University of Groningen

Automating Vector Autoregression on Electronic Patient Diary Data

Emerencia, Ando Celino; van der Krieke, Lian; Bos, Elisabeth H.; de Jonge, Peter; Petkov, Nicolai; Aiello, Marco

Published in:
IEEE Journal of Biomedical and Health Informatics

DOI:
[10.1109/JBHI.2015.2402280](https://doi.org/10.1109/JBHI.2015.2402280)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2016

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Emerencia, A. C., van der Krieke, L., Bos, E. H., de Jonge, P., Petkov, N., & Aiello, M. (2016). Automating Vector Autoregression on Electronic Patient Diary Data. *IEEE Journal of Biomedical and Health Informatics*, 20(2), 631-643. <https://doi.org/10.1109/JBHI.2015.2402280>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Automating Vector Autoregression on Electronic Patient Diary Data

Ando Celino Emerencia, Lian van der Krieke, Elisabeth H. Bos, Peter de Jonge, Nicolai Petkov,
and Marco Aiello, *Senior Member, IEEE*

Abstract—Finding the best vector autoregression model for any dataset, medical or otherwise, is a process that, to this day, is frequently performed manually in an iterative manner requiring a statistical expertise and time. Very few software solutions for automating this process exist, and they still require statistical expertise to operate. We propose a new application called *Autovar*, for the automation of finding vector autoregression models for time series data. The approach closely resembles the way in which experts work manually. Our proposal offers improvements over the manual approach by leveraging computing power, e.g., by considering multiple alternatives instead of choosing just one. In this paper, we describe the design and implementation of *Autovar*, we compare its performance against experts working manually, and we compare its features to those of the most used commercial solution available today. The main contribution of *Autovar* is to show that vector autoregression on a large scale is feasible. We show that an exhaustive approach for model selection can be relatively safe to use. This study forms an important step toward making adaptive, personalized treatment available and affordable for all branches of healthcare.

Index Terms—Electronic patient diary data, statistical software, time series analysis, vector autoregression (VAR).

I. INTRODUCTION

WITH the advances in portable consumer electronics, i.e., phones and tablets with internet access, the medical field has started using electronic patient diaries as an important means of collecting medical data. Electronic patient diary data are data entered by patients in a (web) application. The patient fills out a questionnaire using the application, and the results of the questionnaire are used as data points. Participating patients are asked to fill out the questionnaire either daily or at multiple times per day, at set intervals. Electronic patient diary data (also known as Ecological Momentary Assessments or Experience Sampling Method data) can accurately reflect the momentary state of various aspects of a patient. Analysis of this data can reveal that how the symptoms, emotions, and activity of an individual evolve over time, how they can be predicted, and which

factors contribute to the symptoms, allowing for an effective treatment.

A recent development in the medical field is to analyze electronic patient diary data using vector autoregression (VAR). VAR has its origins in the field of econometrics [1], and is typically used in analyzing and forecasting financial models [2]. In health care practice, the VAR technique provides insight into the temporal dynamics of variables related to symptomatology and functioning. This individual-based approach paves the way for tailoring treatment. VAR has recently been applied in the medical field to find cause-and-effect relations between symptoms using electronic patient diary data [3], [4]. These techniques allow for studying the temporal order of dynamic relationships among variables, which may provide concrete indications for intervention. For example, in psychosomatic research, VAR models can be used to determine, for individual patients, whether inactivity predicts depressive symptoms or whether depressive symptoms predict inactivity. Using VAR results, clinicians can thus derive whether a patient would benefit more from certain medication or from physical exercise.

The application of VAR models to analyze electronic patient diary data is not yet common practice. The main reason is that the construction of VAR models is a time consuming and complex process that requires a statistical expertise. The manual VAR modeling process can take a statistician several hours up to several days, for a single patient. Current available software solutions for automated VAR such as PcGive [5] are a step in the direction of automation but still rely heavily on the expertise of the user in configuring the program correctly, and they do not automate some of the key operations that a statistician might perform when working manually.

To simplify and speed up the VAR modeling process in a way that closely resembles how statisticians work, we developed *Autovar*. *Autovar* automates the process of finding optimal VAR models. *Autovar* is an open-source package written in the statistical programming language R and has a web application front end. *Autovar* finds and evaluates hundreds of potential models in seconds, selects those that are considered valid as determined by an array of tests, and further optimizes the discovered valid models by placing individual constraints. *Autovar* returns every discovered valid model, along with additional summary statistics, including *Granger causality* summary graphs (used for analyzing cause-and-effect relations between time series variables [6]), to provide a comprehensive and robust insight into the possible model space of a set of time series variables.

We modeled the approach of *Autovar* after how a statistician selects and finds VAR models. We identified key decision points in the modeling process, e.g., which statistical tests to perform at

Manuscript received July 29, 2014; revised January 29, 2015; accepted February 3, 2015. Date of publication February 10, 2015; date of current version March 3, 2016. This work was supported by The Netherlands Organization for Health Research and Development (ZonMW) under Contract 300020011.

A. C. Emerencia, N. Petkov, and M. Aiello are with the Department of Computer Science, University of Groningen, 9712 CP Groningen, The Netherlands (e-mail: a.c.emerencia@rug.nl; n.petkov@rug.nl; m.aiello@rug.nl).

