Improving multivariate time series forecasting with random walks with restarts on causality graphs

Piotr Przymus*, Youssef Hmamouche†, Alain Casali‡ and Lotfi Lakhal§
Laboratoire d'Informatique Fondamentale de Marseille, CNRS UMR 7279
Aix-Marseille University
Marseille, France
Email: * eror@mat.umk.pl, † youssef.hmamouche@lif.univ-mrs.fr,
‡ alain.casali@lif.univ-mrs.fr, § lotfi.lakhal@lif.univ-mrs.fr

***Abstract*—Forecasting models that utilize multiple predictors are gaining popularity in a variety of fields. In some cases they allow constructing more precise forecasting models, leveraging the predictive potential of many variables. Unfortunately, in practice we do not know which observed predictors have a direct impact on the target variable. Moreover, adding unrelated variables may diminish the quality of forecasts. Thus, constructing a set of predictor variables that can be used in a forecast model is one of the greatest challenges in forecasting. We propose a new selection model for predictor variables based on the directed causality graph and a modification of the random walk with restarts model. Experiments conducted using the two popular macroeconomics sets, from the US and Australia, show that this simple and scalable approach performs well compared to other well established methods.**

***Index Terms*—Economic forecasting, Forecasting, Time series analysis, Data mining, Machine learning**

I. Introduction

In the age of the Internet of Things, huge amounts of numerical time series are generated for various applications, like biology, medicine, finance, industry, and many others. As a result, the number of time series generated grows very quickly and falls under the category of problems in which the size and the dimensionality of data is a problem itself. This raises the need for time series analysis systems that is ready for such challenges.

One of the main goals behind the time series data mining is to describe the occurring patterns and the evolution of data over the time. Among many applications, time series forecasting takes special place as it influences the decision making processes. The basic approach to forecast is to use one of the univariate forecasting models. Those models are based on the auto-regressive principle, where historic observations of a given time series are used to make the forecasts. The most popular models are the Auto-Regressive (AR) model, the Auto-Regressive Moving Average (ARMA) model, and the Auto-Regressive Integrated Moving Average (ARIMA) model [1].

Because of their construction, one of the main drawbacks of those models is that they skip potentially exploitable, profitable information from other time series. Thus, multivariate forecasting models were developed. The underlying idea is that often the target variable may depend both (i) on its past values and (ii) on the other related variables [2]. The classical examples of such algorithms are extensions of the AR model, like the Vector Auto-Regressive (VAR) model, and its co-integrated variant, the Vector Error Correction Model (VECM) [2].

While both previous models work well with time series having moderate dimension, they may struggle with high dimensional data. In some cases, domain knowledge could be used to select proper predictors [3], [4] but this is not always possible. Therefore, when dealing with high dimensional time series with unclear dependencies structure, two questions arise: how to select (or create) predictor variables automatically for forecasted variable and what is the impact of the number of used predictors on the predictions accuracy. Both questions are intensively studied by researchers, yielding new findings and approaches [5], [6], [3], [4].

New approaches based on regression with shrinkage and/or regularization constrains were proposed. The idea behind is to decrease the importance of dimensions that have low impact on predicted variable. Methods like Least Absolute Shrinkage and Selection Operator (LASSO) or Least-Angle Regression (LARS) are a popular choice [5], [6], [3], [4]. Those approaches address the above problems indirectly by setting different constraints on the coefficients by either reducing the impact or removing the unimportant variables.

Artificial neural networks approaches are also intensively studied in this context. For improving univariate time series forecasting, those approaches have been extended by using other layers typically founded in deep learning approaches and/or by combining them with autoregressive models [7], [8]. The multivariate case can be considered as an extension of univariate models, modified in such a way that it accepts multiple time series as input.

A more direct approach is to use a two step procedure. In the first step data is preprocessed in order to select or create new variables that will be used as input. In the second step a multivariate forecasting model is trained, any multivariate model may be used, including neural networks and models based on shrinkage and/or regularization. Among such methods it is worth to mention dynamic factor analysis, dimension reduction and feature selection methods [9], [10], [5], [3], [11], [12].

In this paper we discuss approaches that falls into the feature

selection methods. The proposed approaches are both based on the notion of causality (more precisely predictive causality) and random walks with restarts on weighted graphs. The main motivation for using causality measures is based on two properties. First, the cause happens before the effects (there is a time lag between). Secondly, the cause carries unique information about about the future effects. Those unique properties make causality measures a perfect candidate in considered use case. To establish the causality we use two well known and most popular notions of causality: Granger causality (a statistical test) and Transfer entropy (a information theory concept). Thus we propose two new approaches Granger graph Random Walks with Restart (GRWR) and Transfer entropy Random Walks with Restart (TRWR). While conceptually and computationally simple, they give competitive results compared to other, often more complicated, methods. To advocate these approaches, we perform extended evaluation on two real macroeconomic datasets [3], [4]; they turn out an attractive alternative.

