Feature generation for multiscale time series forecasting

***Abstract*—The paper presents a framework for the massive multiscale time series forecast. We propose a method of constructing efficient feature description for the corresponding regression problem. The method involves feature generation and dimensionality reduction procedures. Generated features include historical information about the target time series as well as other available time series, local transformations and multiscale features. We apply several forecasting algorithms to the resulting regression problem and investigate the quality of the forecasts for various horizon values.**

I. Introduction

We focus on the problem of forecasting behavior of a device within the concept of Internet of Things. The device at question is monitored by a set of sensors, which produces large amount of multi-scale time series during its lifespan. These time series have various time scales since distinct sensors produce observations with various frequencies from milliseconds to weeks. The main goal is to predict the observations of a device in a given time range.

We assume that the sampling rate of each time series is fixed and each time series has its own forecast horizon. The problem of multi-scale analysis arises in such applications as weather prediction, medical diagnosis and monitoring various sensor time series [1], [2], [3], [4]. Motivation for multi-scale analysis comes from the assumption that the behaviour of complex signals may be governed by essentially different processes at various time scales. Thus, the time series should be modeled separately at each scale. This approach is used in time series classification, prediction and fault detection [5], [3], [6]. Regardless of the goal of multi-scale analysis, it includes sequential averaging of the time series to obtain more coarse-scaled time series [7], or, more rarely, differencing the time series for a more detailed, fine-scaled version of the time series [8]. Averaging and differencing, which is equivalent to application of Haar's wavelet transform [8], may be replaced by any other pair of low and high pass wavelet filters [9] or convolution operation with some kernel function [10]. Using multi-scale approach in time series prediction usually involves determining optimal scales [10], [2], decomposition of time series into separately forecasted components and combination of the obtained forecasts.

The problem gets more complex when the task is to forecast multiple time series, which are characterized with different scales and ranges. Forecasting time series separately might lead to loss of valuable information. On the other hand, forecasting the time series simultaneously might lead to increased errors since not all the time series in the given set necessarily depend on the others. In this paper we propose a novel framework for multiscale time series forecasting, which is based on regression-based forecasts. The goal is to obtain forecasts of all time series from the given set simultaneously. Adopting this approach we endeavour to profit as much as possible from the interconnections between the time series of the set while keeping the decrease in forecasting quality for the independent time series reasonably small. Within the proposed framework the time series of various scales are combined into are single regression problem. The forecasts viewed as target variables of the regression problem, where feature description contains local history of the time series as well as various derivations. We describe the steps of creating feature description to this problem: composition of design matrix, feature generation and selection. Note that the problem of model selection rests beyond the scope of the paper. To illustrate the proposed framework in application to the multiscale data set [11] we use several widely used regression models [12], [13], [14], [15], [16], [17]. The following section provides a brief overview of these methods and provide the motivation to use them.

II. Related work

Along with generic methods of time series forecasting, such as Autoregressive Moving Average Models (ARMA), Autoregressive Integrated Moving Average Models (ARIMA), authors report high predictive performance of the methods, originally developed for classification or regression, applied to forecast time series [12], [13], [14], [15], [18], [16]. Here the input variables are the delayed observations of the time series, and the output is the forecasted value of time series. However, the authors of [15] show that this prediction framework suffers from systematic error that does not converge to zero as the sample size increases, and ensure error convergence applying cubic spline approximation to noisy data, which yields much lower RMSE in case of noisy data.

To extend this one-step-ahead forecasting scheme to the case of multiple predictions, one may use iterative, direct or multiple output strategies [19]. Within the iterative strategy, one-step-ahead forecasts are computed recursively, with the newly predicted values of the time series used as the actual future records. A less prone to error accumulation, though more time consuming method is the direct strategy, which involves estimation of h models to predict h future values of the time series [20]. Finally, the multiple input multiple output (MIMO) strategy allows to obtain h prediction with at one step. The paper [19] compares different strategies of multi-step-ahead prediction in SVR-based forecasting: direct, iterative and multiple output. Regardless of the horizon values, direct and MIMO strategies consistently achieve more accurate forecasts, than the iterative strategy, with MIMO being most accurate in most cases.

To demonstrate the application of the proposed framework of time series forecasting, we utilize Multivariate Linear Regression (MLR) as the naive approach, as well as three more complex models: Random Forests (RF) [12], [13], Support Vector Regression (SVR) [14], [15], [21] and artificial neural networks (ANN) [18], [16]. Random Forests combine decision trees with randomly generated nodes to increase the accuracy of classification or regression [22]. In case of regression trees, each node of the tree splits the input space into two subspaces and each leaf specifies a distinct regression model, which is used for prediction if the input is found in the corresponding region of the input space. Predictions of the trees in the forest are averaged, or, for the probabilistic random forest, the probabilities of the outputs are averaged. The advantage of random forests is their efficiency in case of highly dimensional data due to the randomness incorporated into selecting informative features. Since random forests are essentially ensembles of weak learners, they enjoy high generalization ability, associated with boosting algorithms. Similarly, the formulation of optimization problem within support vector regression promotes its robustness in case of highly dimensional data. The authors of [14], [21] reported high predictive performance of SVR applied to time series forecasting. In case of SVR, MIMO strategy is based on multivariate SVR [23]. Finally, artificial neural networks attract researches and practitioners from various domains [16], [17]. One of the reasons for that is the ability of ANNs to model complex relationships between the input data in such fashion that does not require direct feature engineering. For more suggestions on how to combine these forecasting methods [17], [24] or use them in the multiscale fashion we refer the reader to [9], [25], [5], [26], [27].

