Stacked LSTM Snapshot Ensembles for Time Series Forecasting

Sascha Krstanovic and Heiko Paulheim

University of Mannheim, Germany
Research Group Data and Web Science
sascha@informatik.uni-mannheim.de
heiko@informatik.uni-mannheim.de

Abstract. Ensembles of machine learning models have proven to improve the performance of prediction tasks in various domains. The additional computational costs for the performance increase are usually high since multiple models must be trained. Recently, snapshot ensembles [13] gained attention as they provide a comparably cheap way of ensemble learning for artificial neural networks (ANNs). We extend snapshot ensembles to the application of time series forecasting, which comprises two essential steps. First, we show that determining reasonable selections for sequence lengths can be used to efficiently escape local minima. Additionally, combining the forecasts of snapshot LSTMs with a stacking approach greatly boosts the performance compared to the mean of the forecasts as used in the original snapshot ensemble approach. We demonstrate the effectiveness of the algorithm on five real-world datasets and show that the forecasting performance of our approach is superior to conservative ensemble architectures as well as a single, highly optimized LSTM.

Keywords: Time Series, LSTM, ARIMA, Ensembles, Stacking, Meta-Learning

1 Introduction

Estimating the future development of continuous data generated by one or more signals has been an ongoing research field of interest for various applications. For example, automated financial forecasting is vital in today's markets. Further, sensor generated data driven by the Internet of Things requires robust methods for reliable forecasts of temporal data. Long Short-Term Memory (LSTM) [10] has proven to be an effective method for a variety of sequence learning tasks such as time series forecasting. Relying on a single LSTM, however, is prone to instability due to the dynamic behavior of time series data. Additionally, the optimization of LSTM parameters is a hard problem that requires time intensive fine tuning.

Another difficulty when dealing with time series problems lies in the slicing of the data, i.e., how many past values should be considered for training the model

and generating forecasts. It is common practice to determine the top periodicity using a fast fourier transformation and power spectra, and train one or more models based on that periodicity. This approach is prone to incompleteness because information may be encoded across patterns of varying periodicity in the series. It is also a time consuming task as identifying the optimal sequence length is usually part of a manual preprocessing step. For these reasons, it is a challenge to create machine learning frameworks that are able to produce automated forecasts for a given series. Even a greatly tuned model fails to find important relationships in time series data if the selected time lags can not represent these patterns. Therefore, a framework that can incorporate multiple sequence lengths is desirable.

We introduce a meta learning approach based on snapshot ensembles that provides superior and robust forecast estimates across different datasets. In contrast to the original idea of snapshot ensembles, we do not adapt the parameters of the LSTM but leave them unchanged. Instead, we use different slices of the training data in order to escape local minima and to detect time-dependent patterns. Our proposed approach enables the automated generation of time series forecasts for a given series $y_1, ..., y_n$, including preprocessing steps like data standardization, periodicity detection, data slicing and splitting. Hence, the amount of required manual work is greatly reduced by the proposed framework.

By sequentially training LSTMs with periodicities of decreasing strength, our algorithm is able to learn the different patterns of the respective seasonalities. This allows for higher generalization of the final model, thereby providing estimates that are robust with respect to the underlying data generation process.

The rest of this paper is structured as follows. Section 2 provides an overview of existing approaches to time series forecasting and their application within ensemble frameworks. In Section 3, we introduce the concept of snapshot ensembles and explain our approach for their extension to the task of time series forecasting. We show that our method outperforms previous approaches on five datasets in Section 4. Eventually, we conclude and give an outlook on future research directions in Section 5.

2 Related Work

Time series forecasting is a highly common data modeling problem since temporal data is generated in many different contexts. Classical forecasting approaches are based on autoregressive models such as ARIMA, ARIMAX, and Vector Autoregression (VAR) [7] [20]. Here, a forecast estimate is dependent on a linear combination of the past values and errors. Autoregressive models work well if the assumption of stationarity is true and the series is generated by a linear process [1]. On the other hand, these hard assumptions limit the effectiveness of autoregressive models if one deals with non linear series, as it is the case with the majority of practical time series problems.

