Telescope: An Automatic Feature Extraction and Transformation Approach for Time Series Forecasting on a Level-Playing Field

André Bauer*, Marwin Züfle*, Nikolas Herbst*, Samuel Kounev*, and Valentin Curtef†
*University of Würzburg, Germany – Email: {firstname}.{lastname}@uni-wuerzburg.de
†Cosmo Consult Data Science GmbH, Würzburg, Germany – Email: valentin.curtef@cosmoconsult.com

***Abstract*—One central problem of machine learning is the inherent limitation to predict only what has been learned — stationarity. Any time series property that eludes stationarity poses a challenge for the proper model building. Furthermore, existing forecasting methods lack reliable forecast accuracy and time-to-result if not applied in their sweet spot. In this paper, we propose a fully automated machine learning-based forecasting approach. Our Telescope approach extracts and transforms features from an input time series and uses them to generate an optimized forecast model. In a broad competition including the latest hybrid forecasters, established statistical, and machine learning-based methods, our Telescope approach shows the best forecast accuracy coupled with a lower and reliable time-to-result.**

***Index Terms*—Automatic feature extraction, Combining forecasts, Comparative studies, Forecasting competitions, Long term time series forecasting, Time series**

I. Introduction

As time series forecasting is an essential pillar in many decision-making fields [1], automating the choice and configuration of the most suitable method is a crucial challenge. In the last years, different types of hybrid forecasting methods have been presented to attack the "No-Free-Lunch Theorem" [2], which was initially formulated for optimization problems, but it also appears to be valid in the forecasting context. While statistical models have their difficulties with complex patterns, machine learning-based regressors struggle with non-stationary data [3] to extrapolate for a forecast. From our experience, recently presented hybrid methods are compute-intense, difficult to automatically execute while susceptible to tailoring to a given challenge data set. However, many real-life scenarios where forecasting is needed have stringent requirements on the speed of the forecasting mechanism and the reliability (i.e., accuracy) of the provided forecasts. Also, the end-to-end process of forecast execution from feature and method selection, to data preprocessing, model building and prediction, needs to be fully automated. Consequently, we pose ourselves the following research question: RQ1: *How to build a generic forecasting approach that delivers accurate forecasts while having a low time-to-result variance?*

To achieve a low variance in forecast accuracy, the pre-processing of historical data, and the feature handling (intrinsic extraction, engineering, and selection) must be done in a sophisticated way. On the one hand, the selection of the essential features is a decisive part. On the other hand, transforming historical data may lead to simpler patterns that usually allows more accurate forecasts [1]. Thus, the subsequent research question arises: RQ2: *How to automatically extract and transform features of the considered time-series to increase the forecast accuracy?*

Addressing the questions above, our contribution in this paper is two-fold: (i) We introduce a machine learning-based forecasting method, called Telescope[1], that automatically retrieves relevant information from a given time series. Based on this information, our method extracts and transforms intrinsic features from the input time series and then uses them to generate a forecast model. We integrate different methods to handle non-stationary data introduced by trends and multiplicative effects (Section II). (ii) In the evaluation, we compare our approach to a set of 9 state-of-the-art forecasting methods covering recent hybrid, as well as established machine learning, and statistical approaches. The results show that Telescope achieves the lowest average forecast error while keeping the time-to-result low and reliable (Section III).

II. Telescope Forecasting Workflow

The assumption of data stationarity is an inherent limitation for time series forecasting. Any time series property that eludes stationarity, such as non-constant mean (trend), seasonality, non-constant variance, or multiplicative effect, poses a challenge for the proper model building [4]. Consequently, our approach called Telescope automatically transforms the time series, derives intrinsic features from the time series, selects a suitable set of features, and handles each feature separately.

In general, many systems are governed by human interactions. That is, time series produced or observed by these systems are subjected to human habits and are thus seasonal. Therefore, Telescope is intended to handle seasonal time series. In the unlikely case where no seasonality exists within a time series, the forecasting method has a fallback that is described at the end of this section. Figure 1 shows a high-level view of the forecasting workflow for seasonal time series. Telescope consists of three fundamental phases: (i) preprocessing the time series, (ii) building a model that describes the time series, and (iii) forecasting the future behavior of