III. Problem statement

Consider a large set of time series $\mathfrak{D} = \{\mathbf{s}^{(q)} | \; q = 1, \dots, Q\}$, where each real-valued time series $\mathbf{s}$

$$\mathbf{s} = [s_1, \dots, s_i, \dots, s_T], \quad s_i = s(t_i), \quad 0 \leq t_i \leq t_{\max}$$

is a sequence of observations $s_i = s(t_i)$ of some real-valued signal $s(t)$. Each time series $\mathbf{s}^{(q)}$ has its own sampling rate $1/\tau^{(q)}$:

$$t_i^{(q)} = i \cdot \tau^{(q)}.$$

The task is to obtain forecasts $\hat{s}(t_i)$ of $\mathbf{s} \in \mathfrak{D}$ for $\Delta t_{\text{r}} < t_i \leq T_{\max} + \Delta t_{\text{r}}$, given the set $\mathfrak{D}$ (see Fig. 1). The forecasts $\hat{\mathbf{s}}$ should minimise symmetric mean absolute percentage error:

$$SMAPE(\mathbf{s}, \hat{\mathbf{s}}) = \frac{1}{r} \sum_{i=1}^{r} \frac{2|s_i - \hat{s}_i|}{|s_i + \hat{s}_i|}. \tag{1}$$

Here and throughout this paper we assume that each time series are standardized.

A. Design matrix

We consider the forecasting problem as the multivariate regression problem, where target variables are the vectors of lagged values $s(t_i)$ of all the time series $\mathbf{s} \in \mathfrak{D}$.

Let $\mathbf{x}^*$ denote rows of the design matrix $\mathbf{X}^*$ for the regression problem. Each vector $\mathbf{x}^* = [\mathbf{x}|\mathbf{y}]$ collects all the time series over the time period Δt_{p} (Fig. 2), which stands for the

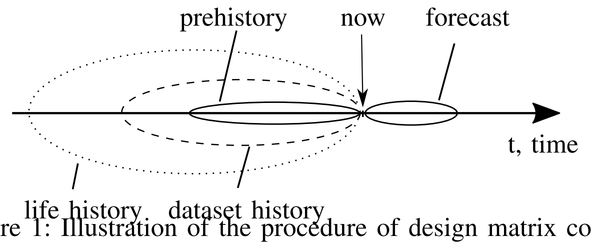


Figure 1: Illustration of the procedure of design matrix composition.

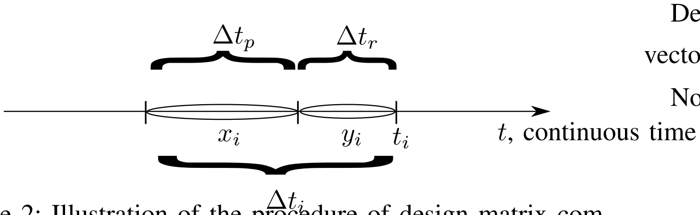


Figure 2: Illustration of the procedure of design matrix composition.

local *prehistory*. The vector $\mathbf{x}^*$ includes samples from previous history of time series from $\mathfrak{D}$ as well as any derivatives or *generated features*. We describe the types of generated features in Section IV.

The design matrix $\mathbf{X}^*$ for the multiscale autoregressive problem statement is constructed as follows. Let $\mathbf{s}_i^{(q)}$ denote the i-th segment of the time series $\mathbf{s}^{(q)}$

$$[\mathbf{x}_i^{(q)}|\mathbf{y}_i^{(q)}] = \quad (2)$$

$$\underbrace{s^{(q)}(t_i - \Delta t_{\mathrm{r}} - \Delta t_{\mathrm{p}}), \dots,}_{\mathbf{x}_i^{(q)}} \underbrace{s^{(q)}(t_i - \Delta t_{\mathrm{r}}), \dots, s^{(q)}(t_i))}_{\mathbf{y}_i^{(q)}}],$$

where $s^{(q)}(t)$ is an element of time series $\mathbf{s}^{(q)}$. To construct the design matrix, select t_i, $i = 1, \dots, m$ from $G = \{t_1, \dots, t_T\}$ such that segments $\mathbf{s}_i = [\mathbf{x}_i|\mathbf{y}_i]$ cover time series $\mathbf{s}$ without intersection in target parts $\mathbf{y}_i$:

$$|t_{i+1} - t_i| > \Delta t_{\mathrm{r}}. \quad (3)$$

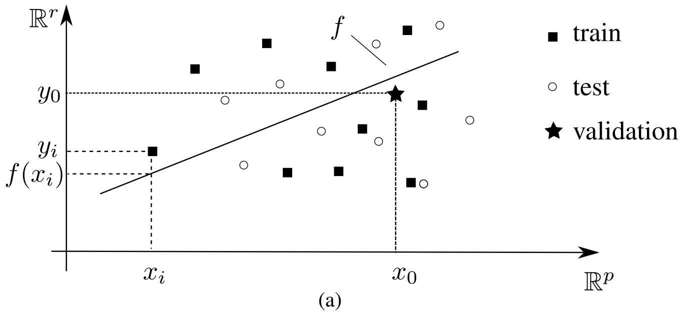


Figure 3: Forecasting as regression problem.