LSTM, a particular variant of artificial recurrent neural networks (RNN), overcomes these shortcomings as it makes no assumptions about the prior dis-

tribution of the data. One can think of RNNs as regular feed-forward networks with loops in them. This enables RNNs to model data with interdependencies such as autoregression. It has been shown that artificial neural networks with one hidden layer can, in theory, approximate a continuous function arbitrarily well [11]. As the RNN gets deeper, vanishing or exploding gradients often lead to poor model performance [4] [18]. LSTMs solve this problem with a gating mechanism that controls the information flow in the neurons. LSTMs show superior performance in a variety of sequence learning tasks such as machine translation [8] [21].

Since autoregressive models perform well for linear series and neural networks for non linear data, there exist a number of hybrid approaches that make use of these characteristics. In those cases, the data is first split into a linear and a non linear component and each one is modeled independently. The individual results are then combined additively to determine the final estimate [2] [3] [22] [24].

The sequential nature of LSTMs has led to them being studied in the context of time series forecasting intensively. [6] [15] [17] describe applications of LSTMs for forecasting tasks. [1] [14] propose frameworks of LSTM ensembles with independently trained models. Finally, snapshot ensembles constitute a way to construct an ensemble of dependent ANNs at comparably low computational costs. A more detailed description is given in Sec. 3.1. We extend this method to recurrent neural networks and sequential problems.

Time series analysis has also been investigated in the framework of convolutional neural networks (CNNs). [30] use an architecture inspired by the recent success of WaveNet for audio generation [31] which achieves competitive forecasting performance with relatively little training data available. A probabilistic approach that combines both RNNs and CNNs in a single framework is given in [32].

Finding periodicities in time series data is a key part in the preprocessing of time series data and proposes a major challenge for the automation of machine generated forecasts. [5] propose a variation of the approximate string matching problem for automated periodicity detection. [26] develop strategies on diversity generation and build ensembles of the resulting models. In [27], a number of heterogeneous models are arbitrated by a meta learner. [29] apply Fourier transformations to the original data for feature generation and use a feed forward neural network for the modeling part based on these features. [28] shift CNN training entirely to the Fourier domain, thereby, achieve a significant speedup with practically no loss of effectiveness. Another approach that exploits fourier transformations is given in [19]. We will use a similar methodology in the course of this paper.

3 Time Series Forecasting and Snapshot Ensembles

Time series data is subject to a number of properties due to interdependencies across observations:

1. Autoregression. In contrast to a machine learning setup where observations are independent from one another, sequence learning tasks are characterized by dependencies between observations. This has effects on data sampling and model evaluation as drawing completely random subsamples is not possible. Hence, a suitable sample strategy is indispensable when modeling temporal data.
2. Structural patterns and changes. Due to trend and seasonality effects, the behavior of a time series is subject to repetition and change at the same time. While similar patters may repeat over time, the frequency and intensity of those are usually not constant. This is one reason why ensemble methods are a powerful tool for time series data as each of the snapshot models incorporates information of different behavior.

3.1 Introduction to Snapshot Ensembles

Snapshot Ensembles propose a novel technique to obtain an ensemble of ANNs at the same computational costs as fully training a single ANN. The central idea is that instead of training a number of independent ANNs, only one ANN must be optimized. In the process of optimization, the ANN converges to a number of different local minima. Every time the ANN reaches a local minimum, the model snapshot is stored along with its architecture and weights. The final weights of a snapshot serve as the weight initialization of the succeeding snapshot LSTM. Finally, each snapshot provides a prediction estimate and the ensemble predictor is calculated as the mean of the snapshot estimates. It was shown that this combination yields advantageous performance compared to the single best estimate [13].

