Feature-driven Time Series Generation

Lars Kegel, Martin Hahmann, Wolfgang Lehner
Technische Universität Dresden
01062 Dresden, Germany
{firstname.lastname}@tu-dresden.de

ABSTRACT

Time series data are an ubiquitous and important data source in many domains. Most companies and organizations rely on this data for critical tasks like decision-making, planning, and analytics in general. Usually, all these tasks focus on actual data representing organization and business processes. In order to assess the robustness of current systems and methods, it is also desirable to focus on time-series scenarios which represent specific time-series features. This work presents a generally applicable and easy-to-use method for the feature-driven generation of time series data. Our approach extracts descriptive features of a data set and allows the construction of a specific version by means of the modification of these features.

Categories and Subject Descriptors

I.6.7 [**Simulation and modeling**]: Simulation Support Systems; G.3 [**Probability and statistics**]: Time Series Analysis; H.2.8 [**Database management**]: Database Applications—*Statistical databases*

Keywords

time series analysis, data generation, business analytics

1. INTRODUCTION

A time series is a sequence of measurements which represents the result of a dynamic process measured at successive time instances. It is a popular and widely applied data type which arises in a multitude of application domains like, for example, in the energy domain, in market research, and in manufacturing processes. But not only are they important as descriptive information of the past. Being processed by data mining techniques, such as clustering, classification or forecasting, they reveal insights into the process behavior which makes them a valuable source for decision-making and planning.

Time series have a high dimensionality due to their length. This is why they are reduced to scalar values, so-called time series characteristics, that represent their overall behavior. For example, moments, extreme values and periodic variation are typical characteristics of a time series. Moreover, they are often decomposed to components such as trend and season before they are reduced to characteristics. These characteristics, which we call features, are for example the slope of a trend or the strength of a season and they are a promising and flexible representation for time series in many domains.

While all data mining techniques differ greatly with respect to their individual goals, they have one thing in common: they require large amounts of time series data covering a variety of possible time series characteristics. The training of data mining techniques on large, variously shaped input leads to better evaluation results and to more robust techniques.

Although there is a high interest in time series analysis, it is difficult for many stakeholders to get access to a satisfying data set. Basically, there are two sources: Stakeholders may retrieve their data from their own measured processes or from closed research projects. This data comes from the same process but it does not necessarily cover all possible characteristics. On the other hand, they have access to publicly available data sets which covers a variety of time series characteristics but where the generating processes are not necessarily compatible. Our research aims to fill this gap by focussing on time series generation. Generated time series can express both, the processes of the given data and the coverage of many different characteristics.

In the present work, we propose an approach for systematical time series generation based on features. By systematically modifying features of components we generate new time series for a given set of target features. These time series keep the nature of given processes and represent features that are not given by the data. Around this approach, we build a visual exploration of features that enables interactive usage.

The structure of the paper is as follows. Section 2 presents our approach as a workflow of analytical tasks and user interaction. Section 3 surveys related work on time series characteristics and time series generation, followed by the conclusion and future work in Section 4.

29^{th} GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), 30.05.2017 - 02.06.2017, Blankenburg/Harz, Germany.

2. OVERVIEW OF OUR APPROACH

Our approach consists of an analytical and interactive part. Figure 1 highlights the major steps as a flowchart. *Time series* are stored in a *database* whose structure is described in more detail in Subsection 2.1. After retrieving the data, time series are transformed. The goal of the *transformation* is to derive time series components and to reduce components to *features*. Thus, this step addresses two tasks: decomposition and feature extraction, which are explained in Subsection 2.2. A feature space visualizes features and allows for interaction.

Time series are generated by modifying specific features such that they satisfy a *target* value given by the user. The *generation* is presented in Subsection 2.3 and results in *generated time series*. Moreover, generated time series are again transformed for being displayed in the feature space. The step of *visualization and interaction* is explained in Subsection 2.4. A generated data set is *exported* to the database.

Throughout this paper, we explain our approach using the following running example.

EXAMPLE 1 (M3-COMPETITION). *The M3-Competition is the latest of three M-Competitions in 2000 [9]. Its goal is the systematic evaluation of forecast method accuracy on a defined data set. The data set consists of 3003 time series that are from different origins (industry, finance, demographic, macro-/microeconomic, other). The values of each time series are measured at defined time intervals (year, quarter, month, other).*

2.1 Database

Our approach is built on top of a database where time series are loaded from and where generated data sets are exported to. As a schema, we adopt the time series relation which is a unified representation for time series in databases [2]. Tuples of the relation consist of time, measurement and categorical attributes that are strictly ordered by time. Time instances are equidistant and complete.