Following (2) and (3), extract segments $[\mathbf{x}_i^{(q)}|\mathbf{y}_i^{(q)}]$, $i = 1, \dots, m$ from all time series $\mathbf{s}^{(q)} \in \mathfrak{D}$ and form the matrix

$$\mathbf{X}^* = \left[\begin{array}{c|c} \underset{1\times n}{\mathbf{x}} & \underset{1\times r}{\mathbf{y}} \\ \hline \underset{m\times n}{\mathbf{X}} & \underset{m\times r}{\mathbf{Y}} \end{array}\right] = \quad (4)$$

$$\left[\begin{array}{ccc|ccc} \mathbf{x}_m^{(1)} & \dots & \mathbf{x}_m^{(Q)} & \mathbf{y}_m^{(1)} & \dots & \mathbf{y}_m^{(Q)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_1^{(1)} & \dots & \mathbf{x}_1^{(Q)} & \mathbf{y}_1^{(1)} & \dots & \mathbf{y}_1^{(Q)} \end{array}\right].$$

Denote a row from the pair $\mathbf{Y}, \mathbf{X}$ as $\mathbf{y}, \mathbf{x}$ and call these vectors the target and the features.

Now we are able the regression problem as follows:

$$\hat{\mathbf{y}} = f(\mathbf{x}, \hat{\mathbf{w}}), \ \hat{\mathbf{w}} = \underset{\hat{\mathbf{w}}}{\arg\min}\, S\big(\mathbf{w}|\mathbf{f}(\mathbf{w}, \mathbf{x}), \mathbf{y}\big). \quad (5)$$

Here the error function is given by $SMAPE$ (1) for each segment $[\mathbf{x}_i|\mathbf{y}_i]$, averaged over all segments $i = 1, \dots, m$ in the test set:

$$S\big(\mathbf{w}|\mathbf{f}(\mathbf{w}, \mathbf{x}), \mathbf{y}\big) = \frac{r}{m} \sum_{i=1}^{m} SMAPE(\mathbf{y}_i, f(\mathbf{x}_i, \mathbf{w})).$$

IV. Feature generation

Denote the generated feature vector as $\boldsymbol{\phi}$. This vector consists of concatenated row-vectors $\boldsymbol{\phi} = [\boldsymbol{\phi}^{(1)}, \dots, \boldsymbol{\phi}^{(Q)}]$, which corresponds to time series local histories $\mathbf{s} = [\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(Q)}]$, modified with set of transformations G. The elements $g : \mathbf{s} \to \boldsymbol{\phi}$ of this set are listed below. The augmented feature set $\boldsymbol{\phi}$ includes

1) the local history of all time series themselves,
2) transformations (non-parametric and parametric) of local history,
3) parameters of the local models,
4) distances to the centroids of local clusters.

A. *Transformations of local history*

We use non-parametric and parametric functions to generate features. The purpose of this block of features is to introduce nonlinearities into the feature space of regression problem (5).

The parametric procedure involves two optimization problems. The first one fixes the vector $\hat{\mathbf{b}}$, collected over all the primitive functions $g = g(\mathbf{b}, s) \in G$, which generate features $\boldsymbol{\phi}$:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\arg\min}\, S\big(\mathbf{w}|\mathbf{f}(\mathbf{w}, \boldsymbol{\phi}), \mathbf{y}\big), \quad \text{where} \quad \boldsymbol{\phi} = g(\hat{\mathbf{b}}, \mathbf{s}).$$

The second one optimizes the transformation parameters $\hat{\mathbf{b}}$ given the obtained model parameters $\mathbf{w}$

$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b}} S\big(\mathbf{b}|\mathbf{f}(\hat{\mathbf{w}}, \boldsymbol{\phi}), \mathbf{y}\big).$$

This parametric feature generation procedure repeats these problems until vectors $\hat{\mathbf{w}}, \hat{\mathbf{b}}$ converge. The initial values of the parameters $\mathbf{b}$ are assigned empirically.

B. Convolutions, statistics and parameters of local history

This block of feature generation functions includes convolutions, time averaging and differencing, and basic statistics of each time series, such as mean and standard deviation, minimum and maximum of the input $\mathbf{x}$. The features from these part can be seen as applying Haar's wavelet transform to each segment [8]. Motivation for this comes from assuming the multi-scale nature of the time series: complex signals may be governed by essentially different processes at various time scales. Averaging of the time series allows to obtain more coarse-scaled time series, while differencing the time series provides a more detailed, fine-scaled version of the time series.