The second contribution of this paper is a consequence of the extended evaluation. We have implemented and tested feature selection methods and forecasting methods that were not evaluated against [3], [4] datasets. Thus, we complement research performed in [3], [4] and re-evaluate their findings with other approaches.

The organization of the paper is as follows. Section II discusses the related work, Section III describes and formulates the notion of predictive causalities, i.e. Granter causality and Transfer Entropy. In Section IV, we present our approach. Then, in Sections V, VI and VII we describe the methods and the models used in the experiments, detail the experiment setup and discuss the results. In the last section we conclude the research.

II. Related work

We present here some of the numerous approaches proposed in the literature to handle the problem of forecasting multivariate time series with large number of predictors. In this section we will assume that we have a n-variable dataset $X(t) = [x_1(t), \dots x_n(t)]$, and a target (forecasted) variable x_f where $0 \leq f \leq n$.

The dynamic factor model (DFM) and its derivatives assume that all the variables are driven by a few common unobserved dynamic factors $F(t) = [f_1(t), ..., f_d(t)]$, with dimension $d \ll n$. Each of the variables can be expressed as $x_i(t) = \lambda_i F(t) + u_i(t)$, where u_i represents idiosyncratic error terms. Forecasting is done for each of the factors $F(t)$ and forecasted values are used to reconstruct forecasts for variables from $X(t)$. Various procedures were used to establish the factors, e.g. principal components or factor analysis can be used [5], [3].

Dimension reduction was also combined with various forecasting models for daily stock market [9] and for time series class prediction in [10]. Due to the large number of features, the authors in [9] propose a data mining forecasting process. First, they use one of three versions of principal component analysis (PCA) to construct a dimension reduction transformation. Then, they use a multilayer neural network in the forecasting step. The main difference to the DFM approach is that the transformed data is put directly in to the forecasting model for target variable $x_i(t)$.

The most popular approach is based on regression with shrinkage and/or regularization. The intuition behind these methods is to fit a forecasting model with some additional constrains. Generally many of predictors in dataset will have a minor or no effect on the response and may discard the effect of the mayor impacting variables. To effectively deal with such variables, various constrains are set upon the coefficient, like shrinking the irrelevant coefficients to zero, or by forcing them to be equal to zero. Depending on the type of the constrains, as a side effect we can also obtain feature selection (like in case of LASSO [13]). A detailed discussion on various methods including evaluation may be found in [5], [6], [3], [4].

Artificial Neural Networks (ANNs) are also very useful in time series forecasting, and can handle data with many variables. Application for forecasting is based on transforming the time series data into supervised learning problem depending on a look-back parameter. Also many linear statistical models can be extended to non-linear version via ANNs. For instance, in [14], the ARIMA model was compared to univariate ANNs model, no model outperform the other for all variables and a combination of these models is proposed to improve the forecast accuracy. In [15], the authors propose the Vector Autoregressive Neural Network (VAR-NN) model, using a multilayer perceptron network. The Long Short Term Memory network model (LSTM) was proposed in [16]. It is a recurrent neural network, characterized by using short and long-term information, which is a main drawback of feed-forward networks.

Finally there are methods that are based on the notion of predictive causality. The main idea is to test if past information of one variable can be used to improve forecasting of another variable. If it is the case, we say that the first variable causes the second one. In this work we focus on two widely used concepts of causality the Granger causality [17] and the transfer entropy [18]. First one is based on a statistical notion of causal influence of variables expressed via vector auto regression. The second one is an information theory measure of directed information transfer between jointly dependent processes in time. Both concepts have a vast variety of applications in econometrics and finance [17], [12], [11], [19].

In [12], the Granger causality is used for feature selection. For each target variable a subset of predictors is selected. The predictor is added to the subset if it causes the target variable but the target does not cause the predictor (based on a threshold statistical test). The authors show good results compared to various dimension reduction methods in various regression and classification tasks. Another approach, based on clustering of the graph of Granger causalities, was proposed in [11]. The model shows good properties, outperforming benchmark models, dimension reduction based models, as well as the model from [12].

III. The graph of causality

Various measures for describing dependencies between variables of multivariate time series were discussed in the literature. Popular choices like correlation and mutual information are symmetric measures, used for time series forecasting (see e.g. [20]). In contrast, the predictive causality describes the predictive relationships between variables, i.e. which variables influence the other variables and thus is not symmetric. In this section we discuss two widely used notions of predictive causality in the context of time series. Namely, the Granger causality and the transfer entropy.