Table 1 represents a time series relation from the M3-Competition. The *meterid* uniquely describes a given time series. *Code* is a categorical attribute, *date* is the time attribute and *consumption* the measurement attribute.

2.2 Transformation

The goal of the transformation is to derive time series components and to reduce components to features. Thus, this step addresses two tasks: decomposition and feature extraction, which are subsequently explained.

2.2.1 Time Series Components

Most existing techniques describe a time series as a combination of three components: a *trend*, a *season* and a *residual* component[11]. The trend represents the long-term change in the mean level of the series, whereas the season describes a cyclical repeated behavior. Residuals usually represent unstructured information that is generally assumed to be random. The sum of these components is called *additive model* and represents economic and energy time series [11].

Whether a series contains a trend or season component has to be checked first. In our automatic approach, this is carried out with test methods. The trend check is done by extracting the long-term mean of the time series and by testing whether this mean is a trend. We use a kernel smoothing method to extract the long-term mean and refer to the

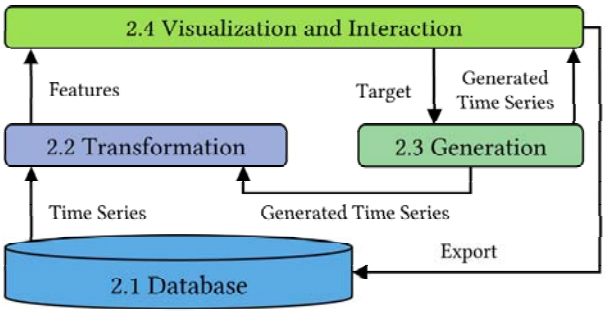


Figure 1: System Overview

bandwidth parameter $b = 2$ given by [11]. Subsequently, the trend test of Mann-Kendall [8] enables us to check whether this mean is considered as a trend.

Second, we check for a seasonal behavior on the detrended series. For this purpose, we rely on the autocorrelation analysis presented by Wang et al. [14] with one modification. The autocorrelation function of the detrended series returns autocorrelation coefficients for all lags up to 1/3 of the series length. Peaks and lows are visible and show which lag has the highest autocorrelation. The season length is the lag of the first peak of a positive autocorrelation that is preceded by a low. As we only want to assert the existence of a season, we modify this method in that we only accept the lag as season length if (1) the autocorrelation difference between peak and low is at least 0.1 [14], (2) the autocorrelation is significant in that it is within the confidence interval and positive, and (3) the lag confirms the season length given by the data.

The lag that is returned either confirms the season length given by the data or it is 1 (no season). If only a season but not a trend component exists we smooth the season with bandwidth $b = 5/L$ (where L is the season length) as given by [11]. If no season component exists but a trend, the trend is the extracted long-term mean. If both components exist we carry out a decomposition which is subsequently explained.

2.2.2 Decomposition

During decomposition, the time series is split into a trend, season and residuals component. The classical technique dates back to the 1920s and is the basis for most subsequent decomposition techniques [7, 4]. The key concept is to retrieve the trend by applying a moving-average process on given time series. Due to several shortcomings (missing robustness, endpoints of a series may not be decomposed), there are more recent techniques.

Cleveland [1] found that *Loess smoothing*, a locally-weighted regression technique, also leads to good results for detrending and deseasonalizing a time series. His method, STL, is considered as a versatile and robust decomposition technique, handling every type of season length and decomposing endpoints [4]. Since this method is widely and recently applied [12], we adopt it in our approach.

2.2.3 Feature Extraction

We aim for reducing components to scalar values, so-called features, that represent their characteristics. Three trend features are chosen: determination, slope, and linearity. Ad-

Table 1: Example Time Series Relation

Code	Meterid	Date	Consumption
SME	N1050	2009-07-14	36.809
SME	N1050	2009-07-15	34.941
Residential	N1052	2009-07-14	15.206
Residential	N1052	2009-07-15	11.256
...	...	...	...

ditionally, we choose the season determination. The trend and season determination describe the influence of the respective components compared to the residuals, the trend slope captures local trend changes, and the trend linearity the similarity of a trend to a linear behavior.

