Forecasting Multivariate Time Series with a Dynamic-System-Based Hybrid Model

Daria Ganaeva and Anna Golovkina(✉)

Saint Petersburg State University, Saint Petersburg, Russia
a.golovkina@spbu.ru

Abstract. This paper proposes a time series forecasting approach that integrates nonlinear dynamic systems reconstruction with statistical forecasting methods. The suggested technique is as follows: the time series is decomposed into dynamic and stochastic components, each of which is used to build independent predictive models. In the end, the forecasting results are combined together. The research compares the prediction results and errors values for with the suggested technique, ARIMA, SVR, and a sparse identification method SINDy on synthetic and real-world data sets. Forecasting accuracy utilizing the suggested technique outperforms that of competing methods. Thus, the proposed approach may be utilized to solve practical issues involving the prediction of the behavior of diverse processes for which the mathematical model is unknown but data are accessible at discrete periods.

Keywords: Dynamic system reconstruction · Time series forecasting · Time series decomposition

1 Introduction

Forecasting is a method for making predictions about the direction of future trends based on historical data and current inputs. Forecasting is the most important optimization concept in terms of energy savings, material savings, boosting efficiency, and making appropriate control decisions.

The forecasting problem can be solved from two angles: building univariate or multivariate time series models. Traditional universal forecasting methods predict variables using only information contained in their own past values and possibly the current and past values of an error term. In contrast, multivariate models attempt to explain changes in a variable by reference to movements in the current or past values of other (explanatory) variables.

One of the most popular univariate methods is autoregressive integrated moving average (ARIMA) modeling, which uses past values of the variable and its differences and moving averages to make future predictions [13,17,19]. Other univariate methods include exponential smoothing and trend analysis. Univariate methods are easy to implement and interpret and can provide accurate forecasts

Supported by Saint Petersburg State University, project ID: 94062114.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023 Workshops, LNCS 14104, pp. 177–191, 2023.
https://doi.org/10.1007/978-3-031-37105-9_13

for simple time series data with a single trend or seasonality. However, they may not perform well for complex data with multiple trends or seasonalities or when external factors are influencing the data.

Multivariate methods, on the other hand, involve modeling multiple time series variables that may be related to each other. One popular multivariate method is vector autoregression (VAR) modeling, which models the relationships between multiple time series variables using lagged values of each variable [2]. Multivariate methods can provide more accurate forecasts for complex data as they can capture the effects of multiple variables and external factors.

Besides the mentioned traditional statistical approaches, in recent years, there has been an increasing interest in machine learning [20] and deep learning-based [5,8,21] methods for time series forecasting, which have shown promising results in both univariate and multivariate settings.

One popular deep learning-based method is the long short-term memory (LSTM) model [11], which is a variant of recurrent neural networks (RNNs) that can capture long-term dependencies in data. LSTMs have been shown to outperform traditional methods such as ARIMA and exponential smoothing for time series forecasting, especially for data with complex patterns and seasonality. Another neural network-based approach is the convolutional neural network (CNN) [12], which can be applied to time series data by treating it as an image and extracting relevant features using convolutional layers.

Among machine learning methods for time series forecasting, the most promising are regression models with different kernels [9] and ensemble methods that combine multiple models to improve the accuracy and robustness of forecasts [7]. One popular ensemble method is the random forest (RF) model [6], which combines multiple decision trees to make predictions. RF has been shown to perform well for time series data with multiple predictors and non-linear relationships. Another ensemble method is the gradient boosting machine (GBM) [16], which combines multiple weak models to create a strong model.

However, these methods also have some limitations. For neural network-based methods, one of their main limitations is their black-box nature, which makes it difficult to interpret the results and understand the underlying relationships between variables. Additionally, these methods may require a large amount of data and computational resources for training and may suffer from overfitting if the data is not carefully preprocessed.

For ensemble methods, the main limitation is that they may be computationally expensive to train and may require careful tuning of hyperparameters to achieve optimal performance. Additionally, these methods may not perform well for data with complex temporal dependencies or non-linear relationships.

Dynamical systems reconstruction approaches are a different class of methods for time series forecasting that involve reconstructing the underlying dynamics of the system from the observed data [1,14,15,23,24]. These methods can potentially overcome some of the limitations of neural network-based and ensemble methods, as they can provide a more interpretable model and do not require extensive hyperparameter tuning. In addition, unlike traditional univariate and

multivariate time series forecasting techniques, this method enables the inclusion of nonlinear behavior in the underlying process.