[1] Telescope at GitHub: https://github.com/DescartesResearch/telescope

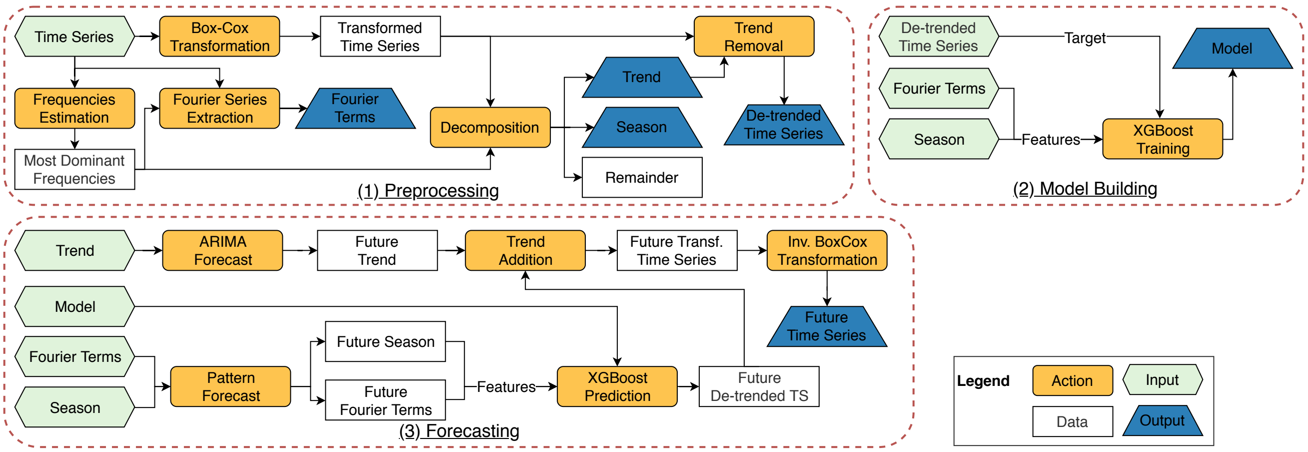

Fig. 1. Telescope workflow for seasonal time series.

the time series. In Figure 1, orange, rounded boxes represent actions, white boxes data, green hexagons the input of a phase, and blue trapezoids the output of a phase.

Preprocessing: Telescope gets as input the *Time Series* and starts with the *Frequencies Estimation*. That is, a periodogram [5] is applied on the input time series to retrieve all frequencies (i.e., the lengths of the periods) within this time series. Then, Telescope iterates over the found frequencies and matches each frequency with reasonable frequencies (e.g., daily, hourly, and yearly). If a frequency matches a reasonable frequency with a tolerance, this frequency is considered. We assume that reasonable frequencies match multiples of natural time units. In accordance to *RQ2*, Telescope selects only the most dominant frequencies. To this end, the *Most Dominant Frequencies* are derived by putting the likely reasonable frequencies to a set. More precisely, the threshold $\geq 50\%$ of the spectral value from the most dominant frequency.

As time series may have multiple seasonal patterns (such as daily and weekly) [1], Fourier terms for each dominant frequency are extracted from the input time series for modelling the different patterns. These *Fourier Terms* are used as input for the *Model Building* and *Forecasting*.

Forecasting methods, especially machine learning methods, struggle with changing variance and multiplicity within a time series [3]. To this end, Telescope performs a Box-Cox transformation [6] to adjust the the input time series. We integrated this step as it reduces both variance and multiplicative effects of the time series that leads to an improved forecast model [1], [4]. As the Box-Cox transformation depends on the transformation parameter, this parameter is estimated by the method proposed by Guerrero [7] and restricted to values ≥ 0.

Although most forecasting methods assume stationary time series (i.e., the mean and variance of a time series do not change over time) [8], many time series exhibit trend or/and seasonal patterns. That is, in practice, time series are usually non-stationary [9]. To tackle the non-stationarity, STL (Seasonal and Trend decomposition using Loess) [10] is used to split the *Transformed Time Series* with the most dominant frequency into the components: *Trend*, *Season*, and *Remainder*. Moreover, the *Trend* is used as input for the *Forecasting* phase and the *Season* is used for both the *Model Building* and *Forecasting* phase. Afterwards, the trend is removed to make the transformed time series trend-stationary. The resulting *De-trended Time Series* is used as as input for the *Model Building*.