C. Parameters of local history forecast

For the time series $\mathbf{s}$ construct the Hankel matrix [28] with a period k and shift p, so that for $\mathbf{s} = [s_1, \dots, s_T]$ the matrix

$$\mathbf{H}^* = \left[\begin{array}{c|cc} s_T & \dots & s_{T-k+1} \\ \vdots & \ddots & \vdots \\ s_{k+p} & \dots & s_{1+p} \\ s_k & \dots & s_1 \end{array}\right], \text{ where } 1 \geqslant p \geqslant k.$$

Reconstruct the regression to the first column of the matrix $\mathbf{H}^* = [\mathbf{h}, \mathbf{H}]$ and denote its least square parameters as the feature vector

$$\boldsymbol{\phi}^{(q)} = \arg\min \|\mathbf{h} - \mathbf{H}\boldsymbol{\phi}\|_2^2. \qquad (6)$$

For the time series $\mathbf{s}^{(q)}$, $q = 1, \dots, Q$ use the parameters $\boldsymbol{\phi}^{(q)}$ as the features.

D. Distances to centroids of local clusters

This procedure applies the kernel trick to the time series. For given local history time series $\mathbf{x}_i^{(q)}$, $q = 1, \dots, Q$ compute k-means centroids $\mathbf{c}_p^{(q)}$, $p = 1, \dots, P$. With the selected k-means distance function ρ construct the feature vector

$$\boldsymbol{\phi}_i^{(q)} = [\rho(\mathbf{c}_1^{(q)}, \mathbf{s}_i^{(q)}), \dots, \rho(\mathbf{c}_P^{(q)}, \mathbf{s}_i^{(q)})] \in \mathbb{R}_+^P. \qquad (7)$$

This k-means of another clustering procedure may use internal parameters, so that there are no parameters to be included to the feature vector or to the forecasting model.

Algorithm 1: Initial train-test splitting procedure.

```
Data: Object-feature matrix $\mathbf{X}^* \in \mathbb{R}^{m\times(n+r)}$. Train to
      test ratio $\alpha \in [0, 1]$.
Result: Train and test, $\mathbf{X}^*_{\text{train}}$, $\mathbf{X}^*_{\text{test}}$.
Set train set and test set sizes:
  $m_{\text{train}} = \lfloor \alpha \cdot m \rfloor, \quad m_{\text{test}} = m - m_{\text{train}}$ ;
Decompose matrix $\mathbf{X}^*$ into train and test matrices $\mathbf{X}^*_{\text{train}}$,
$\mathbf{X}^*_{\text{test}}$:
  $\mathbf{X}^* = \left[\begin{array}{c|c} \underset{m_{\text{test}}\times n}{\mathbf{X}_{\text{test}}} & \underset{m_{\text{test}}\times r}{\mathbf{Y}_{\text{test}}} \\ \underset{m_{\text{train}}\times n}{\mathbf{X}_{\text{train}}} & \underset{m_{\text{train}}\times r}{\mathbf{Y}_{\text{train}}} \end{array}\right]$
```

V. Testing procedure

The algorithm below describes the procedure used to evaluate the forecasting errors within the proposed framework given the model $\mathbf{f}$, data matrix $\mathbf{X}^* \in \mathbb{R}^{m\times(n+r)}$ and fixed parameters train to test ratio α, minimal sample (test) size $m_{\min}$. This procedure involves creation of design matrix (4), generation of augmented feature description $\boldsymbol{\phi}$ and, since it is likely to be redundant, dimensionality reduction. Here we use principal component analysis (PCA) and nonlinear PCA [29].

1) Create design matrix $\mathbf{X}^*$ according to (4) from $\mathfrak{D}$.
2) Split matrix $\mathbf{X}^*$ into train and test matrices $\mathbf{X}^*_{\text{train}}$ and $\mathbf{X}^*_{\text{test}}$ according to the train-test splitting procedure 1
3) Augment $\mathbf{X}^*_{\text{train}}$ with generated features $\boldsymbol{\phi}$
4) Reduce dimensionality of $\mathbf{X}^*_{\text{train}}$
5) Optimize hyper parameters of the model $\mathbf{f}$, using $\mathbf{X}^*_{\text{train}}$
6) For k in $\{1, \dots, m_{\text{test}} - m_{\min}\}$ repeat:
   - define $\mathbf{X}^*_{\text{train},i}$ as $(i+1)$-th to $(i + m_{\min} + 1)$-th rows of $\mathbf{X}^*_{\text{test}}$ and $\mathbf{x}^*_{\text{val},i}$ as the i-th row of $\mathbf{X}^*_{\text{test}}$

$$\mathbf{X}^*_{\text{test}} = \left[\begin{array}{c|c} \dots & \dots \\ \hline \underset{1\times n}{\mathbf{x}_{\text{val},i}} & \underset{1\times r}{\mathbf{y}_{\text{val},i}} \\ \underset{m_{\min}\times n}{\mathbf{X}_{\text{train},i}} & \underset{m_{\min}\times r}{\mathbf{Y}_{\text{train},i}} \\ \hline \dots & \dots \end{array}\right]$$

   - apply feature transformation to $\mathbf{X}^*_{\text{train},i}$, $\mathbf{X}^*_{\text{val},i}$
   - train forecasting model $\mathbf{f}(\mathbf{x}, \hat{\mathbf{w}}_i)$, using $\mathbf{X}^*_{\text{train},i}$

Table I: Regression models.