One popular method for dynamical system reconstruction is the Takens embedding theorem [22], which involves embedding the time series data in a higher-dimensional space to capture the underlying dynamics of the system. Other approaches include delay-coordinate embedding and state-space reconstruction.

A state-space dynamical system usually models the causal relationships between variables using a system of differential equations [3]. Reconstruction methods identify a model of the system's dynamics based on the measurements to estimate the state of the system, which can then be used to predict its future behavior.

The paper proposes a strategy for increasing the predictability and generalization properties of predictive models by combining traditional statistical univariate forecasting approaches with nonlinear dynamic system reconstruction. The suggested technique has the advantage of being able to deal with unevenly distributed and noisy time series data. This is accomplished by additional data creation with spline approximation as well as the decomposition of time series data into dynamic and stochastic components. A dynamic component is used to reconstruct dynamic systems in the form of a system of differential equations, the structure of which is unknown in advance. The stochastic component is fed to the traditional univariate forecasting method like ARIMA. In the end, the predictions from both models are summarized to get the final result.

The remaining part of the paper is structured as follows: Sect. 2 describes time series preprocessing steps, such as data decomposition and regularization; Sect. 3 introduces the components of a hybrid forecasting model and the principles guiding their design; and Sect. 4 compares the proposed approach to established methods on synthetic and real-world time series data.

2 Time Series Preprocessing

Real-world time series often include components with different frequencies, contain noise, or are impacted by external, unknown stochastic factors. Also, the measurements that comprise a time series may contain inconsistent time stamps. These characteristics can affect the effectiveness of a forecasting model. That is why preprocessing steps with the original time series are required.

Recent studies show that incorporating decomposition algorithms with forecasting algorithms usually yields more accurate predictions. What is more, this step is vital for dynamic system reconstruction in state-space form, which is the basis of the proposed hybrid model.

We selected singular spectrum analysis (SSA) over other decomposition methodologies because it does not rely on specific assumptions about the regularity or stationarity of the data. We apply it to decomposing the original time series into dynamic and stochastic components, which are then used independently for ODE-based model identification and statistical model training.

Because ODE reconstruction algorithms are sensitive to the regularity of the data used for identification, we undertake regularization of the dynamical component with spline approximation among other preprocessing steps to generate additional data.

If time series data is of high dimension, it would be ineffective to identify dynamic systems in high-dimensional space. To avoid this, it is proposed to apply clustering-based dimensionality reduction techniques. The steps listed above are discussed in further detail below.

2.1 Time Series Decomposition

Singular spectrum analysis (SSA) is a data-driven technique for analyzing time series data. It involves decomposing a time series into a set of components, each of which represents a different frequency or timescale of the data. SSA is particularly useful for analyzing non-stationary or noisy time series data, as it can effectively separate the signal from the noise.

The SSA algorithm involves several steps. First, the time series data is embedded in a higher-dimensional space using a sliding window technique. Next, the embedded data is decomposed into a set of eigenvectors and eigenvalues using a singular value decomposition. The eigenvectors represent the different components of the time series, while the eigenvalues provide information about their relative importance.

After the decomposition, the components can be reconstructed by combining the eigenvectors with the original time series data. The resulting components can be analyzed separately to identify patterns or trends in the data, or they can be combined to reconstruct the original time series.

One advantage of SSA is that it does not require any prior assumptions about the underlying dynamics of the time series. It is also relatively easy to implement and can be used with a wide range of time series data, including irregularly sampled data. That is why it was selected as a decomposing tool in the proposed algorithm. The basic algorithm consists of two complementary steps: decomposition and reconstruction [10] which are briefly reproduced below for convenience.

Stage 1. Decomposition

Step 1. Embedding. Consider a time series $T = (x(t_0), \ldots, x(t_{M-1})) = (x_0, \ldots, x_{M-1})$ of length M. Let L be a window length, where $1 < L < M$. The embedding procedure forms $K = M - L + 1$ lagged vectors:

$$X_i = (x(t_{i-1}), \cdots, x(t_{i+L-2}))^\top, \quad 1 \le i \le K,$$

of length L. Trajectory matrix $\mathbf{X}$ of time series T has lagged vectors as its columns.

Step 2. Singular value decomposition. The result of this step is the singular value decomposition of the trajectory matrix $\mathbf{X}$. It can be written as:

$$\mathbf{X} = \mathbf{X}_1 + \cdots + \mathbf{X}_d.$$

Here $d = \max\{i : \lambda_i > 0\}$, where $\lambda_1, \ldots, \lambda_L$ are the eigenvalues of matrix $\mathbf{S} = \mathbf{X}\mathbf{X}^\top$ taken in the decreasing order; $\mathbf{X_i} = \sqrt{\lambda_i} U_i V_i^\top$, where $U_1, \ldots, U_L$ is the orthonormal system of the eigenvectors of matrix $\mathbf{S}$, and $V_i = \frac{\mathbf{X}^\top U_i}{\sqrt{\lambda_i}}$.