L. van der Krieke, E. H. Bos, and P. de Jonge are with the Groningen University Medical Hospital, 9700 RB Groningen, The Netherlands (e-mail: j.a.j.van.der.krieke@umcg.nl; elske.bos@umcg.nl; peter.de.jonge@umcg.nl).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JBHI.2015.2402280

which time and how the results should be interpreted, adhering to best-practice guidelines, and encapsulated this knowledge in the program flow of our implementation. In essence, the approach of Autovar can be applied to datasets other than electronic patient diary data. However, since we modeled our approach after experts working on electronic patient diary datasets, some of the steps may be less suited for other domains.

Autovar advances the state of the art by being, to the best of our knowledge, the first demonstration to show fully automated VAR can work at scale. The innovation core of the present work is our novel approach for automating VAR, which leverages computing power to automate more steps of the process than what was done before, lowering the expertize required from users.

In this paper, we introduce our approach for automating VAR, and we explain the design and implementation of Autovar. We compare the performance of Autovar against VAR models manually constructed by experts, and we compare its features against those of other software used for automating VAR.

The rest of this paper is organized as follows. Section II provides a brief introduction to VAR. Section III explains our approach for automating VAR. Section IV presents an evaluation of our approach. Section V discusses the results and compares with related work. Section VI concludes the paper. An online appendix, detailing implementation-specific aspects of Autovar and its web application front end, is available via IEEE Xplore.

II. VECTOR AUTOREGRESSION

Time series data describe the measurements of a set of variables at successive points in time spaced by regular intervals. A VAR model can be specified as a set of equations that express linear dependencies among multiple time series variables [7, pp. 4–5]. Here, we explain VAR using a model with two variables, adapted from [8]. In the formulas below, Act and Dep refer to measurements of the two variables modeled in this example, activity and depression.

$$\begin{aligned} Act_t &= \alpha_0 + \sum_{i=1}^p \alpha_i Act_{t-i} + \sum_{i=1}^p \beta_i Dep_{t-i} + \zeta X_t + \epsilon_{1,t} \\ Dep_t &= \beta_0 + \sum_{i=1}^p \gamma_i Act_{t-i} + \sum_{i=1}^p \delta_i Dep_{t-i} + \eta X_t + \epsilon_{2,t}. \end{aligned} \quad (1)$$

A k -variable VAR model consists of k equations (in the above example, $k = 2$). An *endogenous variable* is a variable whose values are predicted by the VAR model. Thus, each of the k equations predicts the values of an endogenous variable in the model. The equations are parametrized by t , the index (or *time points*) of the time series data. The term p is the *lag order* of the system. A VAR equation predicts the value of an endogenous variable Y at time index t , based on previous values from all endogenous variables in the system, including Y itself, of up to p measurements before t . It is not hard to see that if we have n data points, we can predict $n - p$ values at most. Furthermore, in the following, we assume that there are no missing values in the time series data. The error terms ϵ are the *residuals* of the VAR model. These terms are strictly not part of the VAR equations. They merely denote the difference between the predicted values for the endogenous variables (e.g., Act'_t) and their actual values

Fig. 1. When the lag order $p = 2$ and the number of measurements $n = 7$, the number of predictions and residuals in a VAR model is 5.

(Act_t), such that for the first formula $\epsilon_{1,t} = Act_t - Act'_t$. As Fig. 1 illustrates, for n data points, we have $n - p$ residuals per variable. The lag order p is 2 because the model uses values of at most 2 measurements before t .

The formulas in a VAR model may also include variables that are not endogenous in the system. Such variables are called *exogenous variables*. In (1), X_t is an exogenous variable. We do not consider the exogenous variables to have lagged effects and, thus, we only include their *contemporaneous* values in our formulas, i.e., the values at time t .

A characteristic of VAR is that the contemporaneous effects of endogenous variables are not part of the model specification [7]. In other words, when a prediction for an endogenous variable at time t is based on an endogenous variable at time q , then $q < t$. This facilitates deriving Granger causalities between the endogenous variables.

In (1), the regression coefficients are the terms α_i , β_i , γ_i , δ_i , ζ , and η . A term is *constrained* or *restricted* when its regression coefficient is set to 0. Constraints are used to remove terms that do not contribute significantly to the prediction accuracy of the model. In our approach, each formula may have a distinct set of constraints. For example, some terms may be constrained in the predictions for Act_t that are unconstrained in predictions for Dep_t . We discuss the approach for setting constraints in more detail in Section III-E.

III. AUTOVAR

In Autovar, we mimic the way in which a statistician would manually perform the VAR model selection. There are different manual approaches to the VAR model selection. In our approach, we adhere to best practices such as those described in, e.g., Lütkepohl [7]. For example, our approach incorporates elements to favor simple models that explain more of the data.

There are a number of ways in which the approach of Autovar differs from statisticians working manually. Whenever a statistician would make a decision that cannot objectively be classified as correct in Autovar, we choose to exhaustively try all available options. For example, instead of using a lag order selection criteria to determine which lag order to use, in Autovar, we consider models from every lag order up to a specified maximum.

Following multiple execution paths instead of choosing one naturally leads to a situation wherein multiple models are under consideration. This is the main distinction between not only Autovar and the manual approach, but also between Autovar and other approaches to automated model selection [5], [9], which return one best model. Our approach does not discard

Fig. 2. Flow of information in Autovar.

any valid model found but ranks the returned models by model fit instead.