Model name	Hyper parameters
Baseline method: $\hat{s}_i = s_{i-1}$	None
Multivariate linear regression (MLR) with l_2-regularization	Regularization coefficient: 2
Support vector regression with multiple output (MSVR)	Kernel type: RBF, p_1: 2, p_2: 0, γ: 0.5, λ: 4
Artificial neural network (ANN). Feed-forward ANN with single hidden layer	Hidden layers size: 25
Random forest (RF)	Number of trees: 25 , number of variables for each decision split: 48

- obtain vector of residuals $\boldsymbol{\varepsilon} = \mathbf{y}_{\text{val},i} - \mathbf{f}(\mathbf{x}_{\text{val},i}, \hat{\mathbf{w}}_i)$
- compute forecasting quality:

$$SMAPE(i) = \frac{1}{r}\sum_{t=1}^{r}\frac{2|\varepsilon_t|}{|2(y_{\text{val},i})_t - \varepsilon_t|};$$

7) Return $SMAPE$, averaged over data splits:

$$\text{Error} = \frac{1}{m_{\text{test}} - m_{\min}}\sum_{i=1}^{m_{\text{test}}-m_{\min}} SMAPE(i).$$

The models that we use are listed in the table I along with the optimized hyper parameters.

VI. Computational experiment

This section presents the results of computational validation of the proposed framework.

A. Datasets

The computational experiments demonstrated in this section are based on the Energy-Weather data set [11]. The dataset consists of the Polish electricity load time series and weather time series in Warsaw (Longtitude: 21.25, Latitude: 52.30, Elevation: 94). Energy time series contain hourly records (total of 52512 observations), while weather time series were measured daily and contain 2188 observations. The multiscale time series correspond to the period of 1999 to 2004. The results observed on this data set are illustrative of the proposed framework since the data set contains the time series that are both multiscale and have various nature.

The Energy-Weather data set was used to generate several data sets with artificial inserted missing values. The ratios of missing data are 0.01, 0.03, 0.05 and 0.1.

B. Experimental results

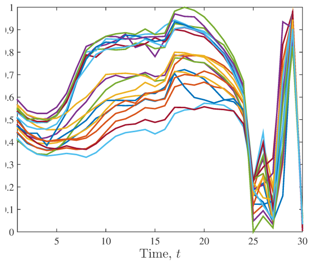
Fig. 4 displays a range of target variables $\mathbf{y}$ generated for the Energy-Weather data set.

Fig. 5 demonstrate the examples of forecasts of individual time series, obtained within the proposed framework. Here the design matrix was augmented with the generated features and PCA was applied to select a subset of features.

Table II lists forecasting errors for the proposed feature generation strategies applied to time series from the original Energy-Weather data set. The errors were computed following the testing procedure, detailed in the section V. After the multiple forecasts were obtained, $SMAPE$ was computed for each time series separately. Multirows labeled "Features" unite results of each model for the particular feature set. The tested options are:

- "History" corresponds to the standard regression-based forecast with no additional features.
- Each multirow from "SSA" to "NW" corresponds to a particular feature set added to historical features separately from other generated features. Here "SSA" stands for parameters of local approximation (6), "Cubic" stands for coefficients of cubic spline approximation, "Conv" — for multiscale features and statistics listed in section and "Centroids" — for the feature set defined by (7).
- "All" stands for all feature generation strategies applied to the dataset, with no feature selection.
- "PCA" and "NPCA" present the results of applying PCA and NPCA after all generation strategies were used.

The top row of Table II lists results of the baseline method: for each time series the next forecasted value is predicted with the most recent observed value. As can be seen from the table, the forecasting quality generally improves for all the time series, even though the weather data is unlikely to depend on the energy consumption and the multiple all-on-all regression could lead to increased errors. The errors of the data sets with missing values increase as the ratio of missing data gets higher but the general pattern does not change. According to our results the feature sets differ very slightly among feature generation strategies and generally demonstrate poorer performance than the historical features, though this is not always the case. We also note that there is no single best or worst combination of model, feature generation and feature



(a)

Figure 4: (a) Target variables of the design matrix composed of the time series from the Energy-Weather data set. (b) Forecasting results for Energy-Weather.

selection strategy for all the time series. This motivates us to direct our further research to ensembles of learners.

VII. Discussion and conclusion

In this paper we have suggested a framework for multi-scale time series forecast. The proposed framework employs regression-based approach combined with feature generation. We have found that even for such naive approach the results are still better then those of the baseline method. Though the results are somewhat discouraging, we expect further improvement associated with application of mixtures of experts [30], ensembles of weak learners, where each learner is relevant to some subspace of the feature space. For this reason we introduce such feature generation strategies, based on local approximation parameters and distances to centroids: these kinds of features have proven efficient in time series classification problems [31].