Stage 2: Reconstruction

Step 3. Grouping. The next step is grouping the terms of the singular value decomposition, which correspond to the separable dynamic (including trend and periodicity) and statistical (residual) components of the series.

The grouping operation divides the set of indices $\{1, \cdots, d\}$ into separate subsets $\{I_1, \cdots, I_m\}$. The previous step's decomposition can be written in a grouped form as follows:

$$\mathbf{X} = \mathbf{X}_{I_1} + \cdots + \mathbf{X}_{I_m}.$$

Here $\mathbf{X}_{I_k} = \mathbf{X}_{i_1} + \cdots + \mathbf{X}_{i_{p_k}}, k = \overline{1, m}, I_k = \{i_1, \ldots, i_{p_k}\}$.

Step 4. Diagonal averaging. Each matrix from the grouped decomposition is converted into a new series of length M in this stage. An $L \times K$ matrix $\mathbf{Y}$ with components of y_{ij} is transformed into a series of $(g_0, \ldots, g_{N-1})$, according to the formula:

$$g_k = \begin{cases} \dfrac{1}{k+1} \displaystyle\sum_{m=1}^{k+1} y^*_{m,k-m+2}, & 0 \le k \le L^* - 1, \\ \dfrac{1}{L^*} \displaystyle\sum_{m=1}^{L^*} y^*_{m,k-m+2}, & L^* - 1 \le k \le K^*, \\ \dfrac{1}{M-k} \displaystyle\sum_{m=k-K^*+2}^{M-K^*+1} y^*_{m,k-m+2}, & K^* \le k \le M. \end{cases}$$

Here $L^* = \min\{L, K\}, K^* = \max\{L, K\}, M = L + K - 1; y^*_{ij} = y_{ij}$ if $L < K$ and $y^*_{ij} = y_{ji}$ if $L \ge K$.

Diagonal averaging applied to matrices X_{I_k} produces series $\widetilde{T}^{(k)} = (\tilde{x}_0^{(k)}, \ldots, \tilde{x}_{M-1}^{(k)})$. The initial series $(x_0, \ldots, x_{M-1})$ is decomposed into the sum of m series:

$$x_n = \sum_{k=1}^{m} \tilde{x}_n^{(k)}.$$

2.2 Time Series Clustering

Real-world data sets typically contain multidimensional time series with large dimensions, making the reconstruction of dynamic systems in high dimensional

space impractical. To address this issue, it is advocated that time series data be clustered based on their similarity, with the premise that one time series has a limited influence on the others. The time series assigned to one cluster are then used to reconstruct intra-cluster dynamic system. The k-Shape [18] technique is used to group data based on correlation.

The k-Shape is a partitional clustering approach that uses an iterative refining procedure similar to the k-means procedure. It employs a shape-based distance measure (SBD) and a centroid computing approach that takes time series shapes into consideration.

The SBD measure is based on cross-correlation with which the similarity of two sequences $\vec{x} = (x_1, \cdots, x_m)$ and $\vec{y} = (y_1, \cdots, y_m)$ can be determined. It can be written as follows:

$$SBD(\vec{x}, \vec{y}) = 1 - \max_w \left(\frac{CC_w(\vec{x}, \vec{y})}{\sqrt{R_0(\vec{x}, \vec{x}) R_0(\vec{y}, \vec{y})}}\right),$$

zero value of which indicates perfect similarity for time series sequences. Here, $CC_w(\vec{x}, \vec{y}) = (c_1, \cdots, c_w) = R_{w-m}(\vec{x}, \vec{y}), \quad w \in \{1, 2, \cdots, 2m-1\}$ is the cross-correlation sequence with length $2m-1$, where $R_{w-m}(\vec{x}, \vec{y})$ is computed, in turn, as:

$$R_k(\vec{x}, \vec{y}) = \begin{cases} \sum_{l=1}^{m-k} x_{l+k} \cdot y_l, & k \geq 0, \\ R_{-k}(\vec{y}, \vec{x}), & k < 0. \end{cases}$$

Centroids are calculated as optimization problems with the goal of minimizing the sum of squared distances to all other time series sequences, as opposed to centroids being calculated as the arithmetic mean of the corresponding coordinates of all sequences of a cluster. The computed sequence, however, is expressed as the maximizer of the squared similarities to all other time series sequences since cross-correlation captures the similarity of time series. A more detailed formulation can be found in [18].

