# GFSM: a Feature Selection Method for Improving Time Series Forecasting

Youssef Hmamouche*, Piotr Przymus†, Alain Casali‡ and Lotfi Lakhal§
LIF - CNRS UMR 7279,
Aix Marseille Université, Marseille, France
Emails: *youssef.hmamouche@lif.univ-mrs.fr, †piotr.przymus@lif.univ-mrs.fr,
‡alain.casali@lif.univ-mrs.fr, § lotfi.lakhal@lif.univ-mrs.fr

***Abstract*—Handling time series forecasting with many predictors is a popular topic in the era of ”Big data”, where wast amounts of observed variables are stored and used in analytic processes. Classical prediction models face some limitations when applied to large-scale data. Using all the existing predictors increases the computational time and does not necessarily improve the forecast accuracy. The challenge is to extract the most relevant predictors contributing to the forecast of each target time series. We propose a causal-feature selection algorithm specific to multiple time series forecasting based on a clustering approach. Experiments are conducted on US and Australia macroeconomic datasets using different prediction models. We compare our method to some widely used dimension reduction and feature selection methods including principal component analysis PCA, Kernel PCA and factor analysis. The proposed algorithm improves the forecast accuracy compared to the evaluated methods on the tested datasets.**

***Keywords–Time Series Forecasting; Feature Selection; Multivariate prediction models; Artificial Neural Networks.***

## I. Introduction

Time series analysis and data mining incorporates a set of tools, methods, and models for describing the evolution of data over time. Such tools play important role in business analysis and business intelligence systems where they generate new, valuable information by combining trends, forecasts, correlations, causalities *etc.* This additional information can be then used to improve the decision-making process and contribute to more intelligent and efficient decisions.

This article is an extended version of research from [1]. Compared to the previous version, we have discussed possible extensions to the algorithm and significantly expanded the experimental section.

In summary, we try to improve the forecasting of a time series using multivariate models, by selecting only the most relevant varaiables. This leads us to the problem of hidden common factors that may cause multiple variables. To overcome this problem, we propose a feature selection algorithm based on the Granger causality graph. Each time series is represented as a node in the graph and the causality is expressed as edges weights. We follow the classical notion of Granger predictive causality presented in [2], in order to compute the dependencies between each two variables.

One of the first successful univariate time series forecasting models was the Auto-Regressive model [3]. Various versions of this model are still in use today. They are based on the same principle. That is, they take into account historic observations in order to predict the future of variable. Univariate models are limited only to one source of information, and thus, they cannot utilize potentially-exploitable time series. To do better, researchers began to introduce multivariate time series analysis and forecasting models capable of exploiting multiple time series [4]. Many of those concepts are still present in forecasting tools nowadays.

Multivariate analysis increases the complexity of models compared to univariate ones, as multivariate models describe the forecasted time series based on $(i)$ its historical observations and $(ii)$ the historical observations of other series in the dataset. On the plus side, utilizing relevant information from other variables [5] may improve the resulting forecast.

Building a model using all existing variables is usually not an viable option as it may add to much noise to fit an accurate model. For example, in [6], [7], based on two macroeconomic datasets, it was found that using more than $30\% - 60\%$ of the existing predictors does not improve forecast quality and in fact may worsen the results. This rises question how to select relevant variables, that was already investigated in several works [5]–[8].

We can distinguish two popular approaches. One is to enforce the model to discard irrelevant information either by shrinking the coefficients or eliminating them. Shrinkage models, Lasso, Lars or regressors based on neural networks are examples of this approach [6], [7].

The second approach is to use a two step model, where (i) a separate procedure is used to extracts relevant information from multiple variables, and then the selected variables are used to build a multivariate forecasting model. The extraction may be done using feature selection, dimension reduction or by using the notion of causality [1], [5], [8]. In the second step, any multivariate forecasting model can be used (including the ones from previous paragraph).

Our proposed approach design is motivated by industrial needs, where precise forecasts are requested in the presence of huge amount of observed variables. The primary goal was to provide a forecast horizon for a set of variables, which allows to do an educated guess when to buy or sell products. The secondary goal was to detect the frauds in public markets. We mainly work on the prices of raw materials and/or finished products available on public markets.

This paper is organized as follows. First, we discuss the related works (Section II) and prediction models (Section III). Next, we talk about causality measures (Section IV). Then, we discuss our approach in Section V and evaluate its performance in Sections VI and VII. Finally, in Section IX, we summarize our contributions and possible future research.

## II. Related work

Using all the existing variables in a multivariate model has some drawbacks. First for Auto-Regressive based models, if the number of regressors is proportional to the sample size, the ordinary least squares (OLS) forecasts may not be efficient [5]. Secondly, the most accurate forecasts are generally obtained using smaller number of predictors [7], [6]. Thus, in this section we discuss works that deal with the problem of large number of predictors.

Feature selection refers to the act of extracting a subset of the most relevant variables (features) of size $k$ from a set of variables of size $n >> k$. Dimension reduction methods generate a new features with lower dimension from the original features by transforming them. Both of them can be used to optimize the inputs of prediction models. If additionally we are interested with descriptive analysis, feature selection techniques give more information as they select existing variables.

The Principal Component Analysis (PCA) is one of the most common dimension reduction methods used. Based on a set of variables, this method takes advantage of the inter-correlation between them [9]. The idea is to generate the principal linearly uncorrelated variables that describe as much as possible the original variables. The Kernel PCA method is a non-linear version of PCA, that extends it by considering the non-linear relationships between variables using kernel techniques [10]. Factor analysis (FACT) is another technique, similar to PCA in the sense that it generates uncorrelated factors of the original variables, additionally it fits a model of error terms associated with factors [9]. Both PCA and FACT are used to construct the dynamic factor model [11], [12], [5], [13]. It is a prediction model that is designed for high-dimensional time series, or time series where the number of observations exceeds the number of variables. The idea is to find a small number of hidden factors (dynamic factors), that drive all of the observed variables. Thus, each variable can be constructed as a combination of those factors. The observed variables forecasts are constructed based on forecasts of dynamic factors.

Another approach for dealing with many predictors is based on the idea of shrinking the coefficients of irrelevant variables towards (or exactly to) 0. This can be achieved by fitting the regression model with constraints on coefficients. There are numerous well known shrinkage/regularization methods, for instance the Lasso [14] and Ridge [15] methods. While they are associated with the problem of multiple regressions, they can be easily adapted to address the problem of forecasting [16], [17], [6], [18], [7].

Artificial neural networks are also a common choice for solving this problem. In [19], the authors propose an automatic approach for stock market forecasting and trend analysis. A pre-processing step was applied using the PCA in order to transform the data into a set of uncorrelated variables and to reduce the dimension of the input variables. Then, an artificial neural network was used for forecasting the stock outputs, and finally a neuro-fuzzy system is used to analyse the forecasts trends. Similarly, in [8], a two step data mining process was proposed for forecasting daily stock market, using PCA as a first step to reduce the dimension of predictor variables, then, a feed-forward neural network is trained for prediction. In [20], when forecasting time series that represent series of brain images, with a number of variables larger than the number of observations, the authors propose a feature selection method based on PCA, Recursive Feature Elimination and Support Vector Machines. In [21], a two-step forecasting approach was presented to forecast two years of Australian electricity load time series. First, correlation, Mutual Information and instance-based feature selection methods are applied in order to extract the relevant informative lag variables. And second, a multivariate artificial neural network and statistical models are applied to make forecasts.

