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.

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 [11], [12], [13], [14], [15], [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 [14] 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 [17]. 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 [18]. Finally, the multiple input multiple output (MIMO) strategy allows to obtain h prediction with at one step. The paper [17] 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) [11], [12], Support Vector Regression (SVR) [13], [14], [19] and artificial neural networks (ANN) [15], [16]. Random Forests combine decision trees with randomly generated nodes to increase the accuracy of classification or regression [20]. 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 [13], [19] reported high predictive performance of SVR applied to time series forecasting. In case of SVR, MIMO strategy is based on multivariate SVR [21]. Finally, artificial neural networks attract researches and practitioners from various domains [16], [22]. One of the reasons for that is the ability os 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 [22], [23] or use them in the multi-scale fashion we refer the reader to [9], [24], [5], [25], [26].

III. Problem statement

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

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

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 $\Delta t_{\mathrm{r}} < t_i \le T_{\max} + \Delta t_{\mathrm{r}}$, given the set $\mathfrak{D}$ (see Fig. 1a, 3). 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.

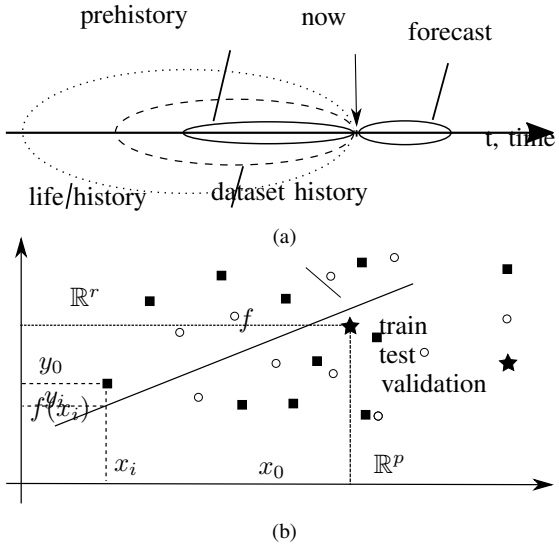


Figure 1: Forecasting (a) as regression problem (b).

A. Design matrix

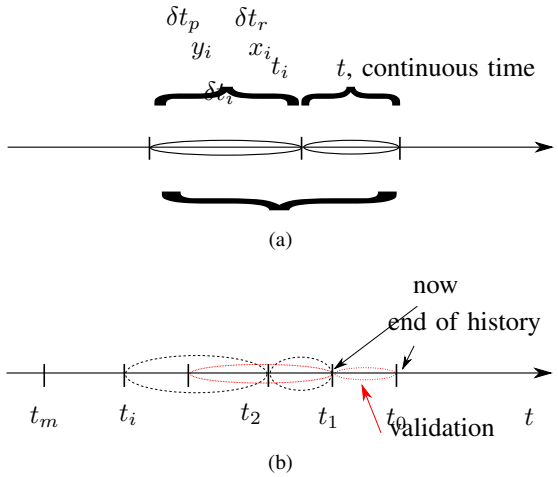


Figure 2: Draw an object from time series history.

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. 2a), which stands for the 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 (Fig. 2b). Let $\mathbf{s}_i^{(q)}$ denote the i-th segment of the time series $\mathbf{s}^{(q)}$

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

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

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

$$|t_{i+1} - t_i| > \Delta t_{\text{r}}. \tag{3}$$

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

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

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

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

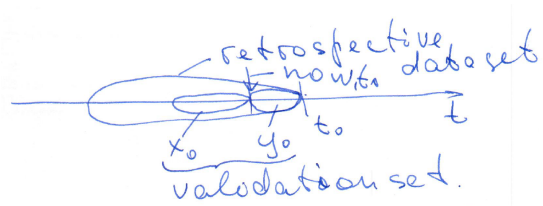


Figure 3: Retrospective forecast includes most recent samples in data set.