Every time a refinement step is repeated, k-Shape completes two steps: updating the cluster memberships by comparing each time series to all computed centroids and assigning each time series to the cluster with the closest centroid using the SBD measure (assigning step); and updating the cluster centroids to account for the changes in cluster memberships (refinement step). Until there is no change in cluster membership or the allotted number of iterations has been reached, the algorithm repeats these processes.

2.3 Data Regularization Using Splines

In some cases time series data points may be located sparsely or irregularly. To address this problem, it is proposed to perform data regularization, which consists of two successive steps: data approximation; generation of additional data using the approximating function.

Interpolation splines are the main candidate for the role of the approximating function, because, firstly, for the right side of the system of differential equations

to be restored in polynomial form, the use of polynomial functions will give better results than, for example, trigonometric ones, and even more so if there is no reason to assume that the process described by the time series is periodic.

Secondly, considering polynomial functions, if the time interval on which the process is described is large, and there is no reason to consider the function describing the process to be smooth enough, then there is no point in using a single polynomial to approximate this function, and there is no point in improving the quality of approximations through the use of polynomials of high degrees - in this case it is better to use a piecewise function, composed of individual polynomials of low degrees, defined on a part of the global time interval.

Finally, since non-linear continuous functions which are complex differentiable in an open set can be represented as power series in the neighbourhood of any given point with any accuracy, and the partial sums of this series are polynomials, it was decided to use approximation by piecewise polynomial functions.

Cubic spline interpolation, the most popular piecewise-polynomial approximation, employs cubic polynomials between each subsequent pair of nodes [4].

3 Hybrid Forecasting Model

3.1 Dynamic Component Forecasting Based on ODE Reconstruction

The proposed method for the reconstruction of a system of ordinary differential equations in polynomial form uses an approach based on the artificial generation of new data.

Let the values of components $X_j(t), j = (1, n)$, of vector $\mathbf{X}(t) \in \mathbb{R}^n$ be known at time points $t_0, \cdots, t_{M+1}$. It is assumed that the time series corresponds to some dynamic process, which can be represented by a system of ordinary differential equations with constant coefficients:

$$\frac{d\mathbf{X}}{dt} = \sum_{k=0}^{N} P^k \mathbf{X}^{[k]}, \tag{1}$$

where $\mathbf{X}(t) \in \mathbb{R}^n$ is a vector describing the state of the system at time point t, $\mathbf{X}^{[k]}$ is k-th Kronecker power of vector $\mathbf{X}$, N is the maximum Kronecker power to which $\mathbf{X}$ is raised, and P^k are coefficient matrices.

To address the problem of sparse or irregular time series, it is proposed first to approximate the components $X_j(t_i), j = \overline{1, n}, i = \overline{1, M}$ of the measurement vector using splines and then generate additional data points using approximating function.

Matrices P^k are unknown and should be found from available measurements $\mathbf{X}(t_0), \cdots, \mathbf{X}(t_{M+1})$ combined with artificially generated data. Similarly to the SINDy method proposed in [3], P^k are found as solution to sparse regression problem:

$$P = \arg\min_{\tilde{P}} \left(\left\| \dot{\mathbf{X}} - A\tilde{P} \right\|_2 \right) \tag{2}$$

where $\dot{\mathbf{X}}$ is obtained by replacing the derivatives in the left side of (1) with finite differences, A is the matrix of measurements and $P = (P^0, \cdots, P^k)$. This problem can be solved using sparse optimization algorithms, for example, LASSO or sequential thresholded least-squares [3] with additional L_1 regularization term in (2). This helps to identify the most valuable features in the system and eliminate others that are less important.

Considering the assumption that one time series does not affect the other immediately, but with some delay, the right side of the system 1 contains polynomial components with $X_j(t - \tau_j)$, where the delay parameters τ_j are selected experimentally.

Forecasting of the dynamic component of time series is carried out using a numerical solution of the reconstructed system (1).

3.2 Stochastic Component Forecasting Model

The statistic components of time series are approximated and predicted using the ARIMA [13] method. It is based on the use of moving average models and autoregressive models. A linear combination of the variable's previous values is used to forecast the variable of interest in an autoregression model. The formula for an autoregressive model of order p is the following

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t,$$

where ε_t is white noise. This model is referred to as an $AR(p)$ model.