A. Overview

The different steps in the approach of Autovar are shown in Fig. 2. Autovar takes as input the time series data and some parameters. This input is used to determine an initial set of *model configurations*, which are specifications for creating a model. We then construct the VAR models based on their model configurations and assess their validity. If a model proves to be invalid, we may choose to modify some of its properties, and reassess several modified variations of the model. If a model was found to be valid, it is added to the results. For every valid model, we also include a constrained version in the results. Finally, we rank the valid constrained and unconstrained models by how well they fit the data, and present these models to the user, along with some summary statistics.

B. Model Configurations

Let a model configuration be defined as a set of parameters that specifies the terms to be included in the formulas of a VAR model, and as such, as a unique specification for a VAR model. Model configurations have a limited number of parameters that all have a limited number of values. Let the model configuration space define the combinatorial space of all possible models that Autovar can return. When searching for valid models, we limit the search to certain parts of this space, with other parts being invalidated by statistical reasoning or tests performed on the dataset.

Fig. 3 shows the six parameters that we use in model configurations.

1) *Trend Variable Inclusion*: When a time series linearly increases or decreases with time t , it is considered stationary around a trend [10]. Autovar employs the Phillips–Perron test [11] to determine whether a trend variable should be included. Throughout this paper, we use the canonical 5% level [12] (corresponding to a p -value ≤ 0.05) as criterion for determining statistical significance.

Fig. 3. Example model configuration.

			Dummy variables for weekdays			Dummy variables for day segments	
			Mon	Tue	Wed	AM	PM
Monday	t	0	1	0	0	1	0
	Act	8	1	0	0	0	1
Tuesday	2	7	0	1	0	1	0
	3	3	0	0	1	0	1
Wednesday	4	8	0	0	1	1	0
	5	5	0	0	1	0	1
	6	8	0	0	0	1	0

Fig. 4. Example showing cyclicity associated with day segments.

We run the Phillips–Perron test for each of the endogenous variables. We add a trend to all VAR equations of the model if for one or more of them the Phillips–Perron test is significant ($p \leq 0.05$) and the trend itself is significant. The Phillips–Perron results are always specifically calculated for each model configuration.

We consider only linear trends, which follow the definition of an exogenous variable $X_t = t$ for integer t with $1 \leq t \leq n$, n being the number of observations in the dataset.

2) *Dummy Variables for Days and Day Segments*: Time series with multiple measurements per day may exhibit cyclicity because events at the same time of day may correlate. For example, Fig. 4 shows a patient with increased depressive symptoms in the evenings. Likewise, time series data may show weekly cyclicity.

Seasonal dummy variables are exogenous variables that are added to a VAR model to account for cyclicity in the series. Seasonal dummy variables are called *dummy variables* because they are zero everywhere except for on specific time points, where their value is 1 [7, pp. 585].

In Autovar, we consider two types of seasonal dummy variables, those for day segments and those for weekdays. To the best of our knowledge, there is no reliable test to indicate whether any weekly cyclicity present would warrant the inclusion of weekday dummy variables in the models. Hence, Autovar explores both options for all otherwise distinct initial model configurations. To reduce the complexity of our approach, we choose to always include dummy variables for day segments in unrestricted models and, thus, their inclusion is not seen as part of the model configurations.

3) *Lag Order*: Recall from Section II that the lag order (or *lag length*) of a VAR model is defined by the highest lag used anywhere in the model. Adding more lags may invalidate a previously valid model, while any lag length on itself may result

Fig. 5. Decision chart for assessing a VAR model validity as implemented in Autovar. Shown here are the properties of valid models, the assumptions whose conjunction defines those properties, and the tests that evaluate those assumptions.

in a valid model. Statisticians working manually cannot feasibly search for valid models in all applicable lag lengths. They often limit their search scope to the lag lengths recommended by a certain lag order selection criteria [7, pp. 135].

We found that testing only the lags recommended by a lag order selection criteria, in practice, frequently results in a significant number of valid models being overlooked. In our approach, we circumvent this problem by choosing to search for VAR models for all lag lengths up to a specified maximum.

4) *Log Transforming the Data:* We define a *log-transformed model* as a model for the (natural) log-transformed dataset. If a log transformation is applied, it is applied to all endogenous variables in the model. A log transformation has a moderating effect on outliers, and can thus result in finding valid models for lag lengths, where there are no valid models without log transformation.

Statisticians working manually may choose to model log-transformed data only if they fail to find valid models without log transformation. However, to minimize information loss, in Autovar, we explore both options for all otherwise distinct initial model configurations.

Since log-transformed models are strictly models of a different dataset, we cannot directly compare their model fit with those of models without log transformation. For a fair comparison, in Autovar, we adjust the calculation of the log-likelihood for log-transformed models to negate the effect of the log transformation on the data (the net effect of this adjustment is to subtract from the log-likelihood the sum of the log-transformed data).

C. Model Validity

Fig. 5 shows a schematic overview for assessing the validity of a VAR model. While the properties and assumptions that define VAR model validity are widely recognized [7, pp. 157–212], the specific tests used to evaluate those assumptions may vary. Electronic patient diary datasets typically span between a few weeks and a few months, which is a level of variation that can be covered without having to change test functions.