Now we are able the regression problem as follows:

$$\hat{\mathbf{y}} = f(\mathbf{x}, \hat{\mathbf{w}}), \;\; \hat{\mathbf{w}} = \arg\min_{\hat{\mathbf{w}}} S\big(\mathbf{w}|\mathbf{f}(\mathbf{w}, \mathbf{x}), \mathbf{y}\big). \tag{5}$$

Here the error function is given by $SMAPE$ (1) for each segment $[\mathbf{x}_i\mathbf{y}_i]$, averaged over all segments $i = 1, \ldots, m$ 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 $\boldsymbol{\phi} = [\boldsymbol{\phi}^{(1)}, \ldots, \boldsymbol{\phi}^{(Q)}]$, which corresponds to time series local histories $\mathbf{s} = [\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(Q)}]$, 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 $\boldsymbol{\phi}$:

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

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

These 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 with a period k and shift p, so that for $\mathbf{s} = [s_1, \ldots, s_T]$ the matrix

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

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

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

For the time series $\mathbf{s}^{(q)}$, $q = 1, \ldots, Q$ 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 $\mathbf{x}_i^{(q)}$, $q = 1, \ldots, Q$ compute k-means centroids $\mathbf{c}_p^{(q)}$, $p = 1, \ldots, P$. With the selected k-means distance function ρ construct the feature vector

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

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 $\mathbf{X}^*_{\text{train}}$, $\mathbf{X}^*_{\text{test}}$:

$$\mathbf{X}^* = \left[\begin{array}{c|c} \mathbf{X}_{\text{test}} & \mathbf{Y}_{\text{test}} \\ {\scriptstyle m_{\text{test}}\times n} & {\scriptstyle m_{\text{test}}\times r} \\ \mathbf{X}_{\text{train}} & \mathbf{Y}_{\text{train}} \\ {\scriptstyle m_{\text{train}}\times n} & {\scriptstyle m_{\text{train}}\times r} \end{array}\right]$$

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 [].

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 $\{1, \ldots, m_{\text{test}} - m_{\min}\}$ 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 $\mathbf{X}^*_{\text{test}}$

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

   - 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 $\mathbf{X}^*_{\text{train},i}$
   - 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

The goal of the experiment is to study the performance of multivariate regression approach to the problem of time series forecasting.

This section presents the results of computational comparison of the forecasting models presented above. To test each model, we used several datasets, described below.

Table I: Regression models.

Model name	Hyper parameters
multivariate linear regression (MVR) with l_2-regularization	regularization coefficient
support vector regression (SVR)	parameters C, λ, ε
artificial neural network (ANN)	number of hidden layers, hidden layers size
random forest (RF)	number of trees, number of variables per tree

A. Datasets

- Energy-Weather dataset. The dataset consists of the Polish electricity load time series [27] 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 Energy-Weather dataset was used to generate several artificial datasets.
- Accelerometry dataset. This dataset consists of accelerometry time series from the Human Activity Sensing Consortium [28]. Each time series in the datasets is a sequence of acceleration records. The time series were recorded for 120 seconds while a subject performed a sequence of activities: stay, walk, jog, skip, stair up or stair down. The sampling rate varies between 10 and 100Hz, but stays constant for each time series.
- Financial dataset [29]. The dataset A (final, complete) from the 2006/07 Forecasting Competition for Neural Networks & Computational Intelligence. The dataset contains 111 monthly time series drawn from homogeneous population of empirical business time series.

B. Experimental results

Each time series from the datasets was converted to the design matrix (4) and used as input data to train regression models according to (5).

Table II lists forecasting errors for the proposed feature generation strategies applied to the test time series from the HASC dataset.

Fig. 4 displays an example of target time series HASC1001 and the results of forecasting a single segment of this time series with the baseline models.

Feature set	Models	MAPE test	MAPE train	AIC
History	VAR	7.215	0.067	1427.927
	SVR	0.718	0.320	1396.463
	Random Forest	0.463	0.206	1388.668
	Neural network	0.624	0.407	1389.319
SSA	VAR	0.755	0.402	1375.959
	SVR	1.348	0.434	1401.053
	Random Forest	0.746	0.335	1413.181
	Neural network	0.544	0.387	1379.299
Cubic	VAR	0.562	0.382	1371.387
	SVR	0.779	0.418	1373.299
	Random Forest	0.647	0.320	1412.390
	Neural network	0.795	0.387	1371.354
Conv	VAR	2.182	2.180	2252.866
	SVR	2.986	0.537	1413.103
	Random Forest	0.583	0.323	1408.932
	Neural network	0.743	0.406	1373.955
NW	VAR	2.021	0.041	1605.095
	SVR	0.714	0.323	1398.156
	Random Forest	0.427	0.195	1381.996
	Neural network	0.815	0.425	1389.418
All	VAR	2.000	2.000	9910.259
	SVR	0.751	0.324	1395.951
	Random Forest	0.415	0.194	1381.906
	Neural network	1.254	0.399	1381.405
PCA	VAR	0.712	0.310	1388.756
	SVR	0.748	0.323	1395.828
	Random Forest	0.504	0.214	1378.500
	Neural network	0.563	0.392	1377.608

Table II: Forecasting errors measured as symmetric MAPE.