A moving average model uses previous prediction errors in a regression-like model rather than prior values of the forecast variable

$$y_t = c + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t,$$

where ε_t is white noise. This model is referred to as an $MA(q)$ model.

Combining differencing with autoregression and a moving average model, a non-seasonal ARIMA model (AUtoRegressive Integrated Moving Average) is obtained. The full model can be written as

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} t + \varepsilon_t,$$

where y'_t is the differenced series. This model is referred to as an $ARIMA(p, d, q)$ model, where p is the order of the autoregressive part, d is the degree of differencing and q is the order of the moving average part.

Once the model order (p, d, q) has been identified, the parameters $c, \phi_1, \cdots, \phi_p$, $\theta_1, \cdots, \theta_q$ need to be estimated. It can be done by using maximum likelihood estimation (MLE) and Akaike's Information Criterion (AIC).

MLE determines the parameter values that maximize the likelihood of obtaining the observed data. MLE is comparable to the least squares estimates that would be obtained by minimizing the error

$$\sum_t \varepsilon_t^2.$$

AIC can be written as

$$AIC = -2\log L + 2(p + q + k + 1),$$

where L is the likelihood of the data, $k = 1$ if $c \neq 0$ and $k = 0$ if $c = 0$.

3.3 Composite Forecast

To predict future values of time series data, it is necessary to sum up at each moment of time the numerical solution of a dynamical system (1) with predictions based on a stochastic model. It should be emphasized, that time discretization of both parts of hybrid models should be consistent.

4 Experiments

This section contains the results of two numerical experiments: a toy example of identifying a dynamic system using synthetic data and the construction of a hybrid forecasting model based on the Abilene real-world data set.

4.1 Synthetic Data

The time series for the system reconstruction was synthetically generated using the numerical solution $(x(t), y(t))$ of the Van der Pol oscillator ODEs:

$$\begin{cases} \dfrac{dx}{dt} = y, \\ \dfrac{dy}{dt} = \mu(1 - x^2)y - x. \end{cases} \tag{3}$$

Specifically, a particular solution of the system (3) obtained on the interval $[0, 10]$ with the initial condition was $(x_0, y_0) = (1, 4)$ and step $h = 0,01$ used as training data. From it, two time series were formed: a regular one with a given thinning step and an irregular one, for which the required number of points was randomly taken.

A particular solution of the system (3) obtained on the interval $[5, 15]$ with the initial condition was $(x_0, y_0) = (1, 1)$ was used as testing data. The results are shown in the Fig. 1, 2. The graphs on the left show data points used as the input dataset, and the graphs on the right show true phase trajectory $y(x)$ of the system, as well as the trajectory of the reconstructed system using the proposed method and SINDy. In the case with regular data points mean absolute errors of the proposed algorithm and SINDy are 0.008 and 0.028 respectively, and in the case with regular data points they are 0.009 and 0.136 respectively.

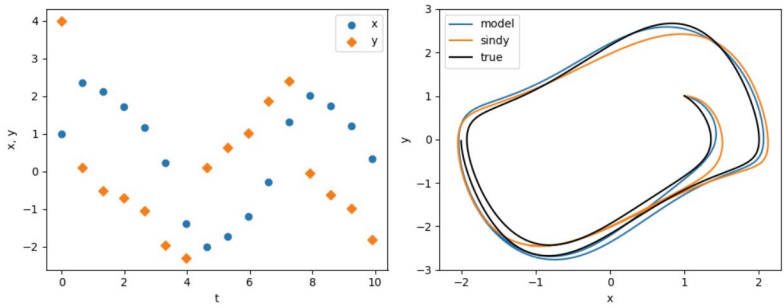


Fig. 1. Reconstruction using 15 regular data points

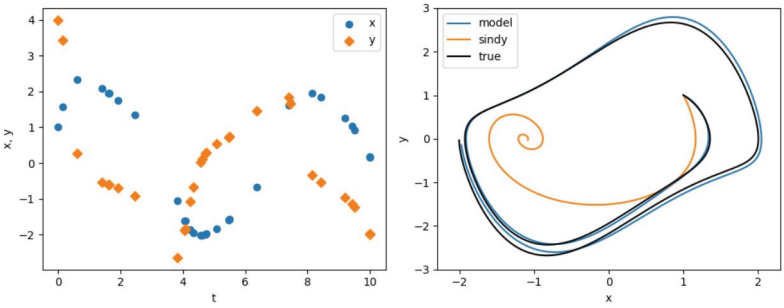


Fig. 2. Reconstruction using 30 irregular data points

4.2 Abilene Real-World Data Set

The proposed hybrid approach is applied to develop a predictive model for the Abilene data set, which describes the transmission of traffic between 12 nodes of the telecommunications network shown in Fig. 3. Each edge means the transmission of traffic between the nodes it connects.