A. The Granger (predictive) causality

The Granger causality [17] is based on the concept of predictability. It supposes that a univariate time series $x(t)$ causes another time series $y(t)$ if using $x(t)$ can significantly improve forecast accuracy of $y(t)$. Thus we will build two models:

$$y(t) = \alpha_0 + \sum_{j=1}^{p} \alpha_j y(t-j) + \epsilon(t), \quad (1)$$

$$y(t) = \alpha_0 + \sum_{j=1}^{p} \alpha_j y(t-j) + \sum_{j=1}^{p} \beta_j x(t-j) + \epsilon(t), \quad (2)$$

where the first model is built using only $y(t)$, and the second model uses both $x(t)$ and $y(t)$. The comparison between the models is done using a statistical test to establish the significance of the comparison. A common choice is to compute the F-test:

$$F = \frac{(\mathrm{SSR}_1 - \mathrm{SSR}_2)/p}{\mathrm{SSR}_2/(n-2p-1)}, \quad (3)$$

where SSR_1 and SSR_2 are the sum of squared residuals related to models 1 and 2 respectively, n is the size of the predicted vector. Then we check two hypothesis, H_0: 'x does not cause y', and H_1: 'x causes y', i.e.

$$H_0 : \forall_{j\in 1,\dots,p}, \beta_j = 0,$$
$$H_1 : \exists_{j\in 1,\dots,p}, \beta_j \neq 0.$$

Under the null hypothesis H_0, F follows the Fisher distribution with parameters p and $(n-2p-1)$. Then the test is carried out at a level α in order to examine the null hypothesis of non causality.

Let us emphasize that the precedent expressions are designed for stationary times series. In general, if the time series are not stationary, a differencing step is required before applying the Ar and the Var models.

B. The transfer entropy

The transfer entropy was formulated by T. Schreiber [18], as a information theory measure of directed (time-asymmetric) information transfer between joint processes. In contrast to G-causality, transfer entropy is not based on predictive causality but on resolution of uncertainty. The transfer entropy from Y to X can be seen as the degree to which Y disambiguates the future of X beyond the degree to which X already does.[1] There is therefore an attractive symmetry between the notions (predicts $\leftrightarrow$ disambiguates), see e.g. [21].

Consider two processes X and Y with probability distribution $p(x)$ and $p(y)$, joint probability $p(x,y)$, and conditional probability $p(x \mid y)$. The mutual information between the two process X and Y measures the mutual dependencies between the two variables. Thus, it is symmetric and does not model the transfer of information from one process to another. The mutual information between two processes can be expressed using Kullback entropy:

$$I(X;Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\cdot log(\frac{p(x|y)}{p(x)p(y)}).$$

In [18], the transfer entropy was suggested as an extension of mutual information, that allows to study the dynamics and non symmetric transfers of information between process. The transfer entropy expresses the information flow from X to Y:

$$T_{X\to Y} = \sum p(y_{t+1}, y_t^{(l)}, x_t^{(m)}) \cdot log(\frac{p(y_{t+1} \mid y_t^{(l)}, x_t^{(m)})}{p(y_{t+1} \mid y_t^{(l)})}), \quad (4)$$

where $z_t^{(l)} = (z_t, \dots, z_{t-l+1})$ for $z = x, y$, and l, m are lag parameters.

IV. Methodology

Our goal is to construct a multivariate model that improves quality of forecasts compared to univariate models and other multivariate models. As we discussed earlier, building a model using a large number of predictors may be problematic, since many of variables may have a minor or no effect on the forecasted variable. What is more, such variables can distort the effect of more significant variables.

A common approach in forecasting is to minimize the impact of irrelevant variables, either by minimizing their impact in the model (by shrinking their coefficients towards 0), or by simply building a model using only selected features.

Unfortunately, the number of possible subsets of the variables grows exponentially with the number of observed variables. Thus we investigate an heuristic algorithm for proper variables subset selection for a forecasting algorithm.

Let $X(t) = [x_0(t), .., x_n(t)]$ be a multivariate time series. We are going to forecast the value of x_f for a fixed $0 \leq f \leq n$. For each pair of variables x_i, x_j, where $0 \leq i \neq j \leq n$, we will compute the causality using either Granger causality test or transfer entropy. We will represent the relationships in a directed weighted graph $G(V, E)$ (or in short G), where each node in V represents a variable x_i, and each edge in E with a weight a represents how variable x_i influences variable x_j under the assumed causality notion. By G^T we denote the transposition of the directed weighted graph G, and by M we will denote its adjacency matrix.

[1] In this section we use the original notation from [18], i.e. we use capital letters for single variables, and time is in the subscript.

A. The framework

In this section we will overview the whole framework, then in following section we will discuss individual steps details. The framework works the following way. First we compute both causality graphs, Granger causality and Transfer entropy as described in IV-B. Then, for each graph we check two strategies:

- **Follow to the source** of causality in the neighbourhood of predicted variable, that is we change the direction of edges and compute the ranking on G^T.
- **Most influential** variables in the neighbourhood of predicted variable, i.e. leave the original orientation of edges.