Subsequently, we explain these features with respect to the additive time series model. Let x_t be the original time series, then

$$x_t = tr_t + seas_t + res_t \tag{1}$$

where tr_t is the trend, $seas_t$ the season and res_t the residual component.

Trend Determination.

According to [14], the *trend determination* represents the influence of the trend component on the time series. The coefficient of trend determination is then

$$R^2_{tr} = 1 - \frac{var(res_t)}{var(res_t + tr_t)} \tag{2}$$

where $var(y_t) = \frac{1}{T-1}\sum_{t=1}^{T}(y_t - \bar{y})^2$ is the sample variance, y_t $(1 \leq t \leq T)$ is a sample of T values and $\bar{y}$ is the sample mean. The trend determination ranges between 0 and 1: $R^2_{tr} = 0$ means that the trend influence is negligible whereas the $R^2_{tr} = 1$ shows a high trend influence.

Trend Slope.

Assuming that there is a linear trend, we can state that a *trend slope* captures an overall increase or decrease of the time series, whereas the trend component tr_t derived by STL captures also local trend changes. In an attempt to identify the slope, we fit a linear regression model to tr_t:

$$tr_t = \theta_1 + \theta_2 \cdot l_t + \delta_t \tag{3}$$

θ_1 is the base value from which the trend starts. The slope is represented by θ_2, subsequently, it will be used as a feature. A high slope results in a high increase of the trend whereas a slope near 0 means that there is no overall increase. l_t is the vector of time instances. The difference between a trend from STL and from linear regression is expressed as δ_t, representing the difference of local trend changes in STL compared to the overall trend behavior.

Trend Linearity.

The *trend linearity* expresses the relation between the linear regression model (3) and the trend component. This feature is captured by

$$R^2_{lin} = 1 - \frac{var(\delta_t)}{var(tr_t)} \tag{4}$$

$R^2_{lin} = 1$ means that the trend is a straight line and the residuals δ_t are negligible. Otherwise, the trend fluctuates.

Season Determination.

The *season determination* represents the strength of the season component on the time series [14]. Analogous to the trend, the coefficient of season determination is

$$R^2_{seas} = 1 - \frac{var(res_t)}{var(res_t + seas_t)} \tag{5}$$

Concluding the transformation, the data set consists of time series components tagged with its respective features. It enables users to generate time series with modified features.

2.3 Generation

In our approach, a time series is generated for a given feature combination or target. Features are representatives of components and we propose to modify components by introducing factors. Subsequently, we first present the modification with factors and their calculation for a given target.

2.3.1 *Modification with Factors*

We describe four factors f, g, h and k that express the modification of a the time series component and that affect the four features trend determination, slope, linearity as well as season determination. Figure 2 gives an overview of how these factors affect the resulting time series.

Trend Determination Factor.

Let f be a factor that varies trend determination:

$$tr_t^{'} = \theta_1 + f \cdot (\theta_2 \cdot l_t + \delta_t) \tag{6}$$

This equation represents the linear regression model that is fitted to the trend. f is a factor applied to the slope θ_t and the difference δ_t. Depending on f, R^2_{tr} increases $(f > 1)$, decreases $(0 \leq f < 1)$, or is left unchanged $(f = 1)$. $f < 0$ is not admissible.

The effect of this factor is represented by Figure 2(a). The plot shows the original trend (blue with triangles) and three modified trends. The latter ones are modified by a trend determination factor $f = 1.25$, $f = 0.75$, and $f = 0.50$, respectively. Overall, the main characteristics of the trend are kept but they are a multiple of the former value. The influence on the trend determination R^2_{tr} is given in the figure's legend.