Model Building: This phase gets as input from the first phase the retrieved intrinsic features *Fourier Terms* and *Season* of the time series as well as the *De-trended Time Series*. To build a suitable forecast model that takes the available features into account, we use machine learning. More precisely, we apply XGBoost (eXtreme Gradient Boosting) [11] as machine learning-based regression method. Consequently, Telescope models how the *De-trended Time Series* can be described by *Fourier Terms* and *Season*. The resulting *Model* is used as input for the *Forecasting* phase.

Note that we exclude the *Remainder* and the *Trend* as features to reduce the model error and later the forecast error. On the one hand, the remainder of the time series is not explicitly considered a feature. That is, the machine learning method learns the remainder as the difference that is missing to recreate the target value fully. On the other hand, as a strong trend both increases the variance and violates stationarity, the trend was removed during the first step to make the time series trend-stationary. Moreover, we choose XGBoost as machine learning method as boosting tree algorithms are time-efficient, accurate, and easy to interpret [12] and XGBoost outperforms other techniques from this field [11]. We excluded other methods like *Support Vector Machine*, *Random Forest*, or neuronal networks due to their unfeasible run-time [4].

Forecasting: This last phase gets as input the *Trend*, *Season*, and the *Fourier Terms* from the *Preprocessing* phase as well as the *Model* from *Model Building* phase. To forecast the *Time Series*, each feature and the *Trend* have to be forecast separately. As the *Season* and the *Fourier Terms* are recurring patterns per definition, they can be merely continued. The resulting *Future Season* and *Future Fourier Terms* are used in conjunction with the *Model* during the *XGBoost Prediction* to build the future de-trended time series.

To predict and assemble the future adjusted time series, the *Future Trend* has to be added to the *Future De-trended TS*. Since the *Trend* contains no recurring patterns, an advanced forecasting method is required to forecast the *Future Trend*. To this end, we apply ARIMA (autoregressive integrated moving average) [13] as it is able to estimate the trend even from a few points. More precisely, a non-seasonal ARIMA model is used to forecast the *Future Trend*.

After the forecast of the *Future De-trended TS* and the *Future Trend*, Telescope assembles both parts to the *Future Transf. Time Series*. As the time series was adjusted with the Box-Cox transformation, the *Future Transf. Time Series* has to be re-transformed. To this end, the inverse Box-Cox transformation with the identical transformation parameter from the *Preprocessing* phase is applied to the *Future Transf. Time Series*. Finally, the forecast of the original time series is returned.

Fallback for Non-Seasonal Time Series: In the case that Telescope has to forecast a non-seasonal time series, the normal workflow cannot be used. The core idea of Telescope is to detect recurring patterns within a time series and use this information to retrieve features. That is, a non-seasonal time series lacks recurring patterns. Further, STL also requires a frequency to decompose the time series into the components: trend, season, and remainder. Consequently, Telescope requires another strategy for non-seasonal time series: first, Telescope adjusts the time series with the Box-Cox transformation as explained in the preprocessing phase. Then, an ARIMA model without seasonality is determined to forecast the adjusted time series. Finally, the forecast is re-transformed with the inverse Box-Cox transformation.

III. Forecasting Method Competition

To evaluate the performance of our approach, we design a broad forecaster competition based on 400 diverse and recorded time series and compare the results against 9 existing forecasting methods. Each forecasting method gets a single time series from the data set as input. That is a major difference to the M4[2] competition where the complete training data set (i.e., all time series) is handed to the algorithms. Before passing the time series to the methods, each time series is split into history consisting of the first 80% values. Then, each method uses the history to learn a model. Afterward, each method forecasts the remaining 20% of the time series at once with a single execution. That is, each method performs multi-step-ahead forecast. The split of the time series allows calculating the symmetric mean absolute percentage error (*sMAPE*) [14] between the forecast and the original last 20% of the time series. Moreover, the comparison of the forecasting methods also takes the time-to-result into account. To evaluate the repeatability of the forecasting results and to quantify the variance in the time-to-result measurement, the whole forecasting procedure (i.e., receiving the time series, estimating the parameters, building the model, and forecasting the time series) is repeated ten times for each time series.

The experiments were deployed in our private cluster that manages 8 identical hosts (HP DL160 Gen9 with 8 physical cores @2.4 GHz (Intel E5-2630v3) and 2×16 GB RAM (DIMM DDR4 RAM operated @1866 MHz)). More precisely, the forecasts were conducted on 4 virtual machines (Ubuntu 18.04.3, 2 vcores, and 4 GB RAM) with R (V 3.4.4), C++ (V 11), or Python (V 3.6.7).