References

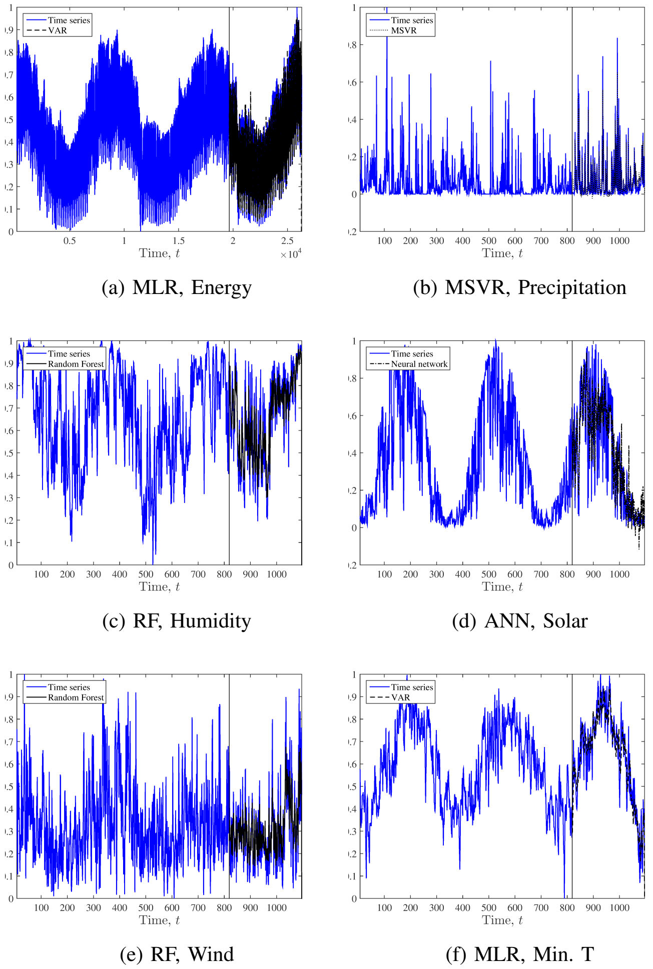
[1] M. D. Costa, C.-K. Peng, and A. L. Goldberger, “Multiscale analysis of heart rate dynamics: Entropy and time irreversibility measures,” *Cardiovascular Engineering*, vol. 8, no. 2, pp. 88–93, 2008.

[2] M. U. Ahmed, N. Rehman, D. Looney, T. M. Rutkowski, P. Kidmose, and D. P. Mandic, “Multivariate entropy analysis with data-driven scales,” *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3901 – 3904, 2012.

[3] P. Cortez, M. Rio, M. Rocha, and P. Sousa, “Multi-scale internet traffic forecasting using neural networks and time series methods,” *Expert Systems*, vol. 29, no. 2, pp. 143–155, 2012.

[4] M. A. R. Ferreira, D. M. Higdon, H. K. H. Lee, and M. West, “Multi-scale and hidden resolution time series models,” *Bayesian Analysis*, vol. 1, no. 4, pp. 947–967, 2006.

[5] Z. Cui, W. Chen, and Y. Chen, “Multi-scale convolutional neural networks for time series classification,” *Computer Vision and Pattern Recognition*, 2016.



(a) MLR, Energy

(b) MSVR, Precipitation

(c) RF, Humidity

(d) ANN, Solar

(e) RF, Wind

(f) MLR, Min. T

Figure 5: Forecasting results for original Energy-Weather data set with all feature generation strategies applied and PCA feature selection.

[6] C. Aldrich and L. Auret, *Process Monitoring and Fault Diagnosis with Machine Learning Methods (Advances in Computer Vision and Pattern Recognition)*. Springer London, 2013, ch. Process Monitoring Using Multiscale Methods, pp. 341–369.

[7] S.-D. Wu, C.-W. Wu, S.-G. Lin, C.-C. Wang, and K.-Y. Lee, “Time series analysis using composite multiscale entropy,” *Entropy*, vol. 15, no. 3, pp. 1069–1084, 2013.

[8] Y. Jiang, C.-K. Peng, and Y. Xu, “Hierarchical entropy analysis for biological signals,” *Journal of Computational and Applied Mathematics*, vol. 236, p. 728742, 2011.

[9] H. Chen, B. Vidakovic, , and D. Mavris, “Multiscale forecasting method using armax models,” Georgia Institute of Technology, Tech. Rep., 2004.

[10] U. Vespier, A. Knobbe, S. Nijssen, and J. Vanschoren, *Machine Learning and Knowledge Discovery in Databases*, ser. Lecture Notes in Computer Science, 2012, vol. 7524, ch. MDL-Based Analysis of Time Series at Multiple Time-Scales, pp. 371–386.