We use four diagnostic tests in our approach, automating their evaluation and interpretation. One test evaluates the model stability (see Fig. 5, left). The other three tests (the *residual diagnostic tests*) evaluate whether the residuals meet the model

assumptions (see Fig. 5, right). We consider a model *valid* when it passes all four tests.

A VAR model is stable when all eigenvalues of its companion coefficient matrix lie inside the unit circle [7], [13], and this assessment is called the eigenvalue test.

The *white noise assumption* states that the residuals of a valid VAR model have serial independence [14], [15]. In Autovar, this assumption is evaluated using the Portmanteau test of Ljung and Box [16] on the residuals [7, p. 169].

The *homoskedasticity assumption* requires that the residuals of a valid VAR model are homoskedastic, i.e., that the variance is stable over time [17]. To evaluate this assumption, we perform the Portmanteau test on the squares of the residuals [18].

The *normality assumption* is evaluated using a Skewness–Kurtosis test [7], [19, p. 174].

D. Handling Invalid Models

When any of the four tests fail, the model is marked as invalid, and will not be included in the list of results. The actions performed when a model fails one of the tests depend on which property is being invalidated, and are described next. The result is typically that one or more variations of the model configuration are queued for assessment.

1) *When the Model is Not Stable:* Trend inclusion in Autovar is determined by the Phillips–Perron test for the initial model configurations. However, if the stability test for a model fails, we toggle the trend inclusion setting (meaning if there was a trend we remove it, and otherwise we add a trend) and queue the modified model configuration for assessment. This step is modeled after the iterative approach of statisticians working manually. If the modified model still fails the stability test, the model configuration is discarded.

2) *When the Model Fails Residual Diagnostic Tests:* When the residuals do not meet the model assumptions, depending on which test failed, a statistician working manually may choose to add more lags or to log transform the dataset. Since Autovar already considers all relevant lag lengths and log-transformed models, such a step is not needed.

Another strategy used by statisticians to solve assumption violation problems is to include special dummy variables in the model that allow residual outliers to be tuned individually [20]. As a result, residuals have fewer outliers and a higher chance of passing the homoskedasticity and normality tests.

We mimicked this process in Autovar. We designed a relaxation procedure that creates dummy variables based on outliers of the residuals of a model that failed the residual diagnostic tests. When we include these dummy variables in the failed model, the resulting model has an increased chance of passing the residual diagnostic tests. In the following, let *masking* an outlier denote including its index in a dummy variable that is 0 everywhere except on the time point of the outlier value.

When any of the three tests (shown in Fig. 5) evaluating the residuals fails, we may queue one or several variations of the model for assessment, each with dummy variables to mask distinct sets of outliers in the residuals of the variables failing one or more tests. When the equation still fails in the new model, we queue a model with increasingly more points masked in

outlier dummy variables, and perform up to three iterations of this procedure per VAR equation or until the equation passes the tests.

The reason for using multiple iterations of masking outliers is that choosing one particular threshold for masking outliers may not perform well on different datasets. Our procedure is modeled after the manual approach of statisticians who plot the residuals and try to add dummy variables for any extreme value. A common substitute for this method is the “factor times standard deviation (std) threshold” approach that we use here. Cousineau and Chartier [21] provide motivation for using specific thresholds. In some fields, it is common to use a threshold (or *factor*) 3.5, while in other fields 3.0 or 2.5 is more commonly used. Thus, in Autovar, we simply iterate over these three factors until we find a valid model. We start with fewer outliers (3.5), and add more outliers only if the tests for an equation keep failing (3.0 and 2.5). In order to favor models that explain more of the data, outliers are masked in dummy variables only if doing so is necessary to establish a model validity.

E. Constraining Valid Models

In the VAR model-fitting process, individual terms can be *constrained* (or *restricted*) per equation, effectively removing them. The goal of setting constraints is to obtain a model with better fit as measured by the Akaike Information Criterion (AIC [22]) or Bayesian Information Criterion (BIC [23]). These criteria include a penalty that scales with the number of estimated coefficients in the model. Thus, removing insignificant terms often improves a model fit. Autovar has the option to optimize either for lower AIC scores or for lower BIC scores (with lower scores indicating a better model fit); hence, in the following, we write *AIC/BIC* to denote whichever information criterion was chosen.

Since statisticians working manually cannot feasibly test millions of constraint configurations, several greedy approaches are used in practice [7, p. 206]. These algorithms have a time complexity of $O(n)$ or $O(n^2)$, with n the number of terms in the equations. For example, in a sequential elimination of regressors strategy [7, p. 211], the term with the highest p -value (i.e., the least significant term) is constrained in an iterative procedure that is ran until the AIC/BIC score no longer decreases. The validity of the model is assessed afterward. This approach uses no intermediate validity testing. The approach is based on the assumption that terms that do not contribute significantly to the model may be removed as long as the model fit improves as a result.

While the approach used for setting constraints in Autovar is similar to the sequential elimination of regressors strategy described earlier, we developed and implemented improvements that result in lower AIC/BIC scores. First, because models to be constrained are initially valid, we may impose the assertion that the resulting constrained models should always be valid as well. We follow a greedy approach and constrain the term with the highest p -value as long as the resulting model remains valid and the AIC/BIC score does not increase. Like other greedy approaches, ours is not guaranteed to always find the best

Algorithm 1: The GetValidModels algorithm.