Methods in Forecasting Competition: For a fair and representative evaluation, we compare Telescope against different methods from different fields classified in three categories: (i) hybrid forecasting methods – *ES-RNN* [15] (developed by Uber and winner of the M4 competition in 2018), *Hybrid* [16], and *Prophet* [17] (developed by Facebook); (ii) machine learning methods – *ANN* [18] (artificial neuronal network), *Random Forest* [19], and *XGBoost* [11]; (iii) established statistical methods – *ETS* [20], *sARIMA* [1] (seasonal ARIMA), and *tBATS* [21]. Note that we use all methods out-of-the-box. That is, there was no parameter tuning (recall "No-Free-Lunch Theorem" [2]) and the methods were used with their default settings. This also applies to the methods deployed in Telescope.

Data Set: To have a sound and broad evaluation of forecasting methods, a highly heterogeneous data set that covers different aspects is required. Indeed, there are numerous data sets available online. However, the M4 competition, for instance, contains 100,000 time series, these time series have low frequencies (1, 4, 12, and 24) and short forecasting horizons (6 to 48 data points). Further, the median length of a time series is 106 that is, for instance, too short for machine learning methods to achieve comparable results [22]. Consequently, we assemble a data set containing 400 publicly available time series that are divided into different domains: gas, electricity, unemployment, calls, requests, stocks, sales prices, exchange rate, birth rate, solar hours, temperature, etc. The time series are publicly available and originate from 50 different sources, including also time series from M4. Further, our data set covers different frequencies (1 to 3600) and lengths (20 to 372,864).

Benchmarking Telescope: Table I shows the average forecast errors based on *sMAPE* $\overline{e}$, the average time-to-result $\overline{t}$, and the standard deviation of each measure for all 10 forecasting methods in competition averaged over all time series and 10 repetitions. Note that the time-to-result for a time series reflects the duration in which the forecast method receives the time series, estimates the parameters, creates the model, and performs the forecast. In the table, the best values (the lower, the better) are highlighted in bold. The results can be summarized as follows: (i) *Telescope* achieves the best forecast accuracy (i.e., lowest forecast error on average based on sMAPE). Furthermore, *Telescope* also exhibits the lowest forecast error variability (i.e., standard deviation). Besides the accuracy, *Telescope* is on average up to 4,500 times faster than the three other most accurate methods. (ii) *sARIMA* has the second-lowest average forecast error. However, sARIMA

[2] M4 competition: https://www.mcompetitions.unic.ac.cy/the-dataset/

TABLE I
FORECAST ERROR AND TIME-TO-RESULT COMPARISON ON ALL TIME SERIES.

	ANN	ES-RNN	ETS	Hybrid	Prophet	Random Forest	sARIMA	tBATS	XGBoost	Telescope
$\overline{e}$ [%]	352.05	29.39	28.95	27.89	52.30	36.72	20.63	23.62	23.85	**19.95**
σ_e [%]	20685.90	40.93	64.57	89.76	98.78	309.97	35.63	76.00	34.67	**31.35**
$\overline{t}$ [s]	2.45	72.52	1.34	3452.53	17.59	6.43	2603.32	27.66	**0.02**	0.59
σ_t [s]	10.21	249.10	7.53	23087.56	215.07	25.62	25644.41	92.92	**0.05**	4.29

has the second slowest time-to-result and the worst time-to-result variation. (iii) *ES-RNN* and *Prophet* are tailored either to the M4 competition or to the Facebook traces and thus exhibit mediocre average forecast error (as well as variation). (iv) The machine learning methods show worse forecast errors than statistical methods.

IV. Related Work

Hybrid methods can be categorized into three groups of approaches each sharing the same basic concept. The first group *Ensemble Forecasting* computes the forecast as a weighted sum of the values derived from applying multiple methods [23], [24]. To increase the forecast accuracy, the second group *Forecast Recommendation* builds a rule set for estimating the assumed best forecasting method based on analyzing specific features of the considered time series [25], [26]. In the last group *Time Series Decomposition*, a time series is decomposed into components, and forecasting methods are applied to each component separately [17], [27].

V. Conclusion

This paper introduces Telescope, a forecasting method that automatically retrieves relevant information from a given time series, with a high forecast accuracy and a low time-to-result variance. More precisely, this approach extracts intrinsic features based on STL decomposition and Fourier terms. Further, Telescope removes the trend component, which is forecast separately, from the time series to handle non-stationarity time series. Then, XGBoost is used for composing the forecast components. In an extensive competition with 9 state-of-the-art forecasting methods on 400 real-world time series, Telescope achieves the lowest average forecast error while keeping the time-to-result low and reliable.