3.2 Extending Snapshot Ensembles to Sequence Problems

Time series forecasting can be interpreted as a sequence learning problem. Given an input sequence of scalars, the objective is to estimate the succeeding values of the sequence. An important task is to determine how many past values should be considered as the features under consideration, i.e., which slice dimension of the series allows for good model generalization. By nature, time series data is dynamic and subject to change over time, so an initial decision is not necessarily a sustainable solution. Designing ensembles of LSTM networks allows us to incorporate multiple sequence lengths into our prediction model. In the following, we explain how.

LSTMs with varying sequence lengths By architecture, LSTMs are only capable to process sequences of equal lengths per epoch, due to the required matrix operations in the optimization process. In many applications, however, varying sequence lengths are inevitable. One example is machine translation where the length of an input sentence can be arbitrarily long [21]. Padding is usually used to overcome that problem [12]. This implicitly means that, although two models

trained with even slightly different sequence lengths have a large intersection of training data, they learn different yet related patterns. This constitutes a promising setting for ensemble learning.

Locating candidate sequence lengths In order to train a number of snapshot LSTMs with different sequence lengths, the first step is to identify the right choices of these. A naive approach is to select sequence lengths from a random distribution. To get sequence lengths that can catch effects of seasonality, we apply a fast fourier transformation (FFT) to the training data and estimate the power spectra [23]. The motivation behind this is that the FFT is an efficient method to extract the right periodicities from a given time series. This allows the snapshots to encode different patterns, seasonalities, and other time-dependent effects in the series.

Generating a snapshot ensemble of LSTMs with varying sequences [13] conduct a variant of simulated annealing in order to adapt the learning rate and escape from local minima. In this case, a snapshot is a further optimization of its predecessor using the identical training data, which leads to a relatively low level of diversity across the snapshots. We propose another strategy in order to increase diversity: Instead of adapting the model parameters, we feed the LSTM with different slices of the data. This is possible because the dimensions of the training data must be identical within a single epoch but not for two separate epochs. Given a set $S = \{s_1, s_2, ..., s_n\}$ of different sequence lengths we store in total n snapshots of the LSTM. After each snapshot based on s_i, the training process is continued with a different data slice through time according to s_{i+1}. The final holdout estimates of the individual snapshots are commonly combined by taking the mean of the base forecasts. This assumes that each snapshot is equally important with respect to the combination of forecasts. In order to allow for more flexibility, we extend the mean function by a meta learner. Ridge Regression has proven to be an effective choice here [25]. The process of the ensemble construction at training time is depicted in Fig. 1 for the example case $S = \{14, 21, 28\}$ and a forecasting horizon of 10. First, the training data $y_1, ..., y_n$ (75% of the total data) is split according to the most potent sequence lengths provided by the FFT (in decreasing order of FFT significance). In our experiments, we use the top 20 sequence lengths. Next, the first snapshot is trained with the respective data slices based on the first sequence length. We train each snapshot for five epochs and standardize the data by its z-transform prior to training. The base LSTM learners' architecture is set up of two LSTM layers with 64 and 128 neurons as well as 20% dropout. Adam is used as the optimizer with a learning rate of 0.001. The weight matrix of the first snapshot is then updated based on the data slices for the second sequence length, and so on. In total, training is done for $5 \cdot 20 = 100$ epochs. After all snapshots are trained, a ridge regression meta model learns how to combine the individual forecasts of the 20 snapshots. Analogously, at test time, all 20 base models provide their