For each graph and strategy we compute the influence ranking of predictor variables following IV-C. For each ranking we generate a number of subsets of input variables. Let m be the maximal number of variables that can be used to forecast the model. We evaluate m subsets of variables $S_1, S_2, \ldots S_m$, where $S_i = \{$ top k variables from ranking $\}$.

Usually, the target variable will have high position in the ranking. But, in some cases it may not be included in the top k values. In such cases the target variable has to be added to the set of used variables.

Fig. 1: Two step flow

Finally, we evaluate various types of forecasting models on each set of top variables, see Fig. 1 for example. In this work first we use TRWR and GRWR to generate the subsets of variables using both strategies. Then each subset is evaluated using VECM and LSTM forecasting models (for details see Section V-C).

Let m be the maximal number of variables that the can be used to forecast the variable x_f, then proposed heuristic requires checking of just $O(m)$ sets of variables.

B. Building the causality graphs

The proposed approach operates on the graphs that describe causality relations between variables using two notions of causality – we construct two directed weighted graphs. The G_T graph represents relations based on the transfer entropy, and the G_G graph represents the Granger causality relations. We construct those graphs as follows.

To construct the transfer entropy graph G_T, for each pair of variables we put an edge with weight equal to equation 4, that is $T_{x_i \to x_j}$. Thus, the entries of the adjacency matrix M are equal to $M_{i,j} = T_{x_i \to x_j}, i \neq j$ and M has zeros on the diagonal, i.e. $M_{i,i} = 0$. To simplify the matter, we use transfer entropy with lag parameter equal to 1.

Construction of the graph based on Granger test G_G requires computation of causality between each pair of variables $(x_i \to x_j)$. This can be computed based on the value of F expressed in equation (3), thus the adjacency matrix form $M_{i,j} = F$. In order to evaluate the significance of causality, a critical value of F can be estimated based on the statistical Fisher-test. Granger causality can also be expressed as $M_{i,j} = 1 - pvalue$, where the $pvalue$ is the probability of observing the given result under the assumption that H_0 is true, which means the probability of non causality. The lag parameter for Granger causality test is established automatically using the Akaikes Information Criterion (AIC) [22], similarly to the way of selecting the lag order for VAR model.

As we compute causality effects between all variables, the resulting graphs are complete. Additionally, based on the properties of equations 4 and 3, both adjacency matrices are non negative.

C. Influence ranking of predictor variables for given target

Having the causality graph and selected target variable, we want to establish the ranking of "influence" among all variables. For that, we use random walks with restarts algorithm (or personalized PageRank algorithm). The algorithm restart point is set to the target variable.

To meet the convergence requirements, we associate a column (right) stochastic matrix with the adjacency matrix M (i.e. a graph in which for each node the sum of the weights of all of the out-edges is equal to 1). This can be achieved by fixing columns that sums to 0, and by scaling the weights in columns. Mind that columns with all elements equal to 0 will be very rare in practice. Nevertheless, they can be either eliminated or explicitly set to $1/N$ for each outgoing element (in case if column represents the target variable). The last required preparation step, rescaling, is done by summing all weights in the column and dividing each element weight by the sum.

Finally, we follow the standard formula for computing the random walks with restarts, i.e.

$$A = \beta \cdot M \cdot v + (1 - \beta) \cdot e_i, \tag{5}$$

where the e_i is $N \times N$ matrix that has 1 for each element in the row i and 0s elsewhere. Resulting matrix is column stochastic and represents irreducible and aperiodic Markov chain. Thus it belongs to the classes of Markov chains and the Perron-Frobenius operators. Commonly RWR is solved by finding the principal eigenvalue and associated principle eigenvector of A, either by using the power iteration method, or by using numeric solvers (e.g. LAPAC).

We will treat the resulting vector as a ranking of how much other variables influence the target one.

D. Scalability

We will assume that computing Granger causality or Transfer entropy for a pair of variables x_i, x_j is doable on a single node. For financial time series this is a reasonably safe assumption. In cases, when the number of observations per time series is huge, it is still possible to scale this step, for details please refer to [23].

Building the causality graph can be efficiently computed in a scalable way, as we have to compute:

- for Granger graph – univariate models for all variables, and the bivariate models for all pairs $x_i, x_j \in X$;
- for Transfer entropy – compute the transfer entropy for all pairs $x_i, x_j \in X$.

This can be easily done in a scalable way as all considered models can be computed independently. Then the results have to be gathered and post processed.

The second step is to compute the random walk with restarts. Once again, for financial applications it is safe to assume that resulting dense matrix will fit computing node memory, and thus principal eigenvector can be computed efficiently on a single node.

Finally various forecasting models have to be computed using the prepared sets of variables. This is trivial as all tasks are independent.