Input: data set D , parameters P (min. lag and max. lag).
Data: functions `evaluate_var_model`, `stability_test`, `portmanteau_tests`, and `skewness_kurtosis_test`.
Output: list of $\langle \text{configuration}, \text{model} \rangle$ tuples, representing the valid models found.

```

 $Q \leftarrow \text{INITIALMODELCONFIGURATIONS}(D, P)$ 
 $R \leftarrow \text{empty list}$ 
 $S \leftarrow \text{empty set}$ 
while  $Q$  is not empty
     $M \leftarrow Q.\text{pop}()$ 
     $B \leftarrow \text{evaluate\_var\_model}(D, M)$ 
     $A \leftarrow \text{TRUE}$ ,  $T \leftarrow \text{FALSE}$ 
     $O \leftarrow \text{empty set}$ 
    if stability_test( $B$ ) fails
        then  $A \leftarrow \text{FALSE}$ ,  $T \leftarrow \text{TRUE}$ 
    if portmanteau_tests( $B$ ) fails
        then  $\begin{cases} A \leftarrow \text{FALSE} \\ \text{for each variable } V \text{ that failed} \\ \text{do insert } V \text{ in } O \end{cases}$ 
    if skewness_kurtosis_test( $B$ ) fails
        then  $\begin{cases} A \leftarrow \text{FALSE} \\ \text{for each variable } V \text{ that failed} \\ \text{do insert } V \text{ in } O \end{cases}$ 
    if  $A$ 
        do  $\begin{cases} \text{add } \langle M, B \rangle \text{ to } R \\ \text{if not } M.\text{restrict} \\ \text{then } \begin{cases} N \leftarrow \text{copy}(M) \\ N.\text{restrict} \leftarrow \text{TRUE} \\ \text{add } N \text{ to } Q \end{cases} \end{cases}$ 
    if  $T$ 
        then  $\begin{cases} N \leftarrow \text{copy}(M) \\ N.\text{trend} \leftarrow \neg N.\text{trend} \\ \text{if } N \notin S \\ \text{then insert } M \text{ in } S, \text{ add } N \text{ to } Q \end{cases}$ 
    for each set  $W \in \{\mathcal{P}(O) - \emptyset\}$ 
         $N \leftarrow \text{copy}(M)$ 
        for each variable  $V \in W$ 
            do if not  $N.\text{outliers}.V = 3$ 
                then  $N.\text{outliers}.V++$ 
        if  $N \notin S$ 
            then insert  $N$  in  $S$ , add  $N$  to  $Q$ 
return ( $R$ )

```

constraints. Second, when constraining the term with the highest p -value is not possible (either because it invalidates the model or because it increases AIC/BIC scores), we continue with the term with the second-highest p -value, and so on. This step causes the constraint-setting algorithm to have quadratic time complexity. However, it does frequently result in better constraints (we refer to Section IV-A for a comparison) and guarantees model validity since the initial models are valid and validity is asserted in every step.

F. Algorithm for Model Selection

We now present the main procedure for selecting valid models in Autovar. The `GetValidModels` function (see Algorithm

1) returns an unordered list of valid VAR models and their configurations, given a dataset and other input parameters. The parameter options P specify the minimum and maximum lag order to consider. If zero-order lag models should be included, minimum lag order $P.min_lag$ is 0, otherwise it is 1.

Algorithm 2: The Initial Model Configurations algorithm.

Input: data set D , parameters P (variable names, max. lag, etc.).
Data: function `phillips_perron`.
Output: queue of tuples of model parameters.
 $Q \leftarrow$ empty queue
for each $l \in [P.min_lag, P.max_lag]$ **do**
 for each $t \in \{FALSE, TRUE\}$ **do**
 for each $d \in \{FALSE, TRUE\}$ **do**
 $\langle$ lag = l ,
 apply_log_transform = t ,
 include_day_dummies = d ,
 add restrict = FALSE,
 outliers = NULL,
 trend = `phillips_perron`(D, t, l)
 to Q
return (Q)

In the first step of the algorithm, we initialize the model configuration queue Q to contain an initial set of model configurations based on the dataset D and given parameters P . These initial model configurations are returned by the InitialModelConfigurations function shown in Algorithm 2. This algorithm returns a queue of initial model configurations for the given parameters. It contains model configurations of lags up to the given maximum lag, with and without weekday dummy variables (if applicable), and with and without log transformation. For each model configuration, the `trend` parameter, which signifies the inclusion of a trend variable in the model, is set according to the Phillips–Perron test as explained in Section III-B1. Furthermore, dummy variables for day segments are included in each model (see Section III-B2).

Returning to Algorithm 1, we initialize R , our return variable, and S , a set to keep track of the model configurations that have been tested so far. We use this set to ensure that we do not evaluate models more than once. We loop through the main body as long as there are model configurations to be tested. We evaluate each model configuration M popped from the queue Q to create a model B .

We proceed to introduce two state flags in the loop body. The variable A is true as long as we consider the model B to be valid. The variable T becomes true when the stability test fails. The set O keeps track of the names of the variables that failed at least one of the residual diagnostic tests.

We first test the stability of the model B using the eigenvalue test. If the model fails the test, we set A to false to denote that the model is invalid. We also set T to true to consider toggling the trend inclusion later on.