[11] [Online]. Available: http://gdudek.el.pcz.pl/varia/stlf-data

[12] R. Yu, Y. Yang, L. Yang, G. Han, and O. A. Move, “Raqa random forest

SMAPE	Data	Energy		Max T.		Min T.		Precipitation		Wind		Humidity		Solar	
Features	Models	test	train	test	train	test	train	test	train	test	train	test	train	test	train
History	Baseline	0.1948	0.2095	0.1040	0.1351	0.1047	0.1141	1.2034	1.2908	0.4581	0.4600	0.1803	0.1918	0.4641	0.5184
History	MLR	0.130	0.143	0.090	1.265	0.395	0.239	0.673	0.108	0.079	0.057	1.164	0.358	0.168	1.247
	MSVR	0.280	0.377	0.227	1.243	0.415	0.369	0.780	0.025	0.055	0.033	1.013	0.114	0.059	0.318
	RF	0.137	0.180	0.111	1.306	0.417	0.257	0.473	0.047	0.043	0.030	1.031	0.214	0.102	0.283
	ANN	0.157	0.180	0.102	2.050	0.596	0.281	1.526	0.089	0.102	0.076	1.418	0.358	0.146	0.661
SSA	MLR	0.130	0.144	0.090	1.349	0.394	0.239	0.929	0.108	0.079	0.057	1.161	0.358	0.169	1.117
	MSVR	0.317	0.413	0.242	1.237	0.422	0.397	0.837	0.029	0.067	0.040	1.016	0.113	0.060	0.351
	RF	0.137	0.181	0.112	1.298	0.438	0.250	0.465	0.047	0.044	0.031	1.013	0.212	0.102	0.280
	ANN	0.171	0.209	0.163	4.207	0.464	0.257	1.077	0.120	0.100	0.087	2.289	0.380	0.190	0.546
Cubic	MLR	0.130	0.143	0.090	1.316	0.395	0.239	0.657	0.108	0.079	0.057	1.164	0.358	0.168	0.668
	MSVR	0.280	0.378	0.227	1.243	0.415	0.369	0.781	0.025	0.055	0.033	1.015	0.114	0.059	0.318
	RF	0.137	0.188	0.112	1.289	0.427	0.259	0.489	0.047	0.045	0.031	1.017	0.216	0.105	0.288
	ANN	0.162	0.232	0.125	2.905	0.599	0.337	1.171	0.103	0.094	0.062	6.119	0.416	0.146	0.523
Conv	MLR	0.126	0.146	0.090	1.457	0.397	0.241	0.762	0.103	0.078	0.057	1.162	0.355	0.168	0.637
	MSVR	0.298	0.395	0.234	1.242	0.417	0.383	0.811	0.026	0.058	0.035	1.068	0.113	0.060	0.331
	RF	0.139	0.211	0.124	1.303	0.432	0.265	0.480	0.049	0.045	0.032	1.013	0.219	0.106	0.274
	ANN	0.183	0.205	0.205	2.353	0.562	0.303	1.586	0.114	0.110	0.107	1.646	0.382	0.168	1.507
Centroids	MLR	0.136	0.164	0.108	1.356	0.420	0.260	0.652	0.097	0.075	0.052	1.213	0.346	0.163	0.892
	MSVR	0.327	0.424	0.247	1.236	0.424	0.408	0.849	0.030	0.069	0.042	0.974	0.113	0.061	0.356
	RF	0.137	0.181	0.109	1.295	0.424	0.261	0.498	0.047	0.043	0.030	1.021	0.210	0.105	0.285
	ANN	0.189	0.277	0.118	2.960	0.464	0.306	0.930	0.122	0.090	0.079	1.551	0.356	0.187	0.481
NW	MLR	0.130	0.149	0.094	1.322	0.411	0.238	0.672	0.114	0.084	0.063	1.194	0.377	0.184	0.619
	MSVR	0.293	0.383	0.228	1.247	0.419	0.375	0.796	0.022	0.048	0.030	0.929	0.130	0.069	0.293
	RF	0.140	0.207	0.129	1.304	0.431	0.275	0.483	0.048	0.044	0.032	1.026	0.219	0.106	0.285
	ANN	0.186	0.193	0.115	6.417	0.494	0.275	1.033	0.124	0.097	0.074	1.358	0.427	0.194	1.264
All	MLR	0.132	0.140	0.100	1.410	0.418	0.244	1.514	0.105	0.082	0.062	1.192	0.369	0.182	0.768
	MSVR	0.323	0.415	0.242	1.238	0.424	0.399	0.845	0.027	0.064	0.038	1.013	0.117	0.061	0.346
	RF	0.139	0.220	0.134	1.292	0.439	0.294	0.495	0.048	0.046	0.033	1.016	0.221	0.106	0.270
	ANN	0.208	0.251	0.233	5.489	0.511	0.323	1.063	0.145	0.110	0.108	2.916	0.359	0.176	2.007
PCA	MLR	0.133	0.159	0.110	1.272	0.422	0.242	4.674	0.115	0.091	0.068	1.234	0.383	0.189	0.692
	MSVR	0.321	0.412	0.241	1.238	0.423	0.397	0.841	0.027	0.063	0.037	1.030	0.118	0.061	0.345
	RF	0.185	0.236	0.155	1.298	0.453	0.311	0.603	0.062	0.053	0.038	1.022	0.225	0.113	0.299
	ANN	0.220	0.256	0.169	10.457	0.506	0.357	2.979	0.150	0.155	0.108	1.468	0.414	0.220	2.433
NPCA	MLR	0.133	0.159	0.110	1.272	0.422	0.242	4.674	0.115	0.091	0.068	1.234	0.383	0.189	0.692
	MSVR	0.321	0.412	0.241	1.238	0.423	0.397	0.841	0.027	0.063	0.037	1.030	0.118	0.061	0.345
	RF	0.184	0.233	0.153	1.300	0.452	0.298	0.610	0.063	0.054	0.040	1.018	0.219	0.110	0.296
	ANN	0.218	0.174	0.172	1.574	0.571	0.367	77.616	0.124	0.104	0.103	1.695	0.407	0.210	8.119

Table II: Forecasting errors measured as symmetric MAPE.

approach for predicting air quality in urban sensing systems sensors 2016," *Sensors*, vol. 16, no. 1, p. 86, 2016.