## III. Prediction Models

In this section, we present a description of prediction models used in this research. Two types of models are discussed, statistical and artificial neural network models.

### A. Statistical models

Many prediction models are based on the same principle as the Auto-Regressive model $\text{AR}(p)$ [22]. The idea is to expresses an univariate time series as a linear function of its $p$ precedent values:

$$y_t = \alpha_0 + \alpha_1 y_{t-1} + \cdots + \alpha_p y_{t-p} + \epsilon_t,$$

where $p$ is the order of the model, $\alpha_0 \ldots \alpha_p$ are the parameters of the model, and $\epsilon_t$ is a white noise error term.

The Moving Average model of order $q$ ($\text{MA}(q)$) has the same expression but for the error terms:

$$y_t = \alpha_0 + \alpha_1 \epsilon_{t-1} + \cdots + \alpha_p \epsilon_{t-q} + \epsilon_t \tag{1}$$

The $\text{ARMA}(p,q)$ model [22] expresses errors terms and past values of the time series in the same model. It can be expressed as follows:

$$y_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{i=1}^{q} \beta_i \epsilon_{t-i} + \epsilon_t.$$

For non-stationary time series, the $\text{ARIMA}(p,d,q)$ model [22] is more preferable; it applies the $\text{ARMA}(p,q)$ model after a differencing step, in order to obtain stationary time series, where $d$ is the order of differencing (computing $d$ times the differences between consecutive observations).

In [23], the Vector Auto-Regressive (VAR) model was introduced as an extension of the AR model. Consider a $k$-dimensional time series $Y_t$, the $\text{VAR}(p)$ system expresses each univariate variable of the multivariate time series $Y_t$ as a linear function of the $p$ previous values of itself and the $p$ previous values of the other variables:

$$Y_t \quad = \alpha_0 + \textstyle\sum_{i=1}^{p} A_i Y_{t-i} + \epsilon_t,$$

where $\epsilon_t$ is a white noise with a mean of zero, and $A_1, \ldots, A_p$ are $(k \times k)$ matrix parameters of the model.

In [24], the Vector Error Correction (VECM) was introduced. This model transforms the VAR model by taking into account non-stationarity of the time series and by including cointegration equations. To simplify matters, let us consider two multivariate time series ($Y_t$) integrated of order one, which means all the variables of $Y_t$ are $I(0)$ or$I(1)$, stationary or integrated of order 1. The VECM Model can be written as follows:

$$\Delta Y_t \quad = \quad \Pi Y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta Y_{t-i} + u_t,$$

where $\Pi$ is the matrix representing the co-integration equations, which can be generated by VAR model, and $\Gamma_i$ are the matrix parameters of the VECM model. If $rk(\Pi) = 0$, then there are no cointegration equations, so that the VECM model is reduced to the VAR form applied after differencing time series.

### *B. Artificial Neural Network Model*

ANNS are increasingly popular for forecasting high-dimensional time series. The relation between target and predictors is encoded in a network of neurons characterized generally by a non-linear function. The ANNS are usually able to model time series more dynamically compared to classical models.

The principle of using ANNS in time series forecasting is simple. It consists of transforming the data into a supervised learning problem, where the inputs represent the lagged values of the predictors and the target variables. Then, the network is initialized, commonly with randomly generated parameters. Finally, the network is trained based on its inner objective function (that depends on the network structure, activation function, and other learning parameters). Formally, given a k-dimensional time series $(y_1, \ldots, y_k)$ and a network function $f_{nn}$, the resulting time series can be expressed as:

$$y_t = f_{nn}(y_{t-1}, \ldots, y_{t-p}, \ldots, y_{k(t-1)}, \ldots, y_{k(t-p)}) + \epsilon_t.$$

During the training step, the parameters of the network are updated through the back-propagation step, using an optimization algorithm, such as Gradient Descent or Stochastic Gradient Descent, which aim to find a local minimum of the error function $\epsilon_t$ using some criteria. Popular criteria choices are the mean squared error ($mse$) or the mean absolute error ($mae$). Note that other metaheuristics can be used to try to find the global optimum of the error function, such as Genetic Algorithms and Particle Swarm Optimization [25], [26].

In [27], the vector autoregressive neural network (VAR-NN) model is investigated, as an extension of the classical VAR process, based on a multi-layer perceptron. In [28], a comparison between the VAR model and a multi-layered feed-forward neural network has been presented for forecasting macroeconomic variables. It was shown that the neural network has a superior forecasts results than the VAR model in this particular case. In [29], the VAR-NN model shows good performance compared with the VARMA model in the task of predicting non linear functions.

The main drawback of multi-layered perceptron neural network is that the neurons are not able to remember past information. Because each neuron provides an output based directly on the activation function and the input values

$$h_t = f(Wx_t + b).$$

Yet, in time series (and sequences in general), maintaining information inside the network may improve the performance of the model. Recurrent Neural Networks (RNNs) were designed to handle this issue. Their structures allow hidden layers to be self-connected, in a way that output may depend on the current input and its previous state. Mathematically, the function of RNNs neurons can be expressed as follows:

$$h_t = f(Wx_t + Uh_{t-1} + b).$$

Unfortunately, the maintained memory in the network is short, because the current state of network depends only on the previous one. The Long Short Term Memory network introduced in [30] was consequently designed to address this problem. The authors propose a specific structure, by adding extra options transforming traditional neurons into blocks, able to model the mechanics necessary for the network to forget and remember informations. Thus, it can learns how to use long-term information passed through the network in the working memory. The mathematical formulation of a hidden LSTM block can be written as follow:

$$\begin{aligned}
&f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f),\\
&i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i),\\
&c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c x_t + U_c h_{t-1} + c_0),\\
&o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o),\\
&h_t = o_t \odot tang(c_t),
\end{aligned}$$

where $\odot$ represents the element-wise multiplication $x_t$ is the input vector, $(W, U, b)$ are the parameters of the model, $\sigma$ and $tanh$ are the sigmoid and tangent activation functions, $f_t$ is the forget gate layer responsible for updating the weight of remembering the previous information, $i_t$ is the input gate layer responsible for updating new information, $c_t$ is the cell state at time $t$, $o_t$ is the output gate layer, and $h_t$ is to output of the cell.

## IV. CAUSALITY MEASURES

Studying the relationships between time series is an important task for multivariate time series analysis, which can be exploited for forecasting. The common measures like Correlation and Mutual Information, are symmetric, so they do not provide enough information about the dependencies between variables, i.e., which variables influence the other. The goal of causality is to detect the impact of one time series on an another one in terms of prediction.

