for both groups of methods. The variation appears to be higher for monthly time series compared to the quarterly data.

Table 3 summarises the performance of individual methods on all time series. The table shows that statistical methods performed better than the ML methods on 52% of the time series datasets. However, the statistical methods also produced the worst results on 82% of the time series. This is due to the worst performance achieved by the random walk method. The method with a higher percentage of wins is the SVR (ML method) with 16% of the time series. This was closely followed by a statistical method (TBATS) with 15% of the time series. The significance of these results is in the evidence of performance complementarity between the methods. Each method (even the RWF) out-performs all

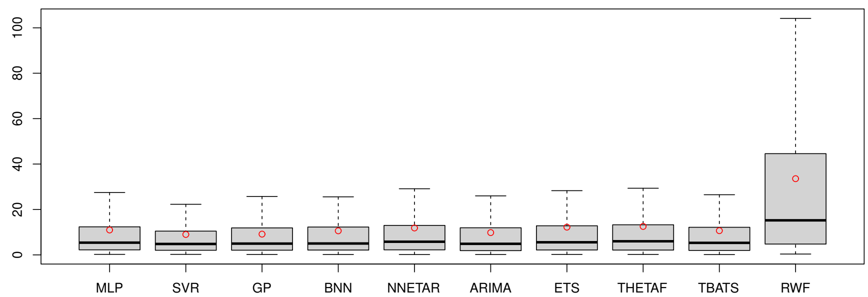

Fig. 3. Boxplots showing the test performance of different methods based on sMAPE. The red points correspond to the means. From the left, the first five are machine learning methods, followed by the statistical methods. This shows that the random walk (RWF) method overshadows the others in terms of bad performance.

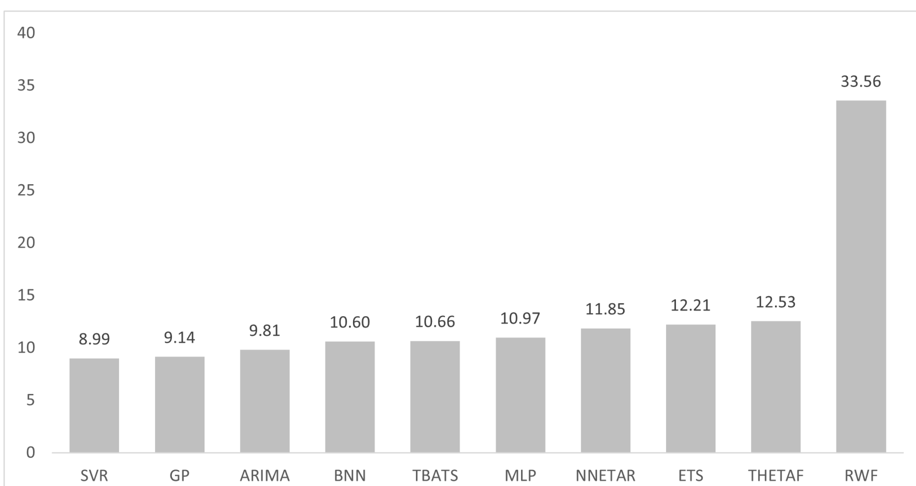

Fig. 4. Methods sorted in ascending order of mean sMAPE, showing that SVR was the best-performing method on average.