Fig. 6 illustrates the results of regression built on the design matrix, augmented with the particular data generation method.

Fig. 7 demonstrates the forecasts, obtained by each model in the proposed framework. Here the design matrix was augmented with the generated features and PCA was applied to select a subset of features.

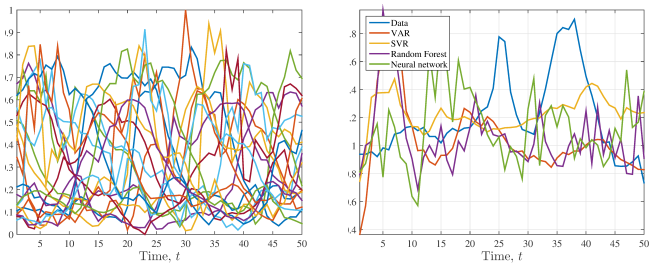
REFERENCES

[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.



(a) (b)

Figure 4: Accelerometry time series from HASC project. (a) Target segments of the time series HASC1001. (b) Forecasting results for HASC1001 (VAR, SVR, Random Forest, Neural network).

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

[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] R. Yu, Y. Yang, L. Yang, G. Han, and O. A. Move, "Raqa random forest approach for predicting air quality in urban sensing systems sensors 2016," *Sensors*, vol. 16, no. 1, p. 86, 2016.

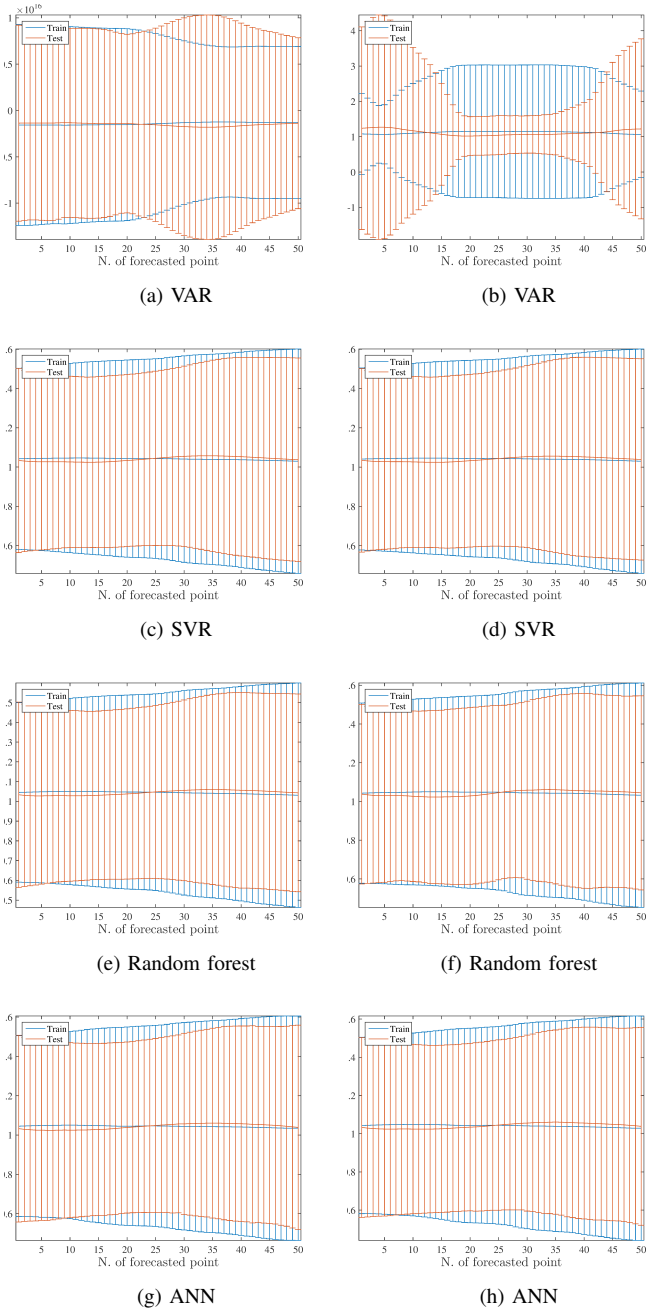
[12] M. J. Kane, N. Price, M. Scotch, and P. Rabinowitz, "Comparison of arima and random forest time series models for prediction of avian influenza h5n1 outbreaks," *BMC Bioinformatics*, vol. 15, no. 276, 2014.