In this section, we discuss two different causality measures, the Granger causality [2], and the Transfer entropy [31]. Let us consider two univariate time series $x_t$, $y_t$. The Granger definition of causality acknowledges the fact that $x_t$ causes $y_t$ if it contains information helpful to predict $y_t$. In other words, $x_t$ causes $y_t$ if a prediction model that uses both $x_t$ and $y_t$ performs better than the one that is based merely on $y_t$.

We detail here the standard Granger causality test [2], which uses the VAR model with a trend term. The test compares two models. The first one only takes into account the precedent values of $y_t$, and the second uses both $x_t$ and $y_t$ in order to predict $y_t$. If there is a significant difference between the two models, then it can be ascertained that the added variable $(x_t)$ causes $y_t$:

$$\begin{aligned}
\text{Model}_1: \quad y_t &= \alpha_0 + \alpha t + \sum_{i=1}^{p} \alpha_i y_{t-i} + \epsilon_t.\\
\text{Model}_2: \quad y_t &= \alpha_0 + \alpha t + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{i=1}^{p} \beta_i x_{t-i} + \epsilon_t.
\end{aligned}$$

The next step of the test is to compare the residual sum of squares (RSS) of these models using the Fisher test. The statistic of the test is expressed as follow:

$$F = \frac{(\text{RSS}_1 - \text{RSS}_2)/p)}{(\text{RSS}_2/n - 2p - 1)},$$

where $\text{RSS}_1$ and $\text{RSS}_2$ are the residual sum of squares related to $\text{Model}_1$ and $\text{Model}_2$ respectively, $n$ is the size of the predicted vector. Two hypotheses are tested,

- $H_0$: $\forall i \in \{1, \dots, p\}, \beta_i = 0$,
- $H_1$: $\exists i \in \{1, \dots, p\}$, $\beta_i \neq 0$.

Under the null hypothesis $H_0$ (the hypothesis that $x$ does not cause $y$), $F$ follows the Fisher distribution with $(p, n-2p-1)$ degrees of freedom. Therefore, the Granger causality test is carried out at a level $\alpha$ in order to examine the null hypothesis. The p-value of the test is the probability to observe the given result under the assumption that $H_0$ is true. In our case, we consider the causality as one minus p-value in order to express values of causalities in the range $[0, 1]$.

Transfer Entropy is another causality measure based on information theory. Before discussing this measure, it is worth to recall the notion of mutual information between two processes (or two univariate time series) $I$ and $J$. It measures the mutual dependencies (symmetric) between the two processes. Consider two processes $I$ and $J$, with probability distribution $p(i)$ and $p(j)$, joint probability $p(i, j)$, and conditional probability $p(i|j)$. The mutual information between the two processes $I$ and $J$ can be expressed using the Kullback entropy as follows:

$$M_{IJ} = \sum_{i \in I, j \in J} p(i,j) log(\frac{p(i|j)}{p(i)p(j)}).$$

The main drawback of this measure is that is symmetric and does not model the transfer of information from one process to another. The transfer entropy was proposed as an extension of mutual information in [31]. And the idea is to add a time shift parameter, that allows modeling the non symmetric transfer of information between processes. Even though Granger causality and Transfer entropy may seem to be based on different concepts, an interesting finding was presented in [32], showing that they are equivalent for variables with normal distributions. The mathematical formulation of transfer entropy from $J$ to the $I$ can be expressed as follows:

$$T_{J \to I} = \sum_{i \in I, j \in J} p(i_{n+1}, i_n^k, j_n^l) log(\frac{p(i_{n+1} \mid i_n^k, j_n^l)}{p(i_{n+1} \mid i_n^k)}.$$

## V. Our proposal

Our approach focuses on the selection of the top predictor variables by describing their hidden dependencies using bivariate causality. Let us consider $Y = \{y_1, y_2, \dots, y_n\}$ a multivariate time series and a target variable $y$. The goal is to select a subset of $Y$, for which we have the more accurate forecasts of $y$.

We assume that causality is more important than correlation measures when forecasting time series. Because, in contrast to correlation, causality models the non-symmetric dependencies between variables. In other words, if two variables are correlated, it does not identify which variable has an impact on the other. By considering this hypothesis, one solution is to choose a set of variables having strong causality regarding to the target $y$. This is equivalent to a univariate feature selection technique that ranks variables based on causality to the target. This approach was already investigated in [33] and [34] using the Granger causality test. The main limitation of those approaches is that they potentially ignore hidden relationship between predictor variables. That is, the same hidden source of information may be exploited despite the fact of using with multiple selected variables.

Let us underline that from a theoretical point of view, there are $\sum_{i=1}^{k} \binom{n}{i}$ possible partitions of size less than or equal to $k$. Where $n$ is the number of original features, and $k < n$. In the general case, i.e., without fixing the maximum number of predictors $k$, there are $2^n$ possible partitions, so $2^n$ possible models [5]. In addition, the causality is generally not a monotone function. As a consequence, finding the best subset of variables that maximizes the multivariate causality is a NP-hard problem.

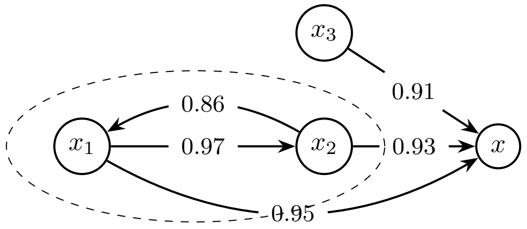


Figure 1. Illustration Of Dependencies Between Time Series Using the Granger Causality Graph.

In Figure 1, we show a small Granger causality graph describing dependencies between 4 variables. Let us try to select two variables as predictors for the target variable $x$. Selecting variables by ranking them according to causality leads to getting $x_1$ and $x_2$ as predictors. However, $x_1$ and $x_2$ might provide the same information because $x_1$ causes $x_2$.

We propose a new method to deal with this problem based on clustering the causality graph or the adjacency matrix using a clustering technique, such as the Partitioning Around Medoids or the hierarchical clustering.

### A. *The proposed algorithm*

The proposed method can be divided into four steps:

- Build the adjacency matrix of causalities between variables.
- Eliminate variables having low causality on the target.
- Cluster the set of the remaining predictors variables, by minimizing the causalities between clusters, and maximizing the causality within clusters.
- Choose one element from each cluster, the one that maximizes the causality on the target variable.