The function `portmanteau_tests` runs the Portmanteau test on the residuals (the white noise assumption) and on the

squares of the residuals (homoskedasticity assumption). Each variable V that fails either of these tests is added to the set O . Furthermore, if any variable fails either of the two tests, we set A to false to denote that the model is invalid.

The function `skewness_kurtosis_test` evaluates the skewness and kurtosis of the model. The model is invalidated (A set to false) if the residuals of any VAR equation show significant skewness or kurtosis. The offending variables are inserted in the set O .

After running the tests, we check whether A is still true to determine if the model passed all tests. If the model passed all tests, we consider it to be valid and add it to the return variable R in a tuple with its model configuration. In addition, if the model was unrestricted, we queue a copy of the model configuration with the `restrict` flag set to true to denote that this is a valid model configuration for which we should try to find constraints. Constraints are set as part of the functionality of the `evaluate_var_model` function (explained in Section III-E). Moreover, recall that constrained models remain valid and, thus, T will never be true and O will always be empty for restricted models.

Next, we check if T is true. Recall that T is true if and only if the stability test failed. In this case, we toggle the inclusion of the trend variable in the model configuration and add the new model configuration N to the queue Q . To ensure that we only toggle the inclusion once, we first check whether N is not in the processed set S . If it is not in this set, we add the original model M to this set S . Note that it is not necessary to add N to this set.

The final for-each statement is for queueing model configurations with more outliers masked in dummy variables for variables that failed at least one of the residual diagnostic tests. We consider all combinations for decreasing the outlier threshold by 0.5 for each failing variable. This number of combinations is $2^f - 1$, with f the number of failing variables and is signified by the power set of O minus the empty set. Recall that we use three levels for thresholding outliers into dummy variables, maintained separately per variable. These levels are used in the `evaluate_var_model` function to add outlier dummy variables to the model.

IV. EVALUATION

Here, we evaluate the practical and theoretical performance of our approach.

A. Comparison With Manual Analysis and PcGive

We compare Autovar to experts working manually and to PcGive, a commercially available program for automating VAR, with respect to the model fit and the model validity.

1) *Dataset:* The dataset consists of a sample of 20 patients with multiple, persistent functional somatic symptoms (FSS). Electronic diaries were used to collect the times series data on stress and FSS. The data were collected between January 2004 and February 2006. The data were preprocessed to yield one measurement per day, resulting in an average of 86 measurements per patient (max. 100, std. 6.58).

The patients helped to identify their three most severe, applicable, or frequent symptoms from the following list: muscle pain (Musc), joint pain (Join), back pain (Back), headache (Head), abdominal pain (Abdo), pelvic pain (Pelv), bowel symptoms (Bowe), dyspepsia (Dysp), nausea (Naus), tight throat (Tigh), chest pain (Ches), weakness (Weak), numbness (Numb), and palpitations. This dataset was collected by Burton *et al.* who provide a full description of how each symptom was measured [24].

2) *Setup*: For each patient, three bivariate datasets were constructed, each one using Stress (Stre) as one of the endogenous variables and one of the three FSS symptoms selected by the patients as the other. Missing data were previously imputed for each individual dataset using the expectation maximization function in SPSS 20. None of the approaches use dummy variables for day segments since there is only one measurement per day.

3) *Manual Analysis*: The manual approach we are comparing to was performed by van Gils *et al.* [25] using STATA 11. We believe that comparing their models against those of Autovar is fair because both approaches use the same diagnostic tests to assess the validity of models.

The manual approach first includes both a linear trend variable and weekday dummy variables, and then removes those that are not statistically significant. The lag order of the model is determined by majority voting of several lag length selection criteria. Specific measures were taken to improve the model, depending on which assumptions of the model were violated according to the diagnostic tests. Residual autocorrelation was solved by including higher lags. Heteroskedasticity and skewness were solved by using a log transformation on the endogenous variables. If the nonnormality merely stemmed from a few outliers, then dummy variables masking outliers at $3 \times \text{std}$ of the residuals were used. Statistically insignificant terms were pruned from the estimated models in descending order of p as long as the BIC score did not increase. No diagnostic tests were performed at intermediate steps when placing constraints.

4) *Autovar Analysis*: Autovar used the same parameters for every patient dataset. The maximum lag length was set to 3 (which is our default value if we do not know anything about the data) and zero-lag models were included. Like the manual approach, constraints were chosen to optimize for low BIC scores. Each dataset was timestamped, allowing Autovar to derive and include seasonal dummy variables. All other settings were left at their default value. If a run returned no valid models, Autovar was called a second time, with identical parameters except with maximum lag at 7 instead of 3 and the lowest factor for masking outliers at $2.5 \times \text{std}$ instead of 3 and including outliers of the squared dataset.

5) *PcGive Analysis*: PcGive does not use the same set of validity tests as the other two approaches; hence, we only report the BIC score of its model and whether it passed its own set of validity tests. The PcGive models were derived using its multiple-equation dynamic modeling function. We provided PcGive with the same number of lags that Autovar required for a valid model, down to a minimum of 3 and including the

zero lag. We also included a constant term and a trend variable. The model type was set to an unrestricted system, and automatic model selection was set at standard target size. Outlier and break detection was set to “large residuals.” All other settings were left at default. For this comparison, we used PcGive 14, which was released in June 2013. We provide further details on the PcGive approach in Section V.