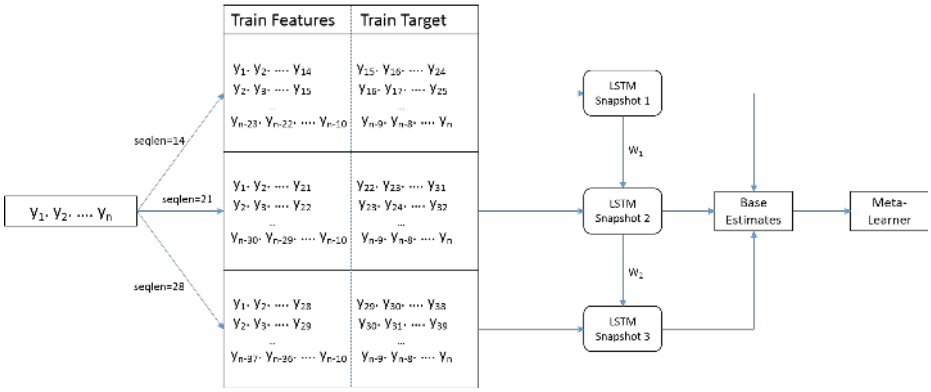

Fig. 1. LSTM Snapshot Training Framework

forecasts to the meta learner, which then combines them to the final estimate for the 10 step ahead forecasts.

4 Experiments

We test the proposed methodology on five data sets of different kind. We train a snapshot ensemble for each data set where we start with the strongest periodicity according to the FFT. Subsequently, each LSTM snapshot is based on the next strongest periodicity. In total, 20 snapshots are trained. An overview of the datasets is given in Table 1 and Fig. 2. Furthermore, Fig. 3 displays the power spectrum for the sunspots series. This example shows that there exist a number of unequally well suited periodicities. Each of these contains different patters which we aim to extract using snapshot ensembles. To show the effectiveness as well as the efficiency of our approach, the performance of the snapshot ensemble is measured against the following three baselines:

1. Independent LSTM ensemble. Instead of continuing the training process by escaping from a local minimum, the LSTM is reinitialized randomly and fed with the new data slices. Instead of n snapshots, we end up with n LSTMs whose training process was completely independent of one another. In contrast to this, a snapshot inherits its initial weights from its preceding snapshot.
2. Single optimized LSTM. The best sequence length according to the FFT is used for the optimization of a single LSTM over all epochs.
3. ARIMA with model selection based on the AIC.

Notably, the total number of epochs is identical for all the neural net approaches. Due to different slices of the training data, the total runtime of the latter approach can slightly differ from the ensemble methods in either direction.