Let us underline that this algorithm is a generalization of ranking methods, in the way that if the number of clusters (input of the method) is equal of the number of variables, then the algorithm will select the first $k$ variables by ranking them according to the causality to the target. In Figure 2, the GFSM (causality-Graph based Feature Selection Method) algorithm summarizes our approach.

```
Input: Set of predictors time series $Y = \{y_1, y_2, \ldots, y_n\}$,
    $y$ the target variable, MinCaus Min-Causality threshold,
    $k$ the selection size
Output: Gfsm-Cl the set of the selected variables associated to $y$
    for $i = 1$ to $n$ do
        if $causality(y_i \rightarrow y) \leq$ MinCaus then
            $Y = Y \setminus \{y_i\}$
        end if
        if $Y.size() \leqslant k$ then
            return Gfsm-Cl $= Y$
        end if
    end for
    /* Build the matrix of causalities.                 */
    Let $MC$ be the adjacency matrix of causalities
    for each $y_i$, $y_j$ in $Y$ such that $i \neq j$ do
        $MC[i,j] = MC[j,i] = 1 - max(causality(y_i \rightarrow y_j), causality(y_j \rightarrow y_i))$
    end for
    /* The clustering step.                             */
    $Clusters = clustering(MC, k)$
    for each Cluster $cl$ in $Clusters$ do
        Gfsm-Cl = Gfsm-Cl $\cup \arg\max_{cl_j \in cl} (causality(cl_j \rightarrow y)$
    end for
    return Gfsm-Cl
```

Figure 2. The GFSM Algorithm.

## B. Scalability of the proposed algorithm

Scalability of the algorithm is crucial when handling high-dimensional time series, thus, in this part we discuss it. Granger causality is computed by comparing forecasting accuracy between univariate model and bivariate model (usually followed by significance test). This requires that two forecasting models have to be constructed to compute the Granger causality. An univariate AR model on target variable and a bivariate VAR model with a target variable and one additional predictor.

The algorithm presented uses the Granger causality graph. It requires computation of a set of tuples $G = (P \times P) \setminus \Delta$, where $P$ is the set of predictors and $\Delta = (p_1, p_2) \in P \times P : p1 = p2$. Therefore, we have to compute univariate models for all elements in $P$ and bivariate models for tuples in $G$. This is trivially scalable as all models can be computed independently. Finally, the results should be grouped by target variable and simple statistical tests of accuracy computed for univariate model on variables $p \in P$, and a bivariate model $(p_1, p_2) \in G$ (simple task in map reduce approach). As a result, we compute the adjacency matrix.

For financial time series it is reasonable to assume that the resulting matrix will have reasonable size and will feet single computation node. In rare cases, when the matrix is very large, scalable clustering algorithms could be used (like k-medoids and other methods investigated in [35] and [36]).

## C. Example

Consider $Y = \{y_1, \ldots, y_8\}$ a set of predictors and a target variable $y_9$. Let us apply the GFSM algorithm in order to select 4 predictor variables from $Y$ that will contribute to forecast $y$.

*1) The matrix of causalities:* First, the algorithm computes the Granger causalities between variables in pairs. In this example, we take the matrix of causalities of a dataset containing nine variables:

$$MC = \begin{bmatrix} 1.00 & 0.935 & 0.999 & 0.999 & 0.832 & 0.998 & 0.998 & 0.933 & 0.998 \\ 0.28 & 1.00 & 0.877 & 0.87 & 0.224 & 0.785 & 0.801 & 0.999 & 0.868 \\ 0.033 & 0.656 & 1.00 & 0.106 & 0.479 & 0.944 & 0.775 & 0.082 & 0.905 \\ 0.028 & 0.647 & 0.239 & 1.00 & 0.483 & 0.944 & 0.776 & 0.096 & 0.905 \\ 0.7 & 0.457 & 0.977 & 0.978 & 1.00 & 0.343 & 0.031 & 0.398 & 0.901 \\ 0.808 & 0.417 & 0.818 & 0.817 & 0.906 & 1.00 & 0.997 & 0.431 & 0.722 \\ 0.274 & 0.742 & 0.992 & 0.992 & 0.942 & 0.959 & 1.00 & 0.906 & 0.788 \\ 0.327 & 0.999 & 0.998 & 0.998 & 0.427 & 0.895 & 0.996 & 1.00 & 0.900 \\ 0.304 & 0.071 & 0.581 & 0.584 & 0.205 & 0.448 & 0.999 & 0.754 & 1.00 \end{bmatrix}$$

*2) Clustering and selecting the variables:* The algorithm partitions the variables based on the symmetrical matrix (as mentioned in the algorithm 2). The idea behind symmetrizing the matrix of causalities is to be able to perform the clustering task. By using the Partitionning Arround Medoids (PAM) [37], let us also underline that this method partitions elements from a symmetric dissimilarity matrix and minimizes dissimilarities within clusters. In our case, the algorithm maximizes causalities within clusters. That is why we use 1 minus the causality matrix as an input of the PAM method.

Then, the algorithm chooses from each cluster the element that has maximal causality on the target. The obtained clustering vector associated to $\{y_1, \ldots, y_8\}$ is $(1, 2, 1, 1, 3, 1, 4, 2)$. And based on the causalities to the target (last column of the adjacency matrix), the selected variables are $\{y_1, y_5, y_7, y_8\}$.

*3) Evaluating the clusters:* The quality of the causalities founded depends on, first the type of the data, and second, on the evaluation of the clustering task. In our case, we evaluate the quality of the clusters using the following objective function:

$$\text{minimize } G(x) \quad = \sum_i^n \sum_j^n (1 - max(c_{ij}, c_{ji})) \times z_{ij},$$

where

1) $z_{ij} = \begin{cases} 1 & \text{if } y_i, y_j \text{ belong to the same cluster} \\ 0 & \text{otherwise.} \end{cases}$
2) $c_{ij}$ = causality$(y_i \rightarrow y_j)$.

This evaluation can be used in general as measure of causal relationships in multivariate time series. In the example, the value of $G$ is 0.000168.

## VI. EXPERIMENTAL EVALUATION

In this section, we describe the experiments design, the datasets used, the forecasting methodology, and the methods and models implemented.

## A. Datasets

The experiments are performed on macroeconomic datasets of US and Australia.

- The US macroeconomic dataset [6]: quarterly numeric time series, containing 143 features and 200 observations, spanning the period 1960 – 2008.
- The Australian macroeconomic dataset [7]: quarterly numeric time series spanning the period 1984 – 2015, comprising 117 variables and 119 observations.

These datasets were used in the context of forecasting with many predictors in [6], [7]. Different models were evaluated using those datasets, the naive-benchmark and the AR (4) as baseline models, the dynamic factor model, the VAR model and other shrinkage methods. Both datasets contain three main series, that we focus on in the experiments:

- US dataset: GDP: Real gross domestic product, CPI (Cpi all items (sa) fred), Fedfuns (Interest rate: federal funds (effective)).
- Australia dataset: GDP: Real gross domestic product, CPI (Consumer Price Index), IBR (overnight interbank rate).

### *B. Methods: feature selection and dimension reduction techniques*

We use the Scikit-learn machine learning module [38] for the implementation of the following methods:

- Principal component analysis (PCA) [39].
- Kernel Principal component analysis (Kernel PCA) [10].
- Factor Analysis (FACT) [9].

And we implemented the following two methods:

- A univariate feature selection method using Granger causality (UFSM), similar to the one proposed in [33].
- Our proposal, by generating two versions using two clustering methods in the Algorithm 2. The first one is based on the Parttionning Arround Medoids (PAM) [37] (pGFSM), and the second (hGFSM) is based on the hieearchical clustering [40].