The graphics below depict the intermediate steps described above and applied to the Abilene time series data. Figure 3 shows how the original time series with traffic levels via one edge is divided up into dynamic and stochastic components. Figure 4 demonstrates forecasts with each component of hybrid predictive model as well as the results of a composite forecast.

The proposed technique is contrasted with two well-known time series forecasting methods: ARIMA and SVR. The predicting outcomes from these models under identical conditions are shown in Fig. 5 . The forecasts from the hybrid model are clearly more accurate than those from the ARIMA and SVR models. To support this conclusion, we use a recursive approach to examine the models for multi-step forecasting tasks. The forecasts are created 1, 2, 3, 4, 6, 8, 12, and 24 h ahead. Mean Absolute Error (MAE) values for the proposed hybrid model, ARIMA, and SVR models are shown in Figs. 6–8, respectively. In these figures, values along ax x correspond to the transmission network's edges (traffic between

two linked nodes). It is clear that, in addition to having reduced error values, the suggested hybrid model produces prediction results with an error that is robust to the forecasting horizon. Table 1 summarizes the numerical values of MAE and Mean Absolute Percentage Error (MAPE) computed as the average between all the edges in the network for the compared models.

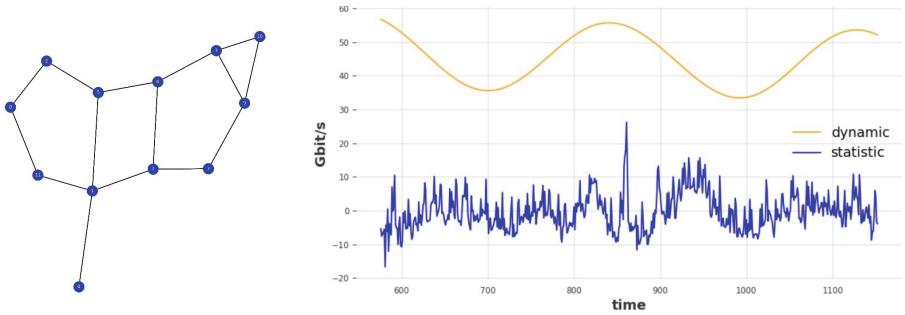


Fig. 3. Abilene network scheme and example of one edge time series decomposition

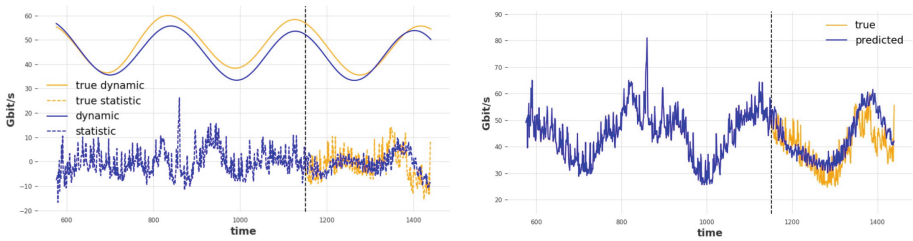


Fig. 4. Forecasting with dynamic and stochastic components forecast (left) and a composite forecast (right)

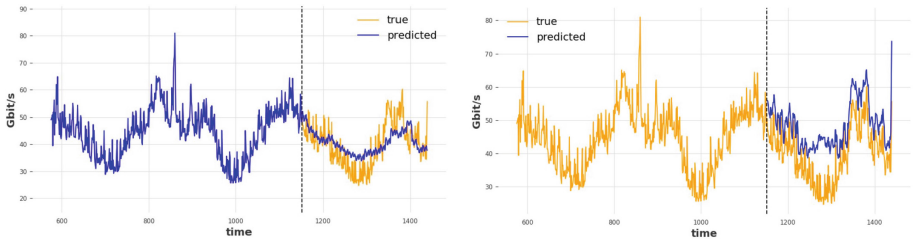


Fig. 5. Forecasting with ARIMA single model (left) and forecasting using SVR single model (right)