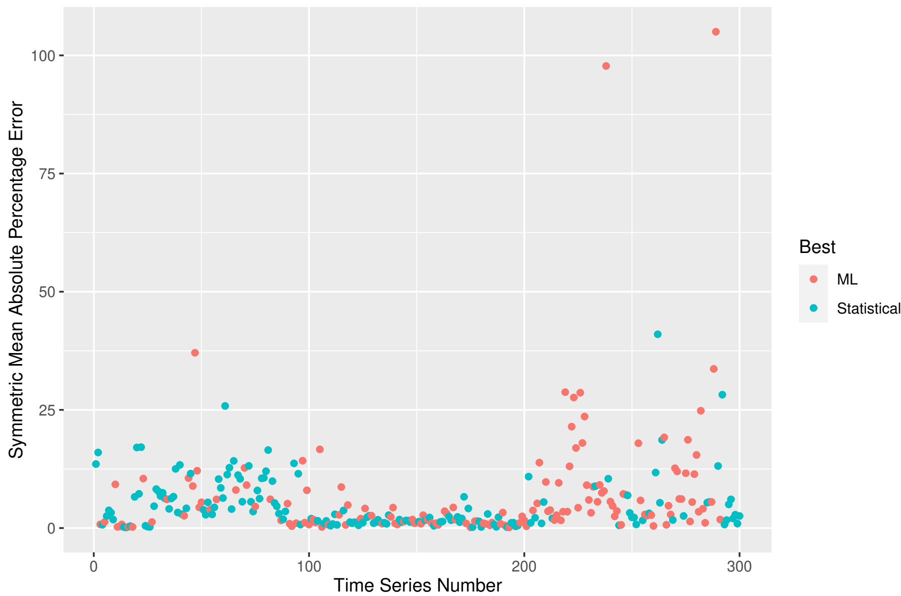

Fig. 5. Multi-step-ahead forecasts performance of the two groups of methods based on sMAPE. The first 100 dots represent the monthly time series followed by quarterly and lastly, the yearly time series represented by the last 100 dots.

other methods on a significant number of time series (represented by the 'Count Best' and '% Best' values in the table). These results motivate for the future use of metalearning to implement automated algorithm selection between different methods based on the features of time series.

Figure 6 shows the performance of the two groups of forecasting methods by length and frequency of the time series. The two groups of methods have performed well on quarterly time series with sMAPE of 2.1% for ML methods and 2.2% for statistical methods. On monthly data, the methods also show similar levels of performance with ML methods performing slightly worse than the statistical methods. Both methods deteriorated in performance on yearly data compared to the other frequencies (monthly and quarterly). Of significant importance is the difference in the performance of these groups on yearly time series. The ML methods performed far better with sMAPE of 10.7% compared to 15.5% for statistical methods. This is surprising considering that the yearly time series data have the lowest average length of 42 training data points compared to the other frequencies. The length of the time series is high for the monthly time series with an average of 366, followed by the quarterly time series with an average length of 90.

Table 3. Group and individual performance of the different forecasting methods illustrating that each method out-performed all other methods on a subset of the time series. Although RWF was the worst performing method overall, it was still the best-performing method on 19 of the 300 time series considered in the study.

Method Category	Method	Count Best	%Best	Count Worst	% Worst
Machine Learning	MLP	11	4%	10	3%
	SVR	47	16%	8	3%
	GP	35	12%	16	5%
	BNN	16	5%	5	2%
	NNETAR	35	12%	19	6%
	Combined	144	48%	58	19%
Statisitcal	ARIMA	36	12%	5	2%
	ETS	14	5%	3	1%
	THETAF	41	14%	18	6%
	TBATS	46	15%	12	4%
	RWF	19	6%	204	68%
	Combined	156	52%	242	81%
Total		300	100%	300	100%

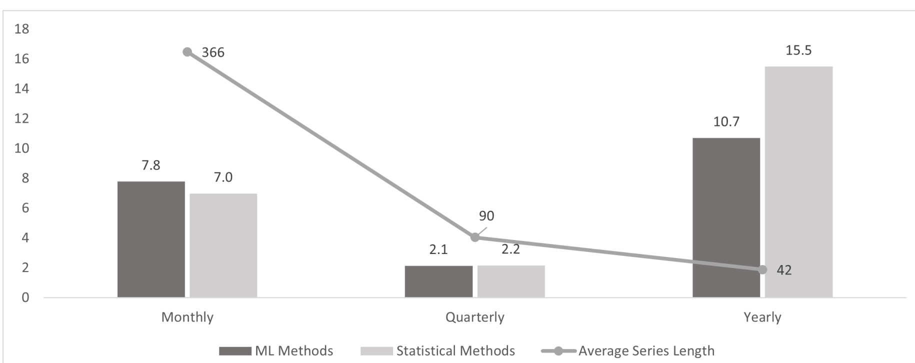

Fig. 6. Comparison of the performance of the two groups by time series frequency on sMAPE.

5 Conclusions

In this paper, we have demonstrated that the ML methods are capable of performing better than their statistical counterparts. This shows that there exists a subset of time series datasets where machine learning methods are able to produce results that are far superior to the statistical methods. This further proves that there is performance complementarity between ML and statistical methods in time series forecasting. Therefore, there is a need to study this subset of time series data in order to establish mechanisms that could assist forecasters in considering these methods for evaluation when selecting appropriate forecasting methods. Such a mechanism could assist the forecaster in determining whether to focus on ML methods or not in advance considering that these methods require a lot of computing time. Future studies could look at the feature design of this subset of time series where ML methods are superior and design a recommendation system for such time series.