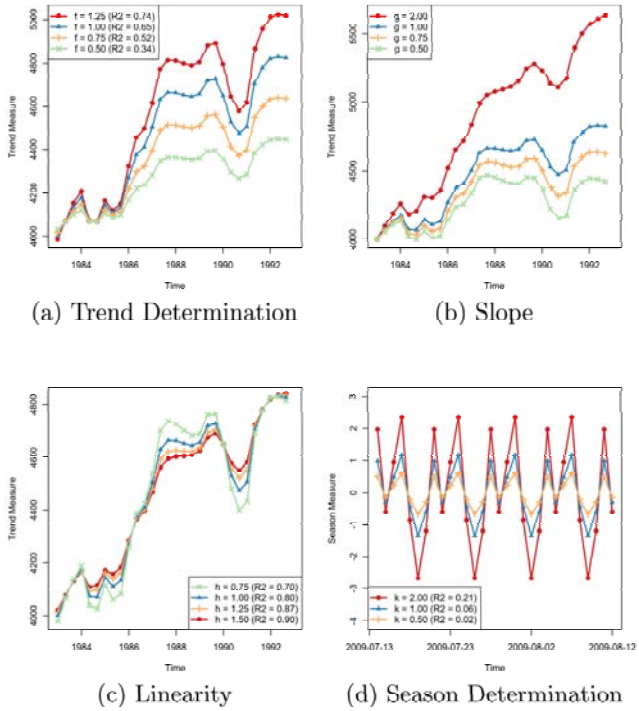
Trend Slope Factor.

Let g be a factor that varies the trend slope:

$$tr_t^{'} = \theta_1 + g \cdot \theta_2 \cdot l_t + \delta_t \tag{7}$$

Again, we apply the factor to the linear regression model. But in this case, only the slope is modified and not the difference δ_t. Depending on g, ϕ_2 increases $(g > 1)$, decreases $(0 \leq g < 1)$, or is left unchanged $(g = 0)$. $g < 0$ is not admissible.

This effect is represented in Figure 2(b). Again, the original trend (blue with triangles) and three modified trends (with factors $g = 2.00$, $g = 0.75$, $g = 0.50$) are shown. All time series start at the same base level and they keep the same trend changes but their directions are different.



(a) Trend Determination (b) Slope

(c) Linearity (d) Season Determination

Figure 2: Scenarios for Modification

Trend Linearity Factor.

We define the trend linearity as the determination of trend changes due to STL and of the linear trend. Let h be a factor that varies the trend linearity:

$$tr_t^{'} = \theta_1 + \theta_2 \cdot l_t + \frac{1}{h} \cdot \delta_t \tag{8}$$

Depending on h, R^2_{lin} increases ($h > 1$), decreases ($0 \leq h < 1$) or is left unchanged ($h = 1$). $h < 0$ is not admissible.

The effect of this factor represented in Figure 2(c). Again, the original trend (blue with triangles) and three modified trends (with factors $h = 1.50$, $h = 1.25$, $h = 0.75$) are shown. If the factor h increases the resulting trend is more linear because the difference δ_t is diminished.

Season Determination Factor.

Let there be a factor k that sets the season determination:

$$seas_t^{'} = k \cdot seas_t \tag{9}$$

Depending on k, R^2_{seas} increases ($k > 1$), decreases ($0 \leq k < 1$), or is left unchanged ($k = 1$). $k < 0$ is not admissible.

This effect is represented by Figure 2(d). The plot shows the original season (blue with triangles) and two modified season components from the Smart Metering Project. The latter ones are modified by a season determination factor $k = 2.00$ and $k = 0.50$, respectively. Modifying the season by factor k leads to higher peaks and lows. The resulting R^2_{seas} is given in the legend.

2.3.2 Feature Target Calculation

Based on time series features and factors, we are able to generate time series that systematically cover features for a given target. Generated time series keep the nature of given time series except for the modified features.

Users set a feature interval which restricts the features of the generated time series. This may be a single value or an interval of values (for more than one target). Moreover, they indicate a number of time series that has to be generated. Within the feature interval, generated time series are equidistant with respect to the modified features so that they cover the features systematically.

Calculating the factor that corresponds to the target requires the calculation of the inverse function of a feature. We exemplify this by showing how a factor f is calculated that modifies the trend determination. Let $x_t = tr_t + seas_t + res_t$ be a time series that is shifted to a feature target $R^{2'}_{tr}$ such that the modified trend is $tr_t^{'}$. Let $\bar{tr_t} = \theta_2 \cdot l_t + \delta_t$ the trend component without the offset component θ_1.

$$\begin{aligned} R^{2'}_{tr} &= 1 - \frac{var(res_t)}{var(res_t + tr_t^{'})} \\ &= 1 - \frac{var(res_t)}{var(res_t + \theta_1 + f \cdot \bar{tr_t})} \\ &= 1 - \frac{var(res_t)}{var(res_t) + f^2 \cdot var(\bar{tr_t}) + 2 \cdot f \cdot cov(res_t, \bar{tr_t}))} \end{aligned}$$