Thus, all steps are naively scalable, under the assumption that a single node is sufficient to compute the causality for two variables and fit a forecasting model for a variable. If this assumption is not met, it is necessarily to use more advanced techniques mentioned above.

V. The evaluation: methods

In this Section we present the evaluated methods that can be divided into two groups: baseline and two step approaches. Short summary of all used approaches can be found in Figure 2.

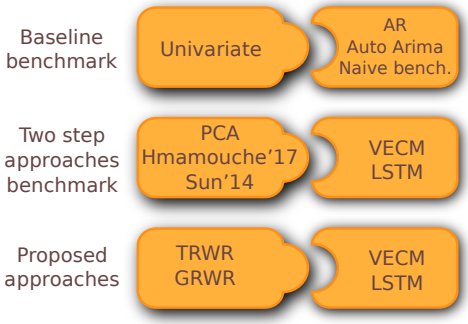

Fig. 2: Evaluation methods summary.

A. The baseline benchmark

Both papers [3], [4] compare advanced methods against a univariate benchmark consisting of well known univariate models. We follow their methodology and compare against following models:

- Naive benchmark: provides forecasts based on the sample mean.
- AR model: predicts a single time series based on its precedent values [1].
- Auto ARIMA model [24]: automatically determines the parameters of the Arima model; takes into account stationary and precedent error terms in addition to precedent values of a univariate time series to predict [1].

B. Two step approaches benchmark

For comparison, we have also implemented variants of several two-steps approaches found in the literature. The difference lays in the second step as we decided to use two forecasting models, LSTM and VECM (see the next section for details).

- An approach based on PCA, see [25]. The PCA is used to generate new variables that are then passed to a forecasting model.
- The Granger feature selection model [12] (referenced as Sun'14). Variables are selected if they have a significant causality on the target. This is done by comparing causality in both direction between the target and the variable. The variable is selected if the difference between the causalities is greater than some threshold.
- The algorithm for selecting variables by clustering the graph of Granger causalities [11] (referenced as Hmamouche'17). This algorithm is based on a Granger causality graph. Then, the graph is clustered and feature selection is performed based on the clustering. For each cluster one representative variable is selected.

Similarly to [3] we compute PCA using full data sample. Granger causality and transfer entropy are also established the same way.

The PCA and Hmamouche'17 will be used to generate subsets of variables of different length. For each length both algorithms have to be recomputed. The Sun'14 algorithm generates just one subset of variables for each target.

C. The forecasting step for two step approaches

This step is used as a second step for methods from section V-B and TRWR and GRWR approaches described in section IV.

One of the forecasting models used is the Vector Error Correction (Vec) Model [2]. For completeness we add here its brief description. This model extends the standard Var model by considering non-stationarity and cointegration. Let $X(t)$ be a multivariate time series, the $Var(p)$ process considers each variable of $X(t) = [x_1(t), \ldots, x_n(t)]$ as a linear combination of its p previous values and of the other variables:

$$X(t) = \alpha_0 + \sum_{i=1}^{p} A_i X(t-i) + \epsilon(t),$$

where ϵ_t is a white noise error terms, and $A_1, \ldots, A_p$ are $(n \times n)$ matrix representing the coefficients of the model.

In the case when $Var(p)$ process is non-stationary (i.e. $det(\Pi = (I_n - A_1 - \cdots - A_n)) = 0$) the Vec model is more appropriate.

Suppose that $X(t)$ has non-stationarity of order 1 or $I(1)$ (i.e. at least one variable of $X(t)$ is $I(1)$), then the Vec Model can be written as follows

$$\Delta X(t) = \Pi X(t-1) + \sum_{i=1}^{p-1} \Gamma_i \Delta X(t-i) + u(t),$$

where $\Delta X(t)$ is the differencing transformation of $X(t)$, (i.e. the difference between two consecutive observations of each non-stationary variable of $X(t)$) and Π is the cointegration matrix.

We evaluate also an ANNs model, that given a n-dimensional time series $y = [y_1, \ldots, y_n]$, a network function f_{nn} and a lag parameter p, predicts y at time t based on the p previous observations variables of all variables of y:

$$y(t) = f_{nn}(y_1(t-1), \ldots, y_n(t-p)) + \epsilon(t).$$

As a network we use the Long Short Term Memory network model (LSTM) [16] that belongs to the Recurrent Neural Networks (RNNs) class. It is characterized by handling the main limitation of feed-forward networks for time series, it maintains the previous information in the network.

In addition, LSTM is a specific model that is widely used in time series prediction and has the feature of exploiting long-term information passed in the network thanks to forgetting and remembering mechanisms. The equations of a hidden LSTM cell can expressed 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, σ 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 the output of the cell. The architecture of the network that we use is simple, we use the same number of units in hidden layer as the number of input variables. The model is build using keras library with adam solver.