### *C. Methods: Prediction models*

The used prediction models can be classified into three types; baseline model, statistical models, and artificial neural networks (ANNS):

- Baseline model: we use the naive-benchmark (4) model, that predict the next value of a variable based on the mean of the last $4$ values,
- Statistical models: we use the AR(4) model and the ARIMA $(4, d, q)$ models (with automatic determination of the parameters $d$ and $q$, see Section III for details.) to analyse forecasts without the use of predictors. And the Vector Error Correction Model (see Section III for mathematical formulation). We use the implementation from the **forecast R** package [41].
- ANNS models: we use the following strategy to build ANNS models. We transform the data into a supervised learning problem, based on the lag parameter, the target variable, and the selected predictors. Then, we adapt two existing ANNs models to our problem. We use the multilayer perceptron and the Long Short Term Memory networks from the deep learning **python** library; **keras** [42].
  For the model based the MLP structure (VARMLP), we use one hidden layer using a simple rule of thump to determine the number of hidden neurons, $2/3\times(n+1)$, where $n$ is the number of inputs. And for the LSTM based model, we use the same number of units in hidden layer as the number of input variables. In both cases, the models inputs depend on the lag parameter $p$ and the number of predictors $k$, $n = k \times p$. Then, the networks are trained using back-propagation through time, and the error functions are minimized using the stochastic gradient decent algorithm.

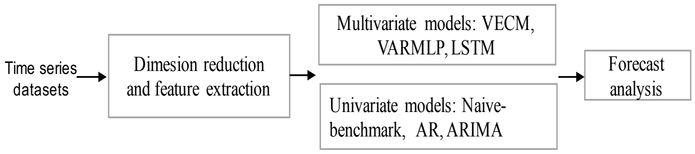

Time series datasets
Dimesion reduction and feature extraction
Multivariate models: VECM, VARMLP, LSTM
Univariate models: Naive-benchmark, AR, ARIMA
Forecast analysis

Figure 3. The Used Forecasting Process.

### *D. Forecasting procedure*

We implemented an automatic step-wise forecasting process, as seen in Figure 3. First, It reduces the number of predictors using feature selection and dimension reduction techniques. Second, it applies the forecasting models. And finally, it evaluates the quality of the obtained forecasts. We note that the reduction step is not required for univariate models, because they consider just the target variable to make forecasts. The pre-processing step consists in transforming time series to stationary via differencing, and this step is not required for all models such VECM and ARIMA as they take it into account automatically. For the proposed algorithm, we use the Granger causality test as causality measure. The lag parameter for Granger causality test is automatically determined using the Akaikes Information Criterion (AIC) [43] with a maximum value of $4$ equivalent to $4$ quarters.

We adopt the same forecasting procedures utilized to forecast US and Australia datasets in [6] and [7]. We consider a lag parameter equivalent to $4$ quarters for prediction models. The number of predictions for testing the models is 100 predictions for US dataset and 75 for Australia dataset. Finally, the prediction step is performed using a rolling window procedure (we move one step each time, update models based on last values, and compute the next forecasts), and we focus on the first horizon forecast.

### *E. Forecast accuracy measures*

Two measures of forecast accuracy are used. The classical root mean square error (RMSE), and the mean absolute scaled error (MASE). The MASE is based on the errors of the forecasts and the mean absolute error of the naive method on the in-sample:

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{h}(y_{t+n} - \hat{y}_{t+n})^2}{h}},$$

$$\text{MASE} = \frac{\frac{1}{h}\sum_{t=n+1}^{n+h} \mid y_t - \hat{y}_t \mid}{\frac{1}{n-1}\sum_{t=2}^{n} \mid y_t - y_{t-1} \mid},$$

where $h$ is the number of predictions, $(\hat{y}_{n+1}, \ldots, \hat{y}_{n+h})$ are the forecasts, $(y_{n+1}, \ldots, y_{n+h})$ are the real values.

## VII. RESULTS

We show in Tables I and II the forecast accuracy relative to naive-benchmark model using the RMSE and the MASE measures. These experiments are made on an single computer with processor 2,2 GHz Intel Core i7 and 16Gb of RAM. The evaluations are performed by executing the models (in rows) on all the features generated by all the methods (in columns). Due to the computational time limitations, and the large number of the obtained models with the associated subset of predictors, we fix the maximum number of the selected features at $20$, and we show only the optimal number $k$ that provides the best performance for each method, with an exception for the method named UFSM, which does no require an input parameter specifying the number of features.

## VIII. Discussion

In this section, we discuss the obtained results, and we try to explain some findings. We also compare the global results with papers [6] and [7].

The main observation is that the two instances of the proposed algorithm, i.e., based on the partitioning and the hierarchical clustering techniques, outperform the other methods for most of the target time series. This is especially visible when they are used with artificial neural networks models.

In general, the results reveal that the forecasts of all univariate models used are improved by multivariate models. However, univariate models are competitive for some variables. This is mainly due to the lack of relevant dependencies between some variables. For example, with the GDP variable of Australian datasets, we slightly improved the naive-benchmark model, only by using the pGFSM and hGFSM methods.

When forecasting the CPI variable of Australian dataset, authors of [7] showed that the used multivariate models do not improve the forecast accuracy of the AR(4) and the naive-benchmark univariate models. Nevertheless, we show that when we use the GFSM method with the LSMT model, we improve the accuracy of forecast for this variable. As a consequence, we argue that some of the obtained forecasts in [7] and [6] can be improved using the proposed feature selecting algorithm and neural network models.

In addition, we note that the best results are obtained generally with a number of variables less than $15$. This confirms relatively the results in [7], where the best results of multivariate models are obtained using a number of predictor variables less than $20 - 40$.

As a side note, we do not discuss the statistical significance of the forecast accuracy. It is worth to mention that some authors, such as [44], have argued that statistical significance testing of forecast accuracy should be avoided, as test results may be misleading and that practice may actually harm the progress of forecasting field. We also notice that for some target variables, in which, the UFSM method is applied to select the predictor variables, the VECM model could not be fitted, and we do not have predictions for those variables. This is mainly due to the important number of features generated by this method, and this may cause some problems with matrix operations (obtaining singular matrix) when fitting the parameters of the VECM model. To avoid this problem, one should reduce more the number of predictor variables or reduce the lag parameter. Another alternative consists in utilizing artificial neural networks or adopt shrinkage approaches to solve linear models.

Finally, we note that for the PCA and Kernel PCA dimension reduction methods, it is possible to have both automatic number of features $k$ or a specific number given in the input. Currently, our proposal requires from the user the number of features to be selected. It is equal to the number of clusters generated by the clustering method used inside the algorithm. Consequently, this algorithm can be extended to provide an automatic number of features by using some methods of automatic selection of the optimal number of clusters [45], [46].