[13] M. J. Kane, N. Price, M. Scotch, and P. Rabinowitz, "Comparison of arima and random forest time series models for prediction of avian influenza h5n1 outbreaks," *BMC Bioinformatics*, vol. 15, no. 276, 2014.

[14] T. B. Trafalis and H. Ince, "Support vector machine for regression and applications to financial forecasting," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN2000)*, 2000, pp. 348–353.

[15] R. Navarrete and D. Viswanath. (2015) Support vector regression, smooth splines, and time series prediction. [Online]. Available: arXiv:1511.00158v1

[16] G. W. Taylor and G. E. Hinton, "Factored conditional restricted boltzmann machines for modeling motion style," *ICML '09 Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1025–1032, 2009.

[17] X. Qiu, Nanyang, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga, "Browse conference publications ¿ computational intelligence in ... help working with abstracts ensemble deep learning for regression and time series forecasting," in *Computational Intelligence in Ensemble Learning (CIEL), 2014 IEEE Symposium on*, 2014.

[18] E. Busseti, I. Osband, and S. Wong, "Compared kernalized regression and 3 types of nn using the data from kaggle competition global energy forecasting competition 2012 - load forecasting," Stanford University,

Tech. Rep., 2012.

[19] Y. Bao, T. Xiong, and Z. Hu, “Multi-step-ahead time series prediction using multiple-output support vector regression,” *Neurocomputing*, vol. 129, pp. 482–493, 2014.

[20] L. Zhang, W.-D. Zhou, P.-C. Chang, J.-W. Yang, and F.-Z. Li, “Iterated time series prediction with multiple support vector regression models,” *Neurocomputing*, vol. 99, no. 1, p. 411422, 2013.

[21] W. Hao and S. Yu, *Knowledge Enterprise: Intelligent Strategies in Product Design, Manufacturing, and Management. Proceedings of PROLAMAT 2006, IFIP TC5 International Conference, June 1517, 2006, Shanghai, China*, 2006, ch. Support Vector Regression for Financial Time Series Forecasting, pp. 825–830.

[22] A. Criminisi, J. Shotton, and E. Konukoglu, *Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning*, ser. Foundations and Trends in Computer Graphics and Vision, 2011, vol. 7, no. 2-3, ch. Regression Forests, pp. 131–148.

[23] F. Pérez-Cruz, G. Camps-Valls, E. Soria-Olivas, J. Pérez-Ruixo, A. Figueiras-Vidal, and A. Artés-Rodríguez, “Multi-dimensional function approximation and regression estimation,” *Artificial Neural Networks –ICANN*, pp. 796–796, 2002.

[24] A. Grover, A. Kapoor, and E. Horvitz, “A deep hybrid model for weather forecasting,” in *KDD ’15 Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 379–386.

[25] B. Zhu, “A novel multiscale ensemble carbon price prediction model integrating empirical mode decomposition, genetic algorithm and artificial neural network,” *Energies*, vol. 5, pp. 355–370, 2012.

[26] Y. Bai, Z. Chen, J. Xie, and C. Li, “Daily reservoir inflow forecasting using multiscale deep feature learning with hybrid models,” *Journal of Hydrology*, vol. 532, pp. 193–206, 2015.

[27] S. Ferrari, F. Bellocchio, V. Piuri, and N. A. Borghese, “Hierarchical approach for multiscale support vector regression,” *IEEE Transactions on Neural Networks Learning Systems*, vol. 23, no. 9, pp. 1448–1460, 2012.

[28] A. Motrenko and V. Strijov, “Extracting fundamental periods to segment human motion time series,” 2016.

[29] M. F. Matthias Scholz and J. Selbig., *Principal Manifolds for Data Visualization and Dimension Reduction*, ser. LNCSE. Springer Berlin Heidelberg, 2007, vol. 58, ch. Nonlinear principal component analysis: neural network models and applications, pp. 44–67.

[30] S. Yuksel, J. N. Wilson, and P. Gader, “Twenty years of mixture of experts,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1177–1193, 2015.

[31] A. Ignatov and V. Strijov, “Human activity recognition using quasiperiodic time series collected from a single triaxial accelerometer,” *Multimedia Tools and Applications*, pp. 1–14, 2015.

[32] R. J. Hyndman, “Another look at forecast-accuracy metrics for intermittent demand,” *Foresight*, no. 4, pp. 43–46, June 2006.

[33] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, *Emerging Intelligent Computing Technology and Applications*, 2012, ch. Time Series Forecasting Using Restricted Boltzmann Machine, pp. 17–22.

[34] [Online]. Available: http://hasc.jp/hc2010/HASC2010corpus/hasc2010corpus-en.html

[35] [Online]. Available: http://www.neural-forecasting-competition.com/NN3/datasets.htm