[13] 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.

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

[15] 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.

[16] G. W. Taylor and G. E. Hinton, "Factored conditional restricted boltzmann machines for modeling motion style," *ICML '09 Proceedings of*



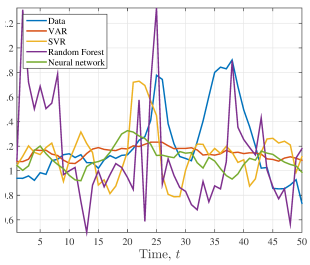
(a) VAR (b) VAR

(c) SVR (d) SVR

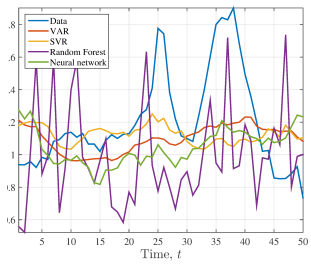
(e) Random forest (f) Random forest

(g) ANN (h) ANN

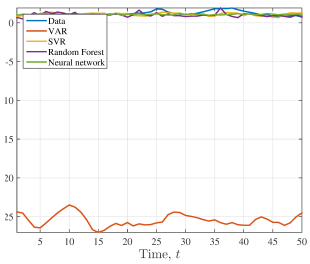
Figure 5: Mean residual by number of forecasted point for historical data only and all generation strategies with PCA.



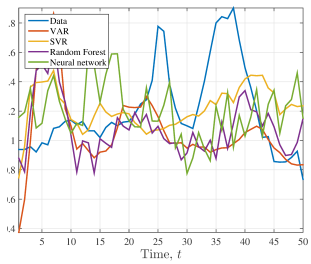
(a) SSA



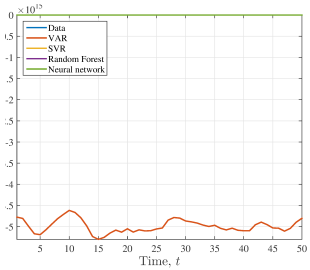
(b) Cubic



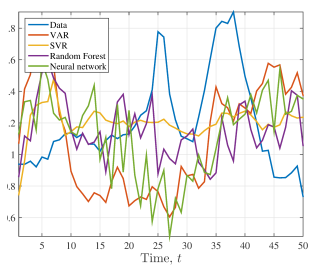
(c) Conv



(d) NW

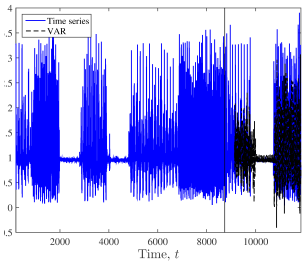


(e) All

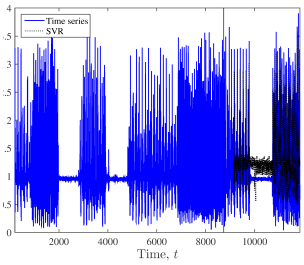


(f) PCA

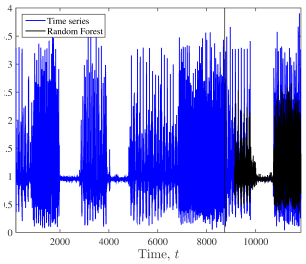
Figure 6: Forecasting results for various feature generation strategies, HASC1001.



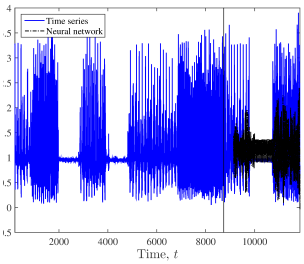
(a) VAR



(b) SVR



(c) Random Forest



(d) ANN

Figure 7: Forecasting results for HASC1001 with all feature generation strategies applied and PCA feature selection.

the 26th Annual International Conference on Machine Learning, pp. 1025–1032, 2009.

[17] 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.

[18] 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.

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

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

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

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

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

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

[31] 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.