6) *Comparison*: Table I shows a comparison of the best models found between Autovar, the manual approach, and PcGive. Note that Autovar always returns multiple models, but this table only shows the results of the best model of each approach. The rows are the datasets. The left column identifies the dataset. The number identifies a patient. For each patient, three datasets are analyzed, each having two endogenous variables, stress and one other FSS symptom indicated by the patient. The remaining columns show the details of the model of Autovar with the lowest BIC score (columns 2–6), the final model obtained in the manual approach (columns 7–11), and the final model returned by PcGive (columns 12–13). The Exogenous variables column denotes which exogenous variables are used in the selected models. The variable Nr denotes the linear trend variable. The variables Mon, Tues, Wed, Thurs, Fri, Sat, and Sun denote dummy variables for the respective weekdays. BIC scores for log-transformed models were compensated (by subtracting the sum of the log-transformed variables from the log-likelihood score) so that they can be compared to those of nonlogtransformed models fairly. Individual numbers denote time points included in exogenous dummy variables for residual outliers. Per row, the best (lowest) BIC score is printed in boldface.

Both Autovar and the manual approach use the same diagnostic tests. Table I has a “pass all tests” column denoting if a model passes all diagnostic tests. Since models returned by Autovar always pass all diagnostic tests, if the value in this column is “No,” it means Autovar returned no models and the rest of the row is left empty. In the manual approach, if the experts found a model for which they considered the violation of the assumptions not severe enough as determined by manual inspection of the histograms of the residuals, they proceeded to use that model for their analysis. The “pass all tests” column uses boldface to denote that the model passes all diagnostic tests when the models of the other approaches did not. For PcGive, we note that its manual states that validity is guaranteed only if the given unrestricted model is valid.

7) *Discussion*: For the datasets used in this experiment, we find that that Autovar performs best with respect to the BIC scores and the number of valid models found (see Table I). Autovar finds the lowest BIC score for 33 datasets (55%), followed by PcGive (18 datasets, 30%), and the manual approach (nine datasets, 15%). Autovar finds a model that passes all its diagnostic tests for 57 of the 60 datasets (95%) compared to 27 (45%) for the manual approach, and seven (12%) for PcGive. Since PcGive uses a different set of validity tests, the remainder of this section focuses on comparing Autovar with the manual approach.

There were 18 datasets (30%) where the best model found by the Autovar and the manual approach differed with respect