Solving the following equation returns factor f which is the positive real solution:

$$\begin{aligned} 0 = {} & f^2 \cdot var(\bar{tr_t}) \cdot (R^{2'}_{tr} - 1) \\ & + f \cdot 2 \cdot cov(res_t, \bar{tr_t}) \cdot (R^{2'}_{tr} - 1) \\ & + R^{2'}_{tr} \cdot var(res_t) \end{aligned}$$

This enables us to modify a time series such that its features is exactly the target value. For the other features this calculation is similar and omitted due to space restrictions.

2.4 Visualization and Interaction

To provide an easy way of generating time series, we propose a visual exploration and interaction approach. It allows users to explore the given data sets and its features, to select the feature interval for which time series will be generated and to display the resulting time series.

Features form dimensions by which time series are sorted. Together, these dimensions build a *feature space* which is described by Kang et al. [5]. In our work, we focus on two features at a time.

Figure 3 shows the instances of the M3-Competition in a two-dimensional scatterplot. The axes show the trend linearity and the season determination. Every dot represents a time series. We choose four time series (red triangles) which exhibit different features and which are also plotted as a line plot. Series N1078 clearly shows a strong season (compared to its residuals) and a very linear behavior. Therefore, it is in the upper right corner of the scatterplot. Series N1085 is less linear due to a trend which is not constantly increasing. While series N0754 is still very linear, it does not show a strong season. Finally, the series N2374 does not exhibit any of these two features. Thus, the users get an insight of (1) how the time series are spread across the feature space and (2) which are the time series that reside in a certain feature interval.

We further the idea of Kang et al. in that users interact with the feature space. Users indicate the feature interval

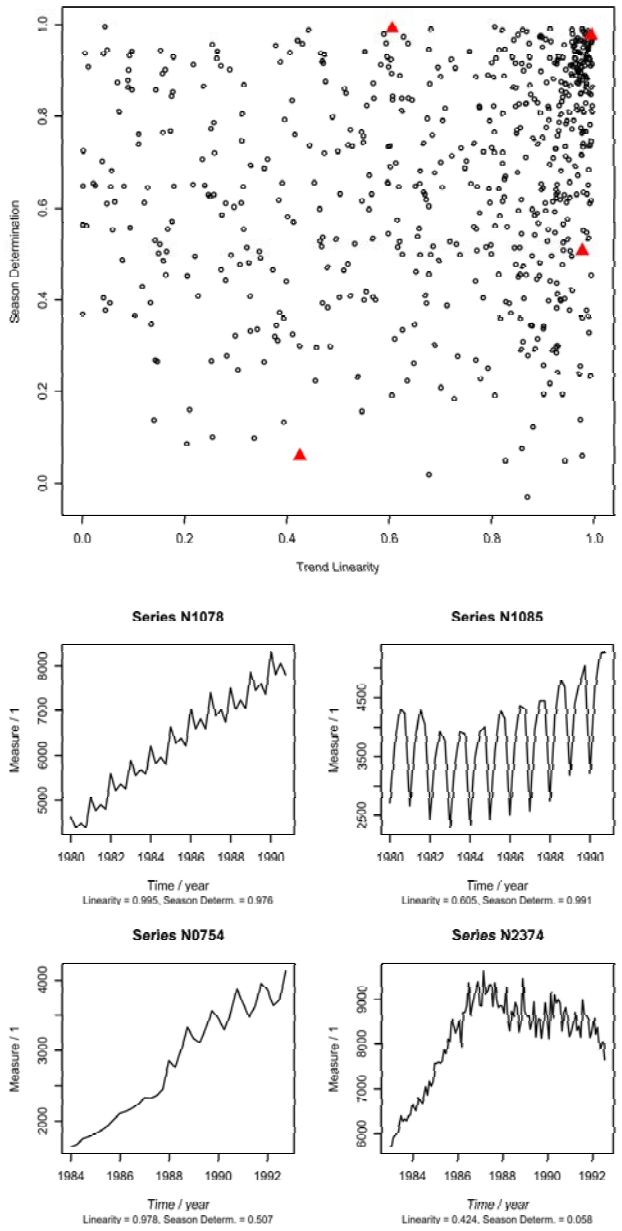


Figure 3: Trend Linearity and Season Determination: Scatterplot and Selected Time Series

either by clicking (indicating a target) or by brushing (indicating a feature interval as a target). Moreover, they indicate a number of time series to be generated. A sample of time series is selected for the given number. Each time series is modified so that their features are moved to the target. In case of a feature interval, the features of generated time series are equally distributed such that they form a grid.