## IX. Conclusions

In the literature, a little attention has been paid to the role of causality in feature selection for multiple time series forecasting. While the impact of direct dependencies between variables is not negligible in many types of real time series, and the causality may help to detect the most relevant predictor variables. In this paper, we investigated its role and we proposed a feature selection algorithm specific to time series forecasting with the idea of avoiding duplicated dependencies between the predictor variables using a clustering approach. The causality measure adopted for the experiments is the Granger causality, but the proposed algorithm is applicable for other similarity measures, for instance, the transfer entropy.

We have presented a benchmark experiments, by evaluating some two-step approaches, that are based on feature selection and dimension reduction techniques as a first step before applying prediction models. Experiments are conducted on real macroeconomic datasets of US an Australia [6], [7]. And we compared the proposed algorithm to some well known exiting methods, using several prediction models. The results show that the proposed algorithm is very competitive for both datasets in terms of RMSE and MASE as forecast accuracy measures, and works well with the VARMLP and the LSTM artificial neural network models.

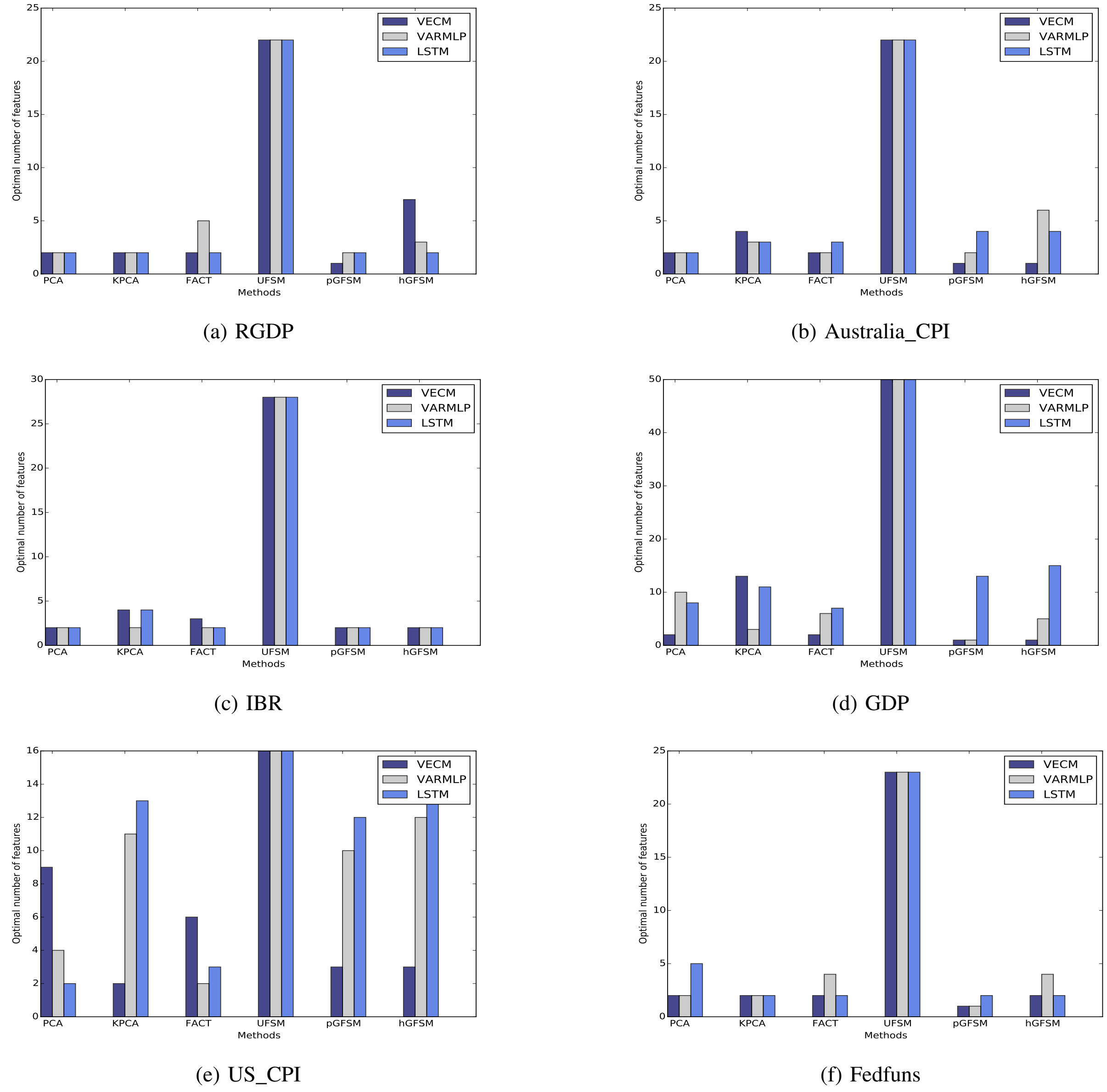
In future work, we aim to adopt a more deeper analysis on the graph of casualties than the clustering approach, in order to tackle dependencies between time series. As in the current work, we test our approach on macroeconomic datasets, we also aim to apply it on other type of data, as well as study the applicability of the feature selection methods on the types of the models (i.e., prediction, regression and others).

TABLE I. The Forecast Accuracy Results Using The Rmse Measure, Relative To The Naive-Benchmark Model For AUSTRALIA And US Datasets.

| Datasets | Series | Models | Rmse | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Pca | Kernel Pca | Fact | Ufsm | pGfsm | hGfsm |
| US DATASET | CPI | Ar | 1.04 | 1.04 | 1.04 | 1.04 | 1.04 | 1.04 |
| | | Arima | 1.05 | 1.05 | 1.05 | 1.05 | 1.05 | 1.05 |
| | | Vecm | 0.97 | 0.97 | 0.95 | 1.27 | 0.81 | 0.76 |
| | | VarMlp | **0.54** | 0.96 | 1.00 | 0.97 | 0.87 | 0.86 |
| | | Lstm | 1.07 | 1.07 | 0.56 | 1.06 | 0.96 | 0.90 |
| | GDP | Ar | 0.92 | 0.92 | 0.92 | 0.92 | 0.92 | 0.92 |
| | | Arima | 0.93 | 0.93 | 0.93 | 0.93 | 0.93 | 0.93 |
| | | Vecm | 0.97 | 0.96 | 0.97 | | 0.88 | 0.88 |
| | | VarMlp | 0.88 | 0.89 | 0.94 | 1.25 | 0.85 | 0.78 |
| | | Lstm | 0.90 | 0.90 | 0.86 | 0.87 | **0.76** | 0.78 |
| | Fedfuns | Ar | 0.96 | 0.96 | 0.96 | 0.96 | 0.96 | 0.96 |
| | | Arima | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 |
| | | Vecm | 1.06 | 1.06 | 1.21 | | 0.98 | 0.94 |
| | | VarMlp | 0.86 | 0.86 | 1.12 | 2.07 | 0.84 | 0.80 |
| | | Lstm | 0.75 | **0.71** | 0.78 | 1.16 | 0.81 | 0.78 |
| AUSTRALIA DATASET | RGDP | Ar | 1.06 | 1.06 | 1.06 | 1.06 | 1.06 | 1.06 |
| | | Arima | 1.07 | 1.07 | 1.07 | 1.07 | 1.07 | 1.07 |
| | | Vecm | 1.24 | 1.99 | 1.34 | 3.28 | 1.32 | 1.29 |
| | | VarMlp | 1.02 | 1.08 | 1.15 | 1.67 | 1.02 | 1.02 |
| | | Lstm | 1.04 | 1.05 | 1.01 | 1.40 | 1.00 | **0.96** |
| | IBR | Ar | 0.92 | 0.92 | 0.92 | 0.92 | 0.92 | 0.92 |
| | | Arima | 0.92 | 0.92 | 0.92 | 0.92 | 0.92 | 0.92 |
| | | Vecm | 1.27 | 1.85 | 1.11 | 2.30 | 1.01 | 1.01 |
| | | VarMlp | 1.23 | 1.64 | 1.03 | 1.64 | 0.86 | 0.88 |
| | | Lstm | 1.14 | 0.86 | 1.02 | 1.71 | 0.80 | **0.75** |
| | CPI | Ar | 0.87 | 0.87 | 0.87 | 0.87 | 0.87 | 0.87 |
| | | Arima | 1.01 | 1.01 | 1.01 | 1.01 | 1.01 | 1.01 |
| | | Vecm | 1.06 | 1.83 | 1.03 | 1.04 | 1.02 | 1.02 |
| | | VarMlp | 0.98 | 1.03 | 1.00 | 1.01 | 0.78 | **0.77** |
| | | Lstm | 1.02 | 1.03 | 1.00 | 1.01 | 0.82 | 0.88 |