VI. The evaluation procedure

A. The datasets

We perform the evaluation on two datasets:

- The US macroeconomic datasets [3]: quarterly numeric time series, containing 143 features and 200 observations, spanning the period $1960 - 2008$.
- The Australian macroeconomic dataset [4]: quarterly numeric time series spanning the period $1984 - 2015$, comprising 117 variables and 119 observations.

For both datasets each variable is transformed to stationary exactly the same way as it is done in original works, for detailed list of variables and used transformations please refer to [3], [4].

As found in [4], those datasets have different properties in the context of multivariate forecasting. First, methods that perform well in [3], may not necessarily work well in [4], and vice versa. Secondly, in [4] univariate models perform exceptionally well compared to multivariate ones.

One important observation done for both datasets is that using more than 40 variables will not improve results for tested multivariate models. Thus, we use this value as a limit of number of subsets that we generate with two step models that support this.

We follow the same data preprocessing and evaluation procedure as described in [3] and [4]. This allows us to relate our findings to the ones presented in both papers.

B. The forecasting procedures

The forecast are generated the rolling-window approach, which is one of the preferred approaches for time series evaluation [3]. In our case the rolling window moves up one quarter at a time. For the US data-sets, the number of predictions is 100 and for the Australian datasets 75.

C. The forecast accuracy measures

We use two measures for forecast accuracy, the root mean squared error ($Rmse$) and the mean absolute scaled error ($Mase$):

$$\text{RMSE} = \sqrt{\frac{\sum_{t=k+1}^{n}(x(t) - \hat{x(t)})^2}{n-k+1}},$$

$$\text{MASE} = \frac{\frac{1}{h}\sum_{t=k+1}^{n} \mid x(t) - \hat{x}(t) \mid}{\frac{1}{k-1}\sum_{t=2}^{k} \mid x(t) - x(t-1) \mid},$$

where $x_i(t)$ is the real time series, $\hat{x}_i(t)$ is the forecast, and $t = k+1$ is the horizon point. The RMSE is a standard measure used in the context of time series forecasting. The MASE is a measure introduced by [26], designed to be a data independent measure. The $Mase$ relates the forecasts errors to the mean absolute error of the naive method on the in-sample.

VII. Results

We computed forecasts for 143 and 117 variables, for US and Australian datasets respectively. For both datasets we used the limit of maximal number of variables in a subset equal to 40. Thus, for each variable we have computed 485 models, composed from: 3 univariate models, and 241 multivariate selection files that are composed with 2 types of multivariate models. Multivariate selection files contains: GRWR (80), TRWR (80), PCA (40), Hmamouche'17 (40), Sun14 (1) files. Let us underline that for TRWR and GRWR models, we use two strategies and thus we generate 80 subset for each model. Moreover, the Sun'14 algorithm automatically selects optimal number of variables and thus generates only one final subset. So in total, we did compute more than $100K$ models, from that we were able to successfully fit 57% of models.[2]

Total computation time was around 1 day on a 2 x Six-Core processor AMD(R) Opteron TM with 32 GB RAM. Not

[2]For intermediate and final forecast files, as well as for some details about failed models please refer to website https://przymusp.github.io/ts-causality-rwr/

Selection alg. + Forecasting alg.	Top-k MASE 1	MASE 2	MASE 3	RMSE 1	RMSE 2	RMSE 3
1. GRWR+Vecm	18	17	30	22	23	30
2. GRWR+lstm	0	1	0	1	0	0
3. Hmamouche'17+Vecm	60	49	45	61	51	53
4. TRWR+Vecm	34	34	48	23	39	34
5. TRWR+lstm	0	1	0	1	0	0
6. U+Ar	15	21	13	17	15	14
7. U+Auto_Arima	17	25	7	21	16	14
Ties						
Total	1	5	0	3	1	2
Ar and Auto_arima	1	4	0	2	1	2
Other ties		(1,3)		(1,3)		
% of dominating results						
GRWR+Vecm	12.6	11.9	21.0	15.4	16.1	21.0
GRWR+lstm	0.0	0.7	0.0	0.7	0.0	0.0
TRWR+Vecm	23.8	23.8	33.6	16.1	27.3	23.8
TRWR+lstm	0.0	0.7	0.0	0.7	0.0	0.0

TABLE I: Results for USA macroeconomic dataset [3].

surprisingly the majority of time was spent on computation of the LSTM type of models.

We gathered the results in two Tables I and II, where we have ranked all methods according to their ascending RMSE and MASE. Note that some of the two step methods are heuristics that search over some number of subsets, thus we report their performance as a group.