The time series generation tool is implemented as an R package [10]. Figure 4 gives an overview of the visualization showing time series of the M3-Competition. The *feature space* shows a scatter plot of the features, the *time series summary* shows a line plot with original time series (black) and the generated time series (red). Below, users may set features that are displayed (*select axis*) and select the *number of time series to generate*. In this example, time series instances are generated (red triangles) whose trend determination is between 0.50 and 1.00 and whose season determination is between 0.00 and 0.50 (blue rectangle). From all available time series (black dots), a sample is selected and shifted towards the target.

3. RELATED WORK

Our approach makes use of time series features for modification and generation. Relevant to this work are data mining techniques, such as clustering, classification and forecasting, that make use of global time series characteristics. We will present these approaches (Subsection 3.1), followed by a review on time series generation (Subsection 3.2).

3.1 Time Series Characteristics

Wang et al. present a time series clustering technique based on time series characteristics [14]. They select characteristics as well as features that are related to time series components. Due to the the dimensionality reduction, their clustering is very fast and it can easily support gapped series. Among their features, they define a trend and a season determination. We adopt these features in our work and introduce a modification for time series generation.

Fulcher and Jones [3] present an application of time series characteristics in classification. They describe time series with thousands of characteristics that arise from many different application domains. Subsequently, an automatical feature selection recommends the most competitive features for the classification.

In forecasting, time series characteristics are a promising representation of the data as they can recommend a suitable forecasting method. Kang et al. [5] present an approach that selects among six forecasting methods and recommends a forecasting method with regard to the time series' characteristics. Although the authors cannot show that a forecasting methods performs best for a given set of characteristics, they can recommend which forecasting method should be avoided. With our generation approach, we enable forecasters to systematically check for which features this recommendation is possible and how this can be used for forecasting recommendation systems [13].

3.2 Time Series Generation

The generation of time series presented by Kang et al. [5] is related to our approach in that it aims at generating data sets for assessing the robustness of data mining methods. The authors reduce time series to global characteristics which are further combined by principal component analysis and visualized in a feature space. In terms of generation, they rely on a genetic algorithm that generates new time series by selection, crossover, and mutation of given time series.

In our work, we limit ourselves to the visualization of features of time series components that are not further combined to prinicipal components. In this case, the feature space gives users better insights and enables them to explicitly generate time series for a given feature such as, for example, a time series with a target trend determination. The generation in our work relies on the time series modification with target features. This makes the generation process reproducible and assumes that no other features are affected by this generation. While Kang et al. focus on generating