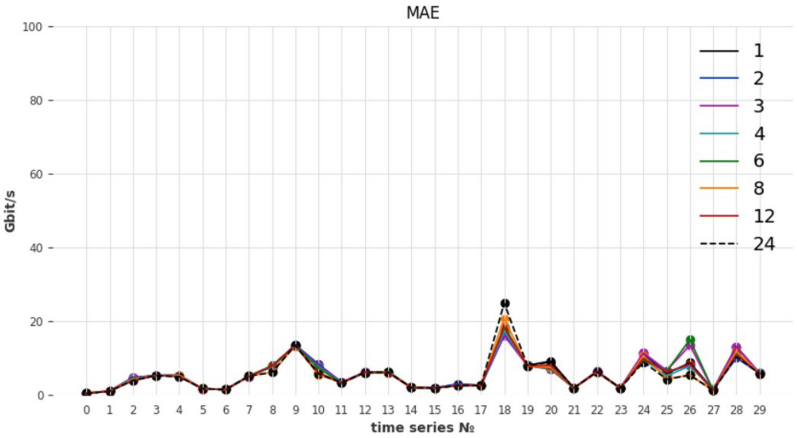


Fig. 6. Proposed hybrid model MAE

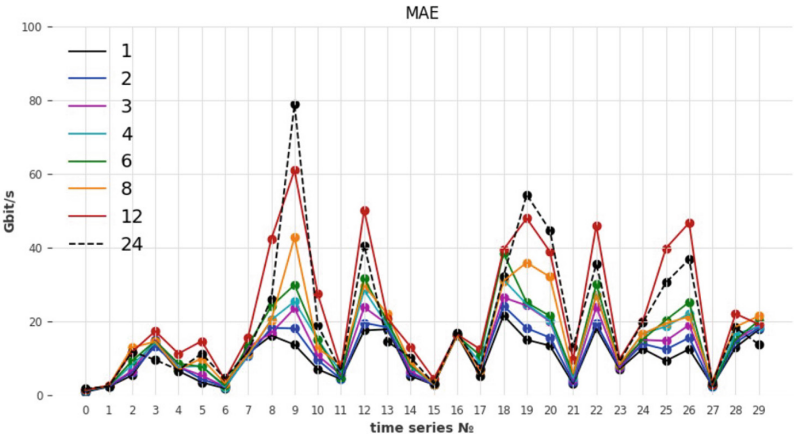


Fig. 7. ARIMA MAE

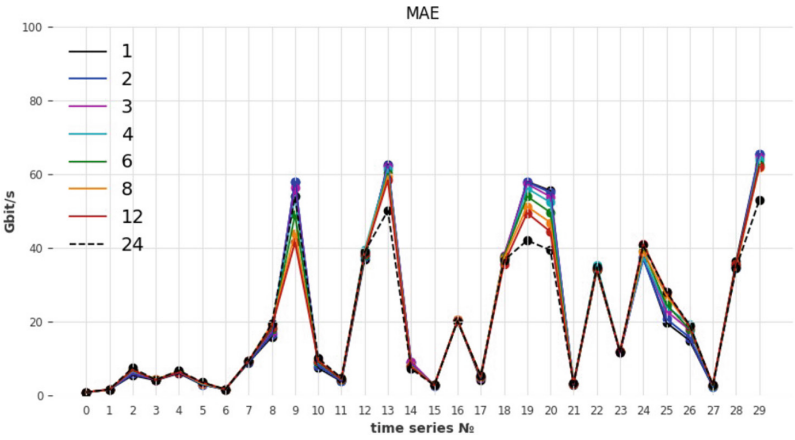


Fig. 8. SVR MAE

Table 1. Mean MAE and MAPE values for hybrid, ARIMA, and SVR models in multi-step forecasting up to 24 h

Step	Our method		ARIMA		SVR	
	MAE Gbit/s	MAPE %	MAE Gbit/s	MAPE %	MAE Gbit/s	MAPE %
1	5.64	7.72	9.93	13.75	19.25	21.41
2	5.45	7.57	11.12	15.37	19.36	21.61
3	5.72	7.76	12.59	16.81	19.61	21.82
4	5.37	7.47	13.89	18.19	19.78	21.99
6	5.66	7.58	15.02	19.25	20.01	22.12
8	5.28	7.39	15.82	20.53	19.21	21.13
12	5.38	7.29	22.55	26.27	20.12	22.26
24	5.36	7.15	19.82	25.87	16.75	18.95

5 Conclusion

The research provides a method for multivariate time series forecasting that combines decomposition, approximation, and statistical modeling approaches, as well as methods for nonlinear dynamic system reconstruction and sparse optimisation. The experimental findings reveal that employing the suggested technique on synthetic and real-world data sets resulted in greater prediction accuracy than SINDy, ARIMA, and SVR. In addition, the suggested hybrid model is verified for extrapolation properties in multi-step forecasting, where it shown error robustness to forecasting horizon in contrast to ARIMA and SVR.