TABLE II. The Forecast Accuracy Results Using The Mase Measure, Relative To The Naive-Benchmark Model For AUSTRALIA And US Datasets.

| Dataset | Series | Models | Mase | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Pca | Kernel Pca | Fact | Ufsm | pGfsm | hGfsm |
| US DATASETS | CPI | Ar | 0.87 | 0.87 | 0.87 | 0.87 | 0.87 | 0.87 |
| | | Arima | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| | | Vecm | 0.87 | 0.87 | 0.91 | 1.52 | 0.91 | 0.87 |
| | | VarMlp | 0.85 | **0.83** | 0.88 | 0.96 | 0.84 | 0.85 |
| | | Lstm | 0.94 | 0.95 | 0.89 | 0.97 | 0.84 | 0.89 |
| | GDP | Ar | 0.92 | 0.92 | 0.92 | 0.92 | 0.92 | 0.92 |
| | | Arima | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| | | Vecm | 0.95 | 0.95 | 1.00 | 0.00 | 0.91 | 0.91 |
| | | VarMlp | 0.90 | 0.90 | 0.99 | 1.26 | 0.82 | 0.82 |
| | | Lstm | 0.92 | 0.93 | 0.85 | 0.89 | **0.77** | **0.77** |
| | Fedfuns | Ar | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 |
| | | Arima | 0.86 | 0.86 | 0.86 | 0.86 | 0.86 | 0.86 |
| | | Vecm | 1.17 | 1.17 | 1.32 | | 1.04 | 0.95 |
| | | VarMlp | 0.93 | 0.93 | 1.16 | 2.06 | 0.93 | 0.85 |
| | | Lstm | 0.80 | 0.75 | 0.80 | 1.21 | 0.85 | **0.82** |
| AUSTRALIA DATASET | CPI | Ar | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 |
| | | Arima | 1.02 | 1.02 | 1.02 | 1.02 | 1.02 | 1.02 |
| | | Vecm | 1.21 | 1.74 | 1.11 | 1.11 | 1.10 | 1.10 |
| | | VarMlp | 1.01 | 1.05 | 1.13 | 1.04 | **0.83** | 0.88 |
| | | Lstm | 1.04 | 1.06 | 1.03 | 1.03 | 0.89 | 0.95 |
| | IBR | Ar | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 |
| | | Arima | 0.85 | 0.85 | 0.85 | 0.85 | 0.85 | 0.85 |
| | | Vecm | 1.33 | 1.52 | 1.20 | 2.18 | 0.98 | 0.98 |
| | | VarMlp | 1.28 | 1.46 | 1.20 | 2.03 | 0.88 | 0.91 |
| | | Lstm | 1.24 | 1.00 | 1.11 | 0.85 | 0.85 | **0.84** |
| | RGDP | Ar | 1.05 | 1.05 | 1.05 | 1.05 | 1.05 | 1.05 |
| | | Arima | 1.08 | 1.08 | 1.08 | 1.08 | 1.08 | 1.08 |
| | | Vecm | 1.37 | 1.96 | 1.46 | 2.98 | 1.33 | 1.33 |
| | | VarMlp | 1.07 | 1.13 | 1.15 | 1.08 | 1.02 | 1.07 |
| | | Lstm | 1.10 | 1.11 | 1.06 | 1.43 | **0.99** | 1.00 |



(a) RGDP

(b) Australia_CPI

(c) IBR

(d) GDP

(e) US_CPI

(f) Fedfuns

Figure 4. The Best Number Of Features According To RMSE Of The Methods Used.

## REFERENCES

[1] Y. Hmamouche, A. Casali, and L. Lakhal, “A causality-based feature selection approach for multivariate time series forecasting,” in DBKDA, 2017, pp. 97–102.

[2] C. W. J. Granger, “Testing for causality,” Journal of Economic Dynamics and Control, vol. 2, Jan. 1980, pp. 329–352.

[3] G. Walker, “On Periodicity in Series of Related Terms,” Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character, vol. 131, no. 818, 1931, pp. 518–532.

[4] P. Whittle, “The Analysis of Multiple Stationary Time Series,” Journal of the Royal Statistical Society. Series B (Methodological), vol. 15, no. 1, 1953, pp. 125–139.

[5] J. H. Stock and M. W. Watson, “Chapter 10 Forecasting with Many Predictors,” in Handbook of Economic Forecasting, C. W. J. G. G. Elliott and A. Timmermann, Eds. Elsevier, 2006, vol. 1, pp. 515–554.

[6] ——, “Generalized Shrinkage Methods for Forecasting Using Many Predictors,” Journal of Business & Economic Statistics, vol. 30, no. 4, Oct. 2012, pp. 481–493.

[7] B. Jiang, G. Athanasopoulos, R. J. Hyndman, A. Panagiotelis, and F. Vahid, “Macroeconomic forecasting for Australia using a large number of predictors,” Monash University, Department of Econometrics and Business Statistics, Monash Econometrics and Business Statistics Working Paper 2/17, 2017.

[8] X. Zhong and D. Enke, “Forecasting daily stock market return using dimensionality reduction,” Expert Systems with Applications, vol. 67, 2017, pp. 126–139.

[9] I. T. Jolliffe, “Principal Component Analysis and Factor Analysis,” in Principal Component Analysis, ser. Springer Series in Statistics. Springer, New York, NY, 1986, pp. 115–128.