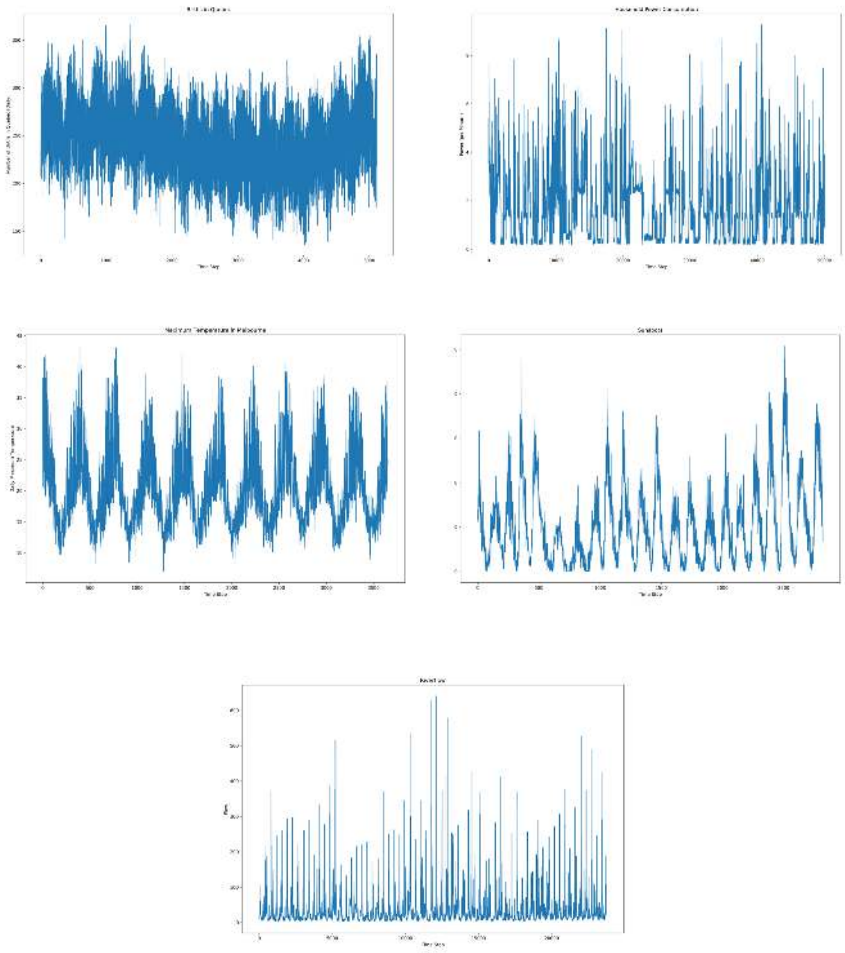

Fig. 2. Graphical Data Overview

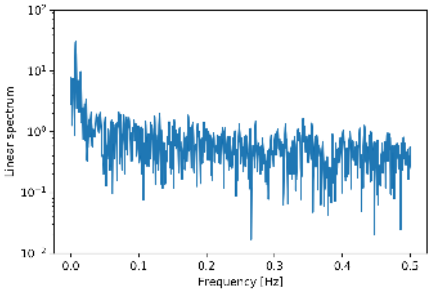

Fig. 3. Power Spectrum of the Sunspots Data Set

4.1 Model Evaluation

We validate the performance of our approach on five different data sets listed in Table 1. Fig. 2 illustrates the series on their original scale. Evidently, each of the datasets has its very own characteristics and dynamics. While the daily birth rates data set shows signs of weak stationarity, the sensor-generated household power dataset depicts more chaotic behavior with random noises. The latter is sampled by the minute. River flow, a monthly sampled time series, is clearly non stationary as well. The series of daily maximum temperatures repeats similar patterns over time as does the births data and shows clear signs of weak stationarity. Somewhere in between those cases fits the monthly sunspots data which shows seasonalities of varying strength and amplitude. Fig. 4 shows the root

Table 1. Datasets of the Experimental Analysis

Data	Number of Observations
Births in Quebec [9]	5,113
Household Power Consumption [16]	50,000
Maximum Temperature in Melbourne[1]	3,650
Number of Sunspots[1]	2,820
Riverflow[1]	23,741

mean square error (RMSE) on the holdout set of each dataset and method. Besides the performance of the stacked ensembles ('Snap Stack': stacked snapshot ensemble, 'ClassEns Stack': stacked ensemble of independently trained LSTMs), metrics for mean ensemble forecasts ('Snap Mean', 'ClassEns Mean') and single model forecasts ('Single opt.') are shown. The key outcomes of the analysis are:

- Snapshot ensembles with Ridge Regression as a meta learner outperform conservative ensembles as well as the single, optimized model in all cases.The traditional ARIMA models show inferior forecasting accuracy.
- On average, the stacked snapshot ensemble performs 4.2 % better than the next best baseline.
- The greatest performance gain obtained by the stacked ensemble is realized for the sunspots data. Here, the stacked snapshot ensembles outperform the next best method by 13.8%, while the performance win for the other four datasets is in a significantly lower range between 1.0% and 3.4%. Looking at the illustrated data in Fig. 2, this is an indication that our approach is particularly suitable for time series with seasonalities of varying intensity. Peaks of different amplitudes are handled well by the stacked snapshot ensemble, which a single model fails to do with a high degree of precision.
- Extending snapshot ensembles by the introduction of a meta learner leads to a great boost in performance compared to the simple mean combiner.