References

[1] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. Melbourne, Australia: OTexts, 2018.

[2] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, Apr 1997.

[3] M. Sugiyama and M. Kawanabe, *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation*. The MIT Press, 2012.

[4] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PloS one*, vol. 13, no. 3, p. e0194889, 2018.

[5] A. Schuster, "The periodgram of magnetic declination as obtained from the records of the greenwich observatory during the years 1871-1895," *Trans. Cambridge Philosophical Soc.*, vol. 18, pp. 107–135, 1899.

[6] G. E. Box and D. R. Cox, "An analysis of transformations," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 211–252, 1964.

[7] V. M. Guerrero, "Time-series analysis supported by power transformations," *Journal of Forecasting*, vol. 12, no. 1, pp. 37–48, 1993.

[8] P. J. Brockwell and R. A. Davis, *Introduction to time series and forecasting*. springer, 2016.

[9] R. Adhikari and R. K. Agrawal, "An introductory study on time series modeling and forecasting," *CoRR*, vol. abs/1302.6613, 2013.

[10] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition procedure based on loess," *Journal of Official Statistics*, vol. 6, no. 1, pp. 3–73, 1990.

[11] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *ACM SIGKDD 2016*. ACM, 2016, pp. 785–794.

[12] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, 2017, pp. 3146–3154.

[13] G. E. Box and G. M. Jenkins, "Time series analysis. forecasting and control," in *Holden-Day Series in Time Series Analysis, Revised ed., San Francisco: Holden-Day, 1976*, 1976.

[14] R. J. Hyndman and A. B. Koehler, "Another Look at Measures of Forecast Accuracy," *International Journal of Forecasting*, pp. 679–688, 2006.

[15] S. Smyl, "M4 Forecasting Competition: Introducing a New Hybrid ES-RNN Model," https://eng.uber.com/m4-forecasting-competition/, 2018, accessed: 2019-09-16.

[16] "Hybrid Forecasting," https://cran.r-project.org/web/packages/forecastHybrid/vignettes/forecastHybrid.html, accessed: 2019-09-16.

[17] S. J. Taylor and B. Letham, "Forecasting at scale," *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018.

[18] R. J. Hyndman, G. Athanasopoulos, C. Bergmeir, G. Caceres, L. Chhay, M. O'Hara-Wild, F. Petropoulos, and S. Razbash, *forecast: Forecasting functions for time series and linear models*, 2018, r package version 8.4. [Online]. Available: http://pkg.robjhyndman.com/forecast

[19] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose, "A state space framework for automatic forecasting using exponential smoothing methods," *International Journal of forecasting*, vol. 18, no. 3, pp. 439–454, 2002.

[21] A. M. De Livera, R. J. Hyndman, and R. D. Snyder, "Forecasting time series with complex seasonal patterns using exponential smoothing," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1513–1527, 2011.

[22] V. Cerqueira, L. Torgo, and C. Soares, "Machine learning vs statistical methods for time series forecasting: Size matters," *arXiv preprint arXiv:1909.13316*, 2019.

[23] R. T. Clemen, "Combining forecasts: A review and annotated bibliography," *International Journal of Forecasting*, vol. 5, no. 4, pp. 559–583, 1989.

[24] V. Cerqueira, L. Torgo, F. Pinto, and C. Soares, "Arbitrated ensemble for time series forecasting," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2017, pp. 478–494.

[25] T. S. Talagala, R. J. Hyndman, and G. Athanasopoulos, "Meta-learning how to forecast time series," Monash University, Department of Econometrics and Business Statistics, Tech. Rep., 2018.

[26] M. Züfle, A. Bauer, V. Lesch, C. Krupitzer, N. Herbst, S. Kounev, and V. Curtef, "Autonomic Forecasting Method Selection: Examination and Ways Ahead," in *Proceedings of the 16th IEEE International Conference on Autonomic Computing (ICAC)*. IEEE, June 2019.

[27] N. Liu, Q. Tang, J. Zhang, W. Fan, and J. Liu, "A hybrid forecasting model with parameter optimization for short-term load forecasting of micro-grids," *Applied Energy*, vol. 129, pp. 336–345, 2014.