References

1. Bongard, J., Lipson, H.: Automated reverse engineering of nonlinear dynamical systems. Proc. Natl. Acad. Sci. **104**(24), 9943–9948 (2007)
2. Box, G.E.P., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time Series Analysis: Forecasting and Control. Wiley series in probability and statistics, Wiley, Hoboken. fifth edn. (2016)
3. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. **113**(15), 3932–3937 (2016)
4. Burden, R.L., Faires, J.D., Burden, A.M.: Numerical Analysis. Cengage Learning. Boston 10 edn. (2016)
5. Chang, Y.Y., Sun, F.Y., Wu, Y.H., Lin, S.D.: A memory-network based solution for multivariate time-series forecasting (2018)
6. Fan, G.F., Zhang, L.Z., Yu, M., Hong, W.C., Dong, S.Q.: Applications of random forest in multivariable response surface for short-term load forecasting. Int. J. Electr. Power Energy Syst. **139**, 108073 (2022)
7. Galicia, A., Talavera-Llames, R., Troncoso, A., Koprinska, I., Martínez-Álvarez, F.: Multi-step forecasting for big data time series based on ensemble learning. Knowl.-Based Syst. **163**, 830–841 (2019)
8. Geng, X., He, X., Xu, L., Yu, J.: Graph correlated attention recurrent neural network for multivariate time series forecasting. Inf. Sci. **606**, 126–142 (2022)
9. Girard, A., Rasmussen, C., Candela, J.Q., Murray-Smith, R.: Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting. In: Advances in Neural Information Processing Systems, vol. 15. MIT Press (2002)
10. Golyandina, N., Nekrutkin, V., Zhigljavsky, A.A.: Analysis of Time Series Structure. Chapman and Hall/CRC, London (2001)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. Huang, S.C., Le, T.H.: Chapter 8 - convolutional neural network architectures. In: Huang, S.C., Le, T.H. (eds.) Principles and Labs for Deep Learning, pp. 201–217. Academic Press (2021)
13. Hyndman, R.J., Athanasopoulos, G.: Forecasting: Principles And Practice. Otexts, Online Open-Access Textbooks, Melbourne, Australia, third print edition edn (2021)
14. Kramer, D., Bommer, P.L., Tombolini, C., Koppe, G., Durstewitz, D.: Reconstructing Nonlinear Dynamical Systems from Multi-Modal Time Series (2022)
15. Kutz, J.N., Brunton, S.L., Brunton, B.W., Proctor, J.L.: Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems. Society for Industrial and Applied Mathematics, Philadelphia (2016)
16. Lainder, A.D., Wolfinger, R.D.: Forecasting with gradient boosted trees: augmentation, tuning, and cross-validation strategies: winning solution to the M5 uncertainty competition. Int. J. Forecast. **38**(4), 1426–1433 (2022)
17. Lütkepohl, H.: New Introduction to Multiple Time Series Analysis. Springer, Heidelberg (2005)
18. Paparrizos, J., Gravano, L.: k-shape: efficient and accurate clustering of time series. ACM SIGMOD Rec. **45**(1), 69–76 (2016)
19. Sapankevych, N.I., Sankar, R.: Time series prediction using support vector machines: a survey. IEEE Comput. Intell. Mag. **4**(2), 24–38 (2009). conference Name: IEEE Computational Intelligence Magazine

20. Shah, S.Y., et al.: AutoAI-TS: AutoAI for time series forecasting (2021)
21. Song, W., Fujimura, S.: Capturing combination patterns of long- and short-term dependencies in multivariate time series forecasting. Neurocomputing **464**, 72–82 (2021)
22. Stark, J., Broomhead, D., Davies, M., Huke, J.: Takens embedding theorems for forced and stochastic systems. Nonlinear Anal. Theory Methods Appl. **30**(8), 5303–5314 (1997)
23. Tandeo, P., Ailliot, P., Sévellec, F.: Data-driven reconstruction of partially observed dynamical systems. preprint, Predictability, probabilistic forecasts, data assimilation, inverse problems/Climate, atmosphere, ocean, hydrology, cryosphere, biosphere/Theory (2022)
24. Williams, M.O., Kevrekidis, I.G., Rowley, C.W.: A data-driven approximation of the Koopman operator: extending dynamic mode decomposition. J. Nonlinear Sci. **25**(6), 1307–1346 (2015)