[1] https://datamarket.com/data/list/?q=, accessed June 1, 2018

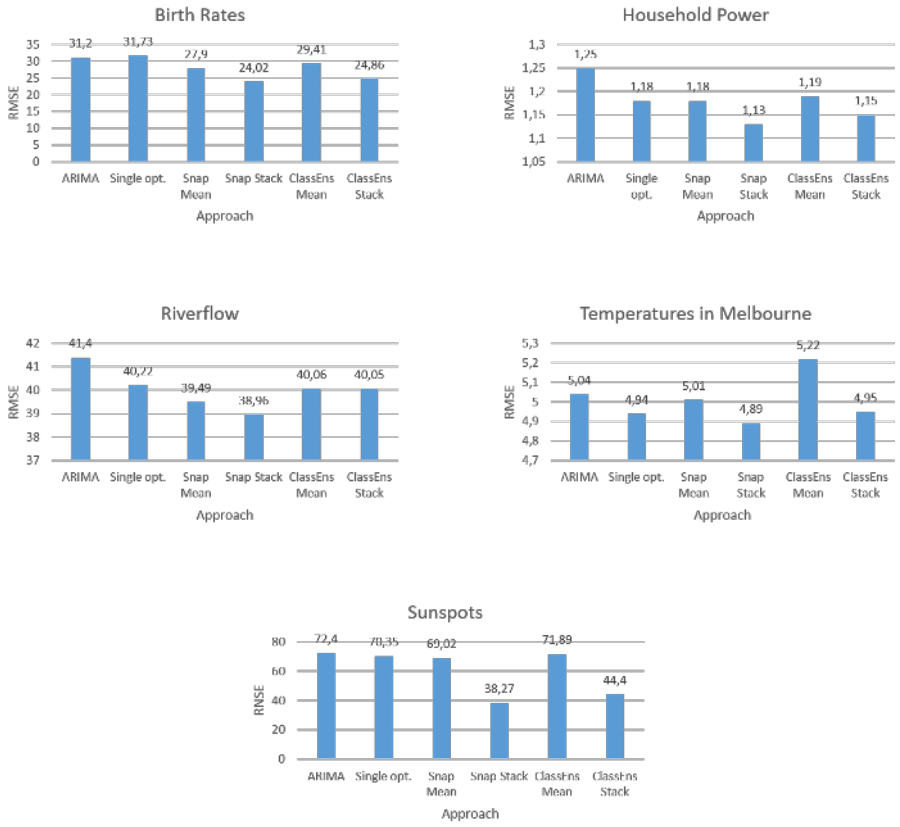

Fig. 4. Model Performance

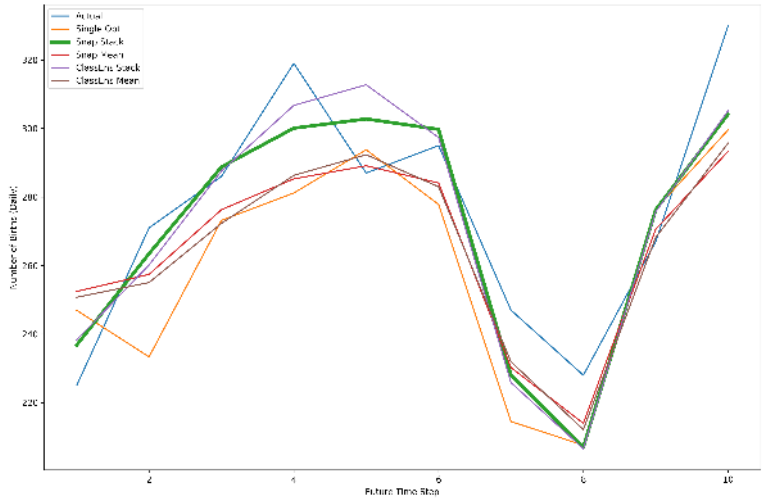

Fig. 5. Exemplary Forecast

- The ensemble forecasts are significantly different from the estimates of the remaining models, based on the paired t-test for significance.
- The single optimized LSTM only shows comparative performance if the structure of the dataset is approximately stationary over time, as in the case of the maximum temperatures series. This supports our hypothesis that snapshot ensembles are particularly suitable for cases where patterns are spread across multiple sequence lengths.
- Reslicing the input data according to the FFT after each snapshot leads to base learners with high diversity. This enables the meta learner to exploit different knowledge that is encoded across the snapshots. As an example, the ordered FFT sequence lengths for the birth rates dataset are as follows: 365, 183, 73, 61, 37, 91, 41, 30, 10, 52, 11, 26, 852, 28, 14, 568, 341, 16, 20, 465. This clearly shows how FFT extracts potent periodicities from the time series as the yearly and monthly seasonalities are immediately detected.

An exemplary 10 step ahead forecast is shown in Fig. 5. Here, the first holdout sequence of the birth rates series along with its model estimates is illustrated. One can see the significant improvements that are attributed to the meta learner, leading to the reduction in forecasting error.

The code for the experiments is available on GitHub[2] .

[2] https://github.com/saschakrs/TS-SnapshotEnsemble, accessed June 1, 2018

5 Future Work and Conclusion

Snapshot ensembles based on FFT sequence lengths are an efficient method to extract diverse patterns from data. We have shown that they yield superior forecasting performance in comparison to the standard optimization of a single LSTM and an ensemble of fully independently trained LSTMs, without the need for additional computational costs. It turned out that these results are stable across different data sets, although the relative performance