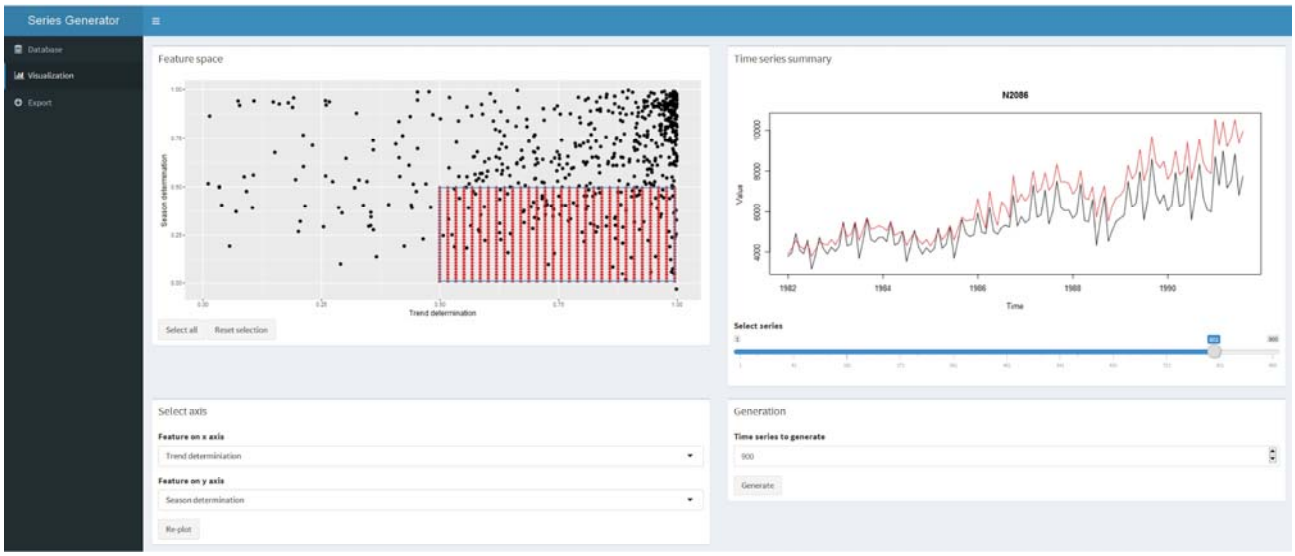


Figure 4: Screenshot of Time Series Generation Tool

time series for assessing the robustness of forecasting methods we also aim to provide data for other analytic purposes and robustness checks.

Recently, we presented Loom, which is a framework for time series generation [6]. A generated data set is either synthetic or derived from a given data set. In the first case, it represents a realization of a mathematical model which defines the time series process. In the second case, the generation from a given data set consists of sampling, recombining, or simulating given time series and their components. Although this system offers different approaches for time series generation, it does cover systematically time series features.

4. CONCLUSION AND FUTURE WORK

Data generation enables users to cover the ubiquity of possible input configurations of a system or a method in order to assess their robustness. Applied on time series, it gives better insights into recorded data and it results in more complete and various scenarios. Features are a generally applicable reduction of time series components that are easily visualized and explored by the feature space. With our application Loom [6], we are now able to generate data sets that systematically cover time series for different targets.

Future work will mainly cover the extension of the feature set in order to provide a variety of time series characteristics and their respective modification rule. By automatically selecting features that are highly discriminatory, we enable users to focus on the most important features of a given data set. Regarding the feature space, we will study the combination of two concurrent features such as, for example, two trend modifications. They may set further constraints for generating time series for a given target.

5. ACKNOWLEDGMENTS

This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731232.

6. REFERENCES

[1] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning. STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *J. Off. Stat.*, 6:3–73, 1990.

[2] U. Fischer. *Forecasting in database systems*. PhD thesis, TU Dresden, Germany, 2014.

[3] B. D. Fulcher and N. S. Jones. Highly comparative feature-based time-series classification. *IEEE Trans. Knowl. Data Eng.*, 26(12):3026–3037, 2014.

[4] R. J. Hyndman and G. Athansopoulos. *Forecasting: principles and practice*. OTexts, 2013.

[5] Y. Kang, R. J. Hyndman, and K. Smith-Miles. Visualising forecasting algorithm performance using time series instance spaces. *Int J Forecast*, 33(2):345–358, 2017.

[6] L. Kegel, M. Hahmann, and W. Lehner. Template-based time series generation with loom. In *Workshops Proc. of EDBT/ICDT*, 2016.

[7] M. Kendall and A. Stuart. *The Advanced Theory of Statistics*, volume 3, pages 410 – 414. Griffin, 1983.

[8] M. G. Kendall. *Rank correlation methods*. 1948.

[9] S. Makridakis and M. Hibon. The M3-Competition: results, conclusions and implications. *Int. J. Forecast.*, 16(4):451 – 476, 2000.

[10] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

[11] R. H. Shumway and D. S. Stoffer. *Time Series Analysis and Its Applications*. Springer, 2011.

[12] M. Theodosiou. Forecasting monthly and quarterly time series using STL decomposition. *Int. J. Forecast.*, 27(4):1178 – 1195, 2011.

[13] R. Ulbricht, C. Hartmann, M. Hahmann, H. Donker, and W. Lehner. Web-based Benchmarks for Forecasting Systems - The ECAST Platform. In *SIGMOD*, pages 2169–2172, 2016.

[14] X. Wang, K. A. Smith, and R. J. Hyndman. Characteristic-based clustering for time series data. *Data Min. Knowl. Discov.*, 13(3):335–364, 2006.