To improve the readability in the tables we present only the methods that ranked in top-$1, 2, 3$, and thus some methods do not appear in the results. In rare cases we have observed ties between methods, majority of them was observed for Ar and Auto Arima models, in two cases there was a tie between Hmamouche'17 and GRWR on LSTM. This can be found in ties sections of the tables.

In the last part of the table we present the percentage of variables for which TRWR and GRWR combined with VECM and LSTM dominated the forecast accuracy.

VIII. Discussion

In [3], [4] authors explored wide range of methods types like Dynamic Factor Model and Ridge regression, LASSO, LARS, Bagging LARS and Bayesian VAR. In this study we complement their research by examining various two step models derived from machine learning and data mining techniques. The evaluation we perform uses similar experimental setup to both papers.

We designed the experiment having in mind the following questions.

Q1: Does using TRWR and GRWR combined with VECM or LSTM improves prediction quality compared with baseline methods (naive benchmark, auto AR and Arima models)?

For US dataset [3] TRWR and GRWR combined with VECM take a considerable lead over Ar and Auto Arima models in top-1. Situation repeats for Australian dataset, where TRWR and GRWR with LSTM take lead over naive benchmark. This comparison is crucial as it is one of the goals of multivariate analysis to improve results compared to univariate. It is also natural that some of the variables work best with univariate models, as there is no guaranty that a causing variable is present in the dataset and using unrelated variables would just add noise. We note that two step approaches with ANNs for Australian dataset perform very well compared to naive benchmark. This puts a new perspective on [4] dataset as on of the [4] conclusions is that the naive sample mean benchmark is very competitive compared to the multivariate models on this data.

	Top-k MASE 1	MASE 2	MASE 3	RMSE 1	RMSE 2	RMSE 3
1. GRWR+Vecm	4	5	4	4	5	4
2. GRWR+lstm	38	26	27	38	26	27
3. Hmamouche'17+Vecm	2	1	2	2	1	2
4. Hmamouche'17+lstm	20	20	24	20	20	24
5. PCA+Vecm	2	0	2	2	0	2
6. PCA+lstm	7	19	8	7	19	8
7. Sun'14+lstm	4	1	1	4	1	1
8. TRWR+Vecm	2	0	1	2	0	1
9. TRWR+lstm	18	23	28	18	23	28
10. U+Ar	1	11	7	1	11	7
11. U+Auto_Arima	12	6	7	12	6	7
12. U+naive_benchmark	7	6	6	7	6	6
Ties						
Total	0	0	2	0	1	0
Ar and Auto_arima	0	0	2	0	1	0
Other ties						
% of dominating results						
GRWR+Vecm	3.4	4.3	3.4	4.3	6.0	3.4
GRWR+lstm	32.5	22.2	23.1	25.6	26.5	32.5
TRWR+Vecm	1.7	0.0	0.9	1.7	0.0	0.9
TRWR+lstm	15.4	19.7	23.9	11.1	15.4	10.3

TABLE II: Results for Australia macroeconomic dataset [4].

Q2: How does TRWR and GRWR compare to other two step approaches?

Both approaches perform surprisingly well compared to other two step approaches. For USA macroeconomic dataset the only two-step competitor is the Hmamouche'17 algorithm. This algorithm takes the major part in top-$1, 2, 3$ ranking. Fortunately, it still leaves some 'space' for TRWR and GRWR algorithms combined with VECM model. Thus TRWR and GRWR overtake the Ar and Auto Arima models, and take the second and third place in top-$1, 2, 3$ in general ranking, see TableI. For Australian dataset the GRWR with LSTM is the top method, followed by Hmamouche'17 with LSTM on second place and TRWR with LSTM. Other two step approaches can be also found in top-1,2,3, see Table II.

A. Threats to validity

Our approaches are based on the causality notion, thus it is pointless to use them for datasets that do not have cause and effect relations between variables.

Causality may change over time, so it is important to update the causality graph when this happens.

Furthermore, the performance of multivariate approaches is condemned to be data dependant, and even within one dataset it is hard to select one model that fits characteristics of all variables [3], [4]. Thus in practice we suggest to test a wide range of multivariate models for each variable.

IX. Conclusions and future work

In this paper we have presented a new approach for feature selection based on (predictive) causality for multivariate time series. Presented approach is based on a variant of random walks with restarts on directed causality graphs. This simple and scalable approach performs surprisingly well compared to other well established approaches. It is easy to compute and implement using existing frameworks. Important aspect of this approach is that it allows low cost computation of multiple subsets of variables compared to other two step approaches.