boost differs depending on the underlying data structure. Our approach enables the automated generation of robust time series forecasts without the assumption of a specified data distribution. This makes the framework a valuable application for systems that require the future estimation of one or more key performance indicators that develop over time.

There is further potential regarding the design of the ensemble architecture: Besides the configuration of the individual base learners, different combiner functions might improve the overall performance for certain problems. In addition to this, we found that five epochs per snapshot lead to good overall performance of the ensemble, however, this parameter could be higher for very complex learning tasks.

It is also possible to extend the ensemble by different model types. Integrating autoregressive models or state-space representations could increase model diversity and thereby lead to a greater performance win by the combiner function.

Finally, LSTM snapshot ensembles are currently limited to univariate time series. Evaluating their applicability to the multivariate case is another challenge worth investigating. It would also be interesting to evaluate the applicability of stacked snapshot ensembles to different sequence learning tasks such as machine translation.

References

1. Ratnadip Adhikari. 2015. A neural network based linear ensemble framework for time series forecasting. Neurocomputing 157 (2015), 231-242.
2. Ratnadip Adhikari and RK Agrawal. 2014. A linear hybrid methodology for improving accuracy of time series forecasting. Neural Computing and Applications 25, 2 (2014), 269-281.
3. Cagdas Hakan Aladag, Erol Egrioglu, and Cem Kadilar. 2009. Forecasting nonlinear time series with a hybrid methodology. Applied Mathematics Letters 22, 9 (2009), 1467-1470.
4. Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks 5, 2 (1994), 157-166.
5. Mohamed G Elfeky, Walid G Aref, and Ahmed K Elmagarmid. 2005. Periodicity detection in time series databases. IEEE Transactions on Knowledge and Data Engineering 17, 7 (2005), 875-887.