References

1. Ajiono, A., Hariguna, T.: Comparison of three time series forecasting methods on linear regression, exponential smoothing and weighted moving average. International Journal of Informatics and Information Systems **6**(2), 89–102 (2023)
2. Armstrong, J.S.: Principles of forecasting: A handbook for researchers and practitioners, chap. Evaluating forecasting methods, pp. 443–472. Springer (2001)
3. Assimakopoulos, V., Nikolopoulos, K.: The theta model: a decomposition approach to forecasting. International Journal of Forecasting **16**(4), 521–530 (2000)
4. Box, G.E., Cox, D.R.: An analysis of transformations. Journal of the Royal Statistical Society: Series B (Methodological) **26**(2), 211–243 (1964)
5. Cerqueira, V., Torgo, L., Soares, C.: A case study comparing machine learning with statistical methods for time series forecasting: size matters. Journal of Intelligent Information Systems **59**(2), 415–433 (2022)
6. Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.: STL: A seasonal-trend decomposition. Journal of Official Statistics **6**(1), 3–73 (1990)
7. De Gooijer, J.G., Hyndman, R.J.: 25 years of time series forecasting. International Journal of Forecasting **22**(3), 443–473 (2006)
8. De Livera, A.M., Hyndman, R.J., Snyder, R.D.: Forecasting time series with complex seasonal patterns using exponential smoothing. Journal of the American Statistical Association **106**(496), 1513–1527 (2011)
9. Hyndman, R.J., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., Petropoulos, F., Razbash, S., Wang, E.: Package 'forecast' (2020), https://cran.r-project.org/web/packages/forecast/forecast.pdf
10. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. International Journal of Forecasting **22**(4), 679–688 (2006)
11. Jankowitz, M.D., Riba, E.M.: Exploring ARIMA, ANN, SVM and regression with ARIMA errors forecasting methods with application to air passenger data. ICIC Express Letters **16**(10), 1063–1070 (2022)
12. Kandananond, K.: A comparison of various forecasting methods for autocorrelated time series. International Journal of Engineering Business Management **4**, 4 (2012)

13. Makridakis, S.: Accuracy measures: theoretical and practical concerns. International journal of forecasting **9**(4), 527–529 (1993)
14. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: The M4 Competition: Results, findings, conclusion and way forward. International Journal of Forecasting **34**(4), 802–808 (2018)
15. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: Statistical and machine learning forecasting methods: Concerns and ways forward. PloS One **13**(3) (2018)
16. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: The M4 Competition: 100,000 time series and 61 forecasting methods. International Journal of Forecasting **36**(1), 54–74 (2020)
17. Mishra, P., Al Khatib, A.M.G., Sardar, I., Mohammed, J., Karakaya, K., Dash, A., Ray, M., Narsimhaiah, L., Dubey, A.: Modeling and forecasting of sugarcane production in india. Sugar Tech **23**(6), 1317–1324 (2021)
18. Okasha, M.K.: Using support vector machines in financial time series forecasting. International Journal of Statistics and Applications **4**(1), 28–39 (2014)
19. Parmezan, A.R.S., Souza, V.M., Batista, G.E.: Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. Information Sciences **484**, 302–337 (2019)
20. R Core Team: R: A Language and Environment for Statistical Computing. https://www.R-project.org/ (Accessed 2 September 2023)
21. Siami-Namini, S., Tavakoli, N., Namin, A.S.: A comparison of ARIMA and LSTM in forecasting time series. In: 2018 17th IEEE international conference on machine learning and applications (ICMLA). pp. 1394–1401. IEEE (2018)
22. Tealab, A.: Time series forecasting using artificial neural networks methodologies: A systematic review. Future Computing and Informatics Journal **3**(2), 334–340 (2018)
23. Tiwari, S., Kumaraswamidhas, L., Garg, N.: Comparison of SVM and ARIMA model in time-series forecasting of ambient noise levels. In: Advances in Energy Technology: Select Proceedings of EMSME 2020. pp. 777–786. Springer (2022)
24. Vivas, E., Allende-Cid, H., Salas, R.: A systematic review of statistical and machine learning methods for electrical power forecasting with reported mape score. Entropy **22**(12), 1412 (2020)
25. Wang, W.C., Chau, K.W., Cheng, C.T., Qiu, L.: A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. Journal of hydrology **374**(3-4), 294–306 (2009)
26. Wolpert, D.H.: The supervised learning no-free-lunch theorems. Soft Computing and Industry pp. 25–42 (2002)
27. Yang, M., Wang, J.: Adaptability of financial time series prediction based on bilstm. Procedia Computer Science **199**, 18–25 (2022)