[10] B. Schölkopf, A. Smola, and K.-R. Müller, “Nonlinear Component Analysis as a Kernel Eigenvalue Problem,” Neural Computation, vol. 10, no. 5, Jul. 1998, pp. 1299–1319.

[11] J. Geweke, “The dynamic factor analysis of economic time series,” Latent Variables in Socio-Economic Models, 1977.

[12] J. H. Stock and M. W. Watson, “Forecasting Using Principal Components From a Large Number of Predictors,” Journal of the American Statistical Association, vol. 97, no. 460, Dec. 2002, pp. 1167–1179.

[13] J. H. Stock and M. Watson, “Dynamic Factor Models,” in Oxford Handbook on Economic Forecasting. Oxford University Press, 2011.

[14] R. Tibshirani, “Regression Shrinkage and Selection Via the Lasso,” Journal of the Royal Statistical Society, Series B, vol. 58, 1994, pp. 267–288.

[15] A. E. Hoerl and R. W. Kennard, “Ridge Regression: Biased Estimation for Nonorthogonal Problems,” Technometrics, vol. 12, no. 1, 1970, pp. 55–67.

[16] J. H. Wright, “Forecasting US inflation by Bayesian model averaging,” Journal of Forecasting, vol. 28, no. 2, Mar. 2009, pp. 131–144.

[17] A. Carriero, G. Kapetanios, and M. Marcellino, “Forecasting large datasets with Bayesian reduced rank multivariate models,” Journal of Applied Econometrics, vol. 26, no. 5, Aug. 2011, pp. 735–761.

[18] D. Korobilis, “Hierarchical shrinkage priors for dynamic regressions with many predictors,” International Journal of Forecasting, vol. 29, no. 1, Jan. 2013, pp. 43–59.

[19] A. Abraham, B. Nath, and P. K. Mahanti, “Hybrid Intelligent Systems for Stock Market Analysis,” in Computational Science - ICCS 2001. Springer, Berlin, Heidelberg, May 2001, pp. 337–345.

[20] H. Yoon and C. Shahabi, “Feature subset selection on multivariate time series with extremely large spatial features,” in Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on. IEEE, 2006, pp. 337–342.

[21] I. Koprinska, M. Rana, and V. G. Agelidis, “Correlation and instance based feature selection for electricity load forecasting,” Knowledge-Based Systems, vol. 82, Jul. 2015, pp. 29–40.

[22] G. Box, “Box and Jenkins: Time Series Analysis, Forecasting and Control,” in A Very British Affair, ser. Palgrave Advanced Texts in Econometrics. Palgrave Macmillan UK, 2013, pp. 161–215.

[23] M. H. Quenouille, “The analysis of multiple time-series,” 1957.

[24] S. Johansen, “Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models,” Econometrica, vol. 59, no. 6, 1991, pp. 1551–1580.

[25] J. N. D. Gupta and R. S. Sexton, “Comparing backpropagation with a genetic algorithm for neural network training,” Omega, vol. 27, no. 6, Dec. 1999, pp. 679–684.

[26] K. Khan and A. Sahai, “A Comparison of BA, GA, PSO, BP and LM for Training Feed forward Neural Networks in e-Learning Context,” International Journal of Intelligent Systems and Applications, vol. 4, no. 7, pp. 23–29.

[27] D. U. Wutsqa, “The Var-NN Model for Multivariate Time Series Forecasting,” MatStat, vol. 8, no. 1, Jan. 2008, pp. 35–43.

[28] A. D. Aydin and S. C. Cavdar, “Comparison of Prediction Performances of Artificial Neural Network (ANN) and Vector Autoregressive (VAR) Models by Using the Macroeconomic Variables of Gold Prices, Borsa Istanbul (BIST) 100 Index and US Dollar-Turkish Lira (USD/TRY) Exchange Rates,” Procedia Economics and Finance, vol. 30, Jan. 2015, pp. 3–14.

[29] D. U. Wutsqa, S. G. Subanar, and Z. Sujuti, “Forecasting performance of VAR-NN and VARMA models,” in Proceedings of the 2nd IMT-GT Regional Conference on Mathematics, 2006.

[30] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., vol. 9, no. 8, Nov. 1997, pp. 1735–1780.

[31] T. Schreiber, “Measuring Information Transfer,” Physical Review Letters, vol. 85, no. 2, Jul. 2000, pp. 461–464.

[32] L. Barnett, A. B. Barrett, and A. K. Seth, “Granger causality and transfer entropy are equivalent for Gaussian variables,” Physical Review Letters, vol. 103, no. 23, Dec. 2009.

[33] Y. Sun, J. Li, J. Liu, C. Chow, B. Sun, and R. Wang, “Using causal discovery for feature selection in multivariate numerical time series,” Machine Learning, vol. 101, no. 1-3, Jul. 2014, pp. 377–395.

[34] X. Zhang, Y. Hu, K. Xie, S. Wang, E. W. T. Ngai, and M. Liu, “A causal feature selection algorithm for stock prediction modeling,” Neurocomputing, vol. 142, Oct. 2014, pp. 48–59.

[35] M. C. K. Babu, P. Nagendra, “IJETT - Survey on Clustering on the Cloud by UsingMap Reduce in Large Data Applications,” International Journal of Engineering Trends and Technology.

[36] Y. Wu, Y. Zhu, T. Huang, X. Li, X. Liu, and M. Liu, “Distributed Discord Discovery: Spark Based Anomaly Detection in Time Series,” in 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, Aug. 2015, pp. 154–159.

[37] A. P. Reynolds, G. Richards, B. de la Iglesia, and V. J. Rayward-Smith, “Clustering Rules: A Comparison of Partitioning and Hierarchical Clustering Algorithms,” Journal of Mathematical Modelling and Algorithms, vol. 5, no. 4, Dec. 2006, pp. 475–504.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, no. Oct, 2011, pp. 2825–2830.

[39] M. E. Tipping and C. Bishop, “Probabilistic Principal Component Analysis,” Journal of the Royal Statistical Society, Series B, vol. 21/3, Jan. 1999.

[40] F. Murtagh and P. Legendre, “Ward’s Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward’s Criterion?” Journal of Classification, vol. 31, no. 3, Oct. 2014, pp. 274–295.

[41] R. Hyndman, M. O’Hara-Wild, C. Bergmeir, S. Razbash, and E. Wang, “Forecast: Forecasting Functions for Time Series and Linear Models,” Feb. 2017.

[42] F. Chollet and others, “Keras: Deep learning library for theano and tensorflow,” URL: https://keras. io/k, 2015.

[43] H. Akaike, “A new look at the statistical model identification,” IEEE Transactions on Automatic Control, vol. 19, no. 6, Dec. 1974, pp. 716–723.

[44] J. S. Armstrong, “Significance tests harm progress in forecasting,” International Journal of Forecasting, vol. 23, no. 2, Apr. 2007, pp. 321–327.

[45] L. Kaufman and P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, Sep. 2009.

[46] R. Tibshirani, G. Walther, and T. Hastie, “Estimating the number of clusters in a data set via the gap statistic,” Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 63, no. 2, Jan. 2001, pp. 411–423.