6. Felix A Gers, Douglas Eck, and Jrgen Schmidhuber. 2002. Applying LSTM to time series predictable through time-window approaches. In Neural Nets WIRN Vietri-01. Springer, 193-200.
7. James Douglas Hamilton. 1994. Time series analysis. Vol. 2. Princeton university press Princeton.
8. Zhen He, Shaobing Gao, Liang Xiao, Daxue Liu, Hangen He, and David Barber. 2017. Wider and Deeper, Cheaper and Faster: Tensorized LSTMs for Sequence Learning. In Advances in Neural Information Processing Systems. 1-11.
9. Keith W Hipel and A Ian McLeod. 1994. Time series modelling of water resources and environmental systems. Vol. 45. Elsevier.
10. Sepp Hochreiter and Jrgen Schmidhuber. 1997. Long short-term memory. Neural computation 9, 8 (1997), 1735-1780.
11. Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. Neural networks 2, 5 (1989), 359-366.
12. Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In Advances in neural information processing systems. 2042-2050.
13. Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Li, John Hopcroft, and Kilian Weinberger. [n. d.]. Snapshot Ensembles: Train 1 Get M for Free. In Proceedings of the International Conference on Learning Representations (ICLR 2017).
14. Sascha Krstanovic and Heiko Paulheim. 2017. Ensembles of Recurrent Neural Networks for Robust Time Series Forecasting. In International Conference on Innovative Techniques and Applications of Artificial Intelligence. Springer, 34-46.
15. Martin Lngkvist, Lars Karlsson, and Amy Loutfi. 2014. A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recognition Letters 42 (2014), 11-24.
16. M. Lichman. 2013. UCI Machine Learning Repository. (2013). http://archive.ics.uci.edu/ml
17. Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. 2015. Long short term memory networks for anomaly detection in time series. In Proceedings. Presses universitaires de Louvain, 89.
18. Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In International Conference on Machine Learning. 1310-1318.
19. Deepak Sharma, Biju Issac, GPS Raghava, and R Ramaswamy. 2004. Spectral Repeat Finder (SRF): identification of repetitive sequences using Fourier transformation. Bioinformatics 20, 9 (2004), 1405-1412.
20. R.H. Shumway and D.S. Stoffer. 2010. Time Series Analysis and Its Applications: With R Examples. Springer New York.
21. Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Advances in neural information processing systems. 3104-3112.
22. LiWang, Haofei Zou, Jia Su, Ling Li, and Sohail Chaudhry. 2013. An ARIMA-ANN hybrid model for time series forecasting. Systems Research and Behavioral Science 30, 3 (2013), 244-259.
23. Peter Welch. 1967. The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. IEEE Transactions on audio and electroacoustics 15, 2 (1967), 70-73.
24. G Peter Zhang. 2003. Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50 (2003), 159-175.

25. Le Zhang and Ponnuthurai Nagaratnam Suganthan. 2017. Benchmarking Ensemble Classifiers with Novel Co-Trained Kernal Ridge Regression and Random Vector Functional Link Ensembles [Research Frontier]. IEEE Computational Intelligence Magazine 12, 4 (2017), 61-72.
26. Mariana Oliveira and Luis Torgo. 2014. Ensembles for time series forecasting. JMLR: Workshop and Conference Proceedings 39:360-370
27. Vitor Cerqueira, et al. 2017. Arbitrated Ensemble for Time Series Forecasting. Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Cham.
28. Harry Pratt, et al. 2017. FCNN: Fourier Convolutional Neural Networks. Machine Learning and Knowledge Discovery in Databases. Springer, Cham.
29. Himanshu Gothwal, Silky Kedawat, and Rajesh Kumar. 2011. Cardiac arrhythmias detection in an ECG beat signal using fast fourier transform and artificial neural network. Journal of Biomedical Science and Engineering 4.04 (2011): 289.
30. Anastasia Borovykh, Sander Bohte, and Cornelis W. Oosterlee. 2018. Conditional time series forecasting with convolutional neural networks. Journal of Computational Finance.
31. Aäron Van Den Oord, et al. 2016. WaveNet: A generative model for raw audio. SSW.
32. Ruofeng Wen, Kari Torkkola, and Balakrishnan Narayanaswamy. 2017. A Multi-Horizon Quantile Recurrent Forecaster.NIPS 2017 Time Series Workshop.