To evaluate the robustness of the proposed approach, we have conducted experiments on two macroeconomic datasets [3], [4]. The results are promising, as the TRWR and GRWR algorithms combined with VECM and LSTM models get top-1 results for significant number of variables in both datasets. Additionally, conducted experiments complements research done in [3], [4] with evaluation of variety of two step models. To make a full connection with results [3], [4], we plan to investigate one more research question: *Q3: How does TRWR and GRWR compares to shrinkage, DFM, LASSO, LARS – based on the literature results [3], [4]?* To tackle this question we plan to incorporate methods used in those papers into our framework. We look optimistic to this comparison, as we observe improvement compared to one of the conclusions in [4]. While, in [4] authors note that it is difficult to outperform the naive sample mean benchmark, we note that the two step approaches with LSTM's outperforms the naive sample benchmark on many variables.

We conclude that the proposed approach is quite competitive compared to other two step approaches and univariate models. Furthermore, it has potential compared to statistical approaches discussed in [3], [4].

In future work we plan to extend the evaluation with extra statistics and make a full comparison with methods used in [3], [4]. As for the proposed selection methods we plan to automatically reduce the number of candidate subsets generated. We also plan to investigate other causality related measures like 'momentary information transfer' [27]. In the light of theoretical findings of [21] that Granger causality and transfer entropy are equivalent for Gaussian variables, it would be also interesting to compare both GRWR and TRWR on datasets transformed using Box-Cox transformation.

References

[1] 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, pp. 161–215.

[2] S. Johansen, "Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models," *Econometrica*, vol. 59, no. 6, pp. 1551–1580.

[3] J. H. Stock and M. W. Watson, "Generalized Shrinkage Methods for Forecasting Using Many Predictors," *Journal of Business & Economic Statistics*, vol. 30, no. 4, pp. 481–493.

[4] B. Jiang, G. Athanasopoulos, R. J. Hyndman, A. Panagiotelis, and F. Vahid, "Macroeconomic forecasting for Australia using a large number of predictors."

[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, vol. 1, pp. 515–554.

[6] J. H. Stock and M. Watson, "Dynamic Factor Models," in *Oxford Handbook on Economic Forecasting*. Oxford University Press.

[7] M. Thielbar and D. A. Dickey, *Neural Networks for Time Series Forecasting: Practical Implications of Theoretical Results*.

[8] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks."

[9] X. Zhong and D. Enke, "Forecasting daily stock market return using dimensionality reduction," *Expert Systems with Applications*, vol. 67, pp. 126–139.

[10] E. Siggiridou and D. Kugiumtzis, "Granger Causality in Multivariate Time Series Using a Time-Ordered Restricted Vector Autoregressive Model," *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1759–1773.

[11] Y. Hmamouche, A. Casali, and L. Lakhal, "A Causality-Based Feature Selection Approach For Multivariate Time Series Forecasting," *The Ninth International Conference on Advances in Databases, Knowledge, and Data Applications*, pp. 97–102.

[12] Y. Sun, J. Li, J. Liu, C. Chow, B. Sun, and R. Wang, "Using causal discovery for feature selection in multivariate numerical time series," *Mach Learn*, vol. 101, pp. 377–395.

[13] R. Tibshirani, "Regression Shrinkage and Selection Via the Lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288.

[14] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175.

[15] D. U. Wutsqa, "The Var-NN Model for Multivariate Time Series Forecasting," *MatStat*, vol. 8, no. 1, pp. 35–43.

[16] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780.

[17] C. W. J. Granger, "Testing for causality," *Journal of Economic Dynamics and Control*, vol. 2, pp. 329–352.

[18] T. Schreiber, "Measuring Information Transfer," *Phys. Rev. Lett.*, vol. 85, no. 2, pp. 461–464.

[19] F. J. P. Thomas Dimpfl, "Using transfer entropy to measure information flows between financial markets."

[20] I. Koprinska, M. Rana, and V. G. Agelidis, "Correlation and instance based feature selection for electricity load forecasting," *Knowledge-Based Systems*, vol. 82, pp. 29–40.

[21] 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.

[22] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723.

[23] M. A. Rehab and F. Boufares, "Scalable Massively Parallel Learning of Multiple Linear Regression Algorithm with MapReduce," in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 2, pp. 41–47.

[24] R. Hyndman, M. O'Hara-Wild, C. Bergmeir, S. Razbash, and E. Wang, "Forecast: Forecasting Functions for Time Series and Linear Models."

[25] H. Yoon and C. Shahabi, "Shahabi: Feature Subset Selection on Multivariate Time Series with Extremely Large Spatial Features Data Mining Workshops," in *ICDM Workshops 2006. Sixth IEEE International Conference on Volume , Issue , Dec. 2006 Page(s):337 - 342 Digital Object Identifier 10.1109/ICDMW.2006.81*.

[26] R. J. Hyndman, "Another Look at Forecast-Accuracy Metrics for Intermittent Demand," *Foresight, International Journal of Applied Forecasting*, pp. 43–46.

[27] B. Pompe and J. Runge, "Momentary information transfer as a coupling measure of time series," *Phys. Rev. E*, vol. 83, no. 5, p. 051122.