TABLE I
COMPARISON OF BEST MODELS FOUND BY AUTOVAR VERSUS MANUAL ANALYSIS VERSUS PCGIVE 14

Dataset	Autovar					Manual					PcGive 14	
	pass all tests	lag order	log transform	Exogenous variables	BIC	pass all tests	lag order	log transform	Exogenous variables	BIC	pass all tests	BIC
33 Stre Bowe	Yes	3	No	Nr, Mon, Tues, Fri, 68, 83	1475.797	Yes	1	No	41, 68, 83	1497.361	No	1491.905
33 Stre Musc	Yes	3	No	Mon, Tues, Wed, Fri, 41, 68	1424.141	Yes	2	No	41, 68	1450.193	No	1444.139
33 Stre Naus	Yes	2	Yes	Mon, Tues	1422.735	Yes	1	Yes	–	1436.123	No	1459.204
35 Stre Musc	Yes	3	No	Tues, Thurs, 36, 42	1296.764	Yes	1	No	36, 42	1325.821	No	1308.909
35 Stre Head	Yes	3	No	36, 42	1397.385	No	1	No	36, 42	1429.595	No	1399.030
35 Stre Bowe	Yes	1	No	Nr, 11, 36, 42	1275.545	No	1	No	36, 42	1288.66	No	1259.342
36 Stre Bowe	Yes	1	No	Nr, 50	1277.77	Yes	1	No	50	1286.525	No	1251.550
36 Stre Join	Yes	2	No	Tues, 50	1209.422	Yes	1	No	3, 50	1229.236	No	1197.215
36 Stre Head	Yes	3	No	Tues, 50	1138.817	Yes	1	No	2, 50	1187.865	Yes	1147.795
38 Stre Musc	Yes	3	No	–	1191.668	Yes	2	No	Sun, Mon, Thurs, Fri	1196.786	Yes	1193.703
38 Stre Pelv	Yes	6	No	Mon, Fri, 81	1140.047	Yes	4+11	No	Mon, 81	1078.198	Yes	1146.002
38 Stre Dysp	Yes	2	No	–	1262.886	Yes	1+11	No	Nr, Mon, Sat	1112.852	Yes	1260.316
40 Stre Musc	Yes	3	No	Nr, 33, 40, 47	1191.358	Yes	3	No	Nr, 33, 40, Mon	1198.033	No	1209.986
40 Stre Dysp	Yes	2	Yes	Nr, 16	1140.603	No	2	No	Nr, Mon, 8, 33, 64	1112.703	No	1122.509
40 Stre Tigh	Yes	1	Yes	2, 26, Mon, Tues, Fri	1290.652	No	3	No	Mon, Fri, 17	1238.909	No	1236.806
42 Stre Musc	Yes	3	Yes	Nr, Mon, Tues, Wed, Thurs, Fri	1393.393	No	2	No	Nr, Sat, Sun, 84	1427.589	No	1423.641
42 Stre Dysp	Yes	3	Yes	Sun, Mon, Tues, Wed, Thurs, Fri	1231.745	No	1	No	Sat, Sun, 6, 72, 78, 84	1299.618	No	1306.122
42 Stre Head	Yes	3	Yes	Mon, Tues, Wed, Thurs, Fri	1437.51	Yes	3	No	Sun, Mon, Fri, Sat, 12, 84	1491.578	No	1502.099
44 Stre Bowe	Yes	3	Yes	Nr	1476.624	No	5	No	Sat, Sun, 15, 21	1437.3	No	1474.465
44 Stre Join	Yes	3	Yes	35	1538.723	Yes	2	Yes	–	1561.128	No	1513.972
44 Stre Head	Yes	3	Yes	–	1473.467	No	1	No	6, 15, 21, 69	1503.363	No	1491.802
45 Stre Musc	Yes	3	Yes	6, 61	1274.814	Yes	3	Yes	6, 61	1286.713	No	1348.999
45 Stre Abdo	Yes	3	Yes	Nr	1216.919	Yes	1	Yes	Nr	1231.142	No	1320.165
45 Stre Dysp	Yes	1	Yes	Nr	1258.349	Yes	1	Yes	Nr	1268.778	No	1337.668
46 Stre Join	Yes	5	Yes	Tues, 38, 43, 46, 61	952.962	No	1	Yes	2, 5, 7, 43, 48, 62	936.2602	No	1061.963
46 Stre Abdo	No					No	1	Yes	Sun, 2, 5, 7, 43, 48, 62	952.7747	No	1077.846
46 Stre Ches	Yes	7	Yes	Sun, Mon, Tues, Wed, Thurs, 22, 43	663.091	No	1	Yes	Nr, 2, 5, 7, 43, 48, 62	659.5551	No	679.200
48 Stre Join	Yes	2	No	–	1415.307	Yes	2	No	–	1418.621	No	1395.512
48 Stre Musc	Yes	2	No	–	1415.563	Yes	2	No	–	1417.924	No	1400.062
48 Stre Abdo	Yes	2	No	Fri	1447.276	Yes	2	No	–	1454.208	No	1425.590
49 Stre Bowe	Yes	6	No	Nr, Wed, Fri, 35	1423.461	No	1	Yes	Nr, Sun, Mon, Tues, Thurs, Sat, 53, 58, 70	1476.034	No	1453.757
49 Stre Musc	Yes	3	No	Nr, Fri, 35	1430.609	No	1	No	Nr, 35	1467.913	No	1436.234
49 Stre Join	Yes	3	No	Nr, Mon, Tues, Thurs, Fri, 35	1417.059	No	2	No	Nr, Mon, 35	1440.792	No	1416.620
52 Stre Join	No					No	1	Yes	Nr, Sun, Mon, Tues, Sat, 10, 26, 53	1095.838	No	1016.793
52 Stre Pelv	No					No	2	Yes	Nr, Sat, 10, 26	1143.362	No	1071.698
52 Stre Naus	Yes	2	Yes	Nr, Sun, Mon, Tues, Wed, Fri, 10, 26	1063.162	Yes	2	Yes	Nr, Mon, Thurs, Fri, Sat, 10, 26, 53	1065.268	No	1088.074
53 Stre Naus	Yes	6	Yes	Mon, Tues	1426.306	No	2	Yes	–	1454.526	No	1505.249
53 Stre Musc	Yes	5	Yes	–	1272.809	No	4	No	39, 55, 80	1308.432	No	1293.011
53 Stre Numb	Yes	7	Yes	–	1387.167	No	3	Yes	Nr	1462.394	No	1444.496
54 Stre Abdo	Yes	7	No	Nr, Sun, 18, 27, 37, 42	1352.274	No	1	Yes	Nr, Sat, 21, 27, 42, 53, 71	1461.412	No	1364.378
54 Stre Musc	Yes	2	No	Tues	1439.613	Yes	1	No	Nr, Tues, Sat	1480.292	Yes	1430.030
54 Stre Tigh	Yes	2	No	–	1424.033	Yes	2	No	Sat	1426.254	Yes	1405.171
56 Stre Join	Yes	3	Yes	Thurs	1379.798	No	1	No	Nr, Thurs, 35, 43	1380.596	No	1353.073
56 Stre Head	Yes	7	No	Nr, Wed, Thurs, 43	1342.21	No	4	No	Nr, Mon, Thurs, 35, 46	1393.287	No	1334.264
56 Stre Weak	Yes	3	Yes	Mon, Tues, Thurs, 8, 59	1403.717	No	1	Yes	Thurs, 8	1415.349	No	1385.307
57 Stre Musc	Yes	1	Yes	Nr, Thurs, 38, 50, 90	1160.214	No	6	Yes	Nr, 38, 50, 90	1135.99	No	1178.149
57 Stre Bowe	Yes	5	Yes	Nr, 38, 50, 67, 90	1133.843	No	1	Yes	Nr, 38	1189.526	No	1147.513
57 Stre Weak	Yes	2	Yes	Nr, 38, 50, 90	1221.747	No	1	No	Nr, 38, 90	1326.232	No	1295.919
58 Stre Bowe	Yes	3	No	Nr	1508.038	No	2	No	Nr, Mon, Tues, Thurs, Fri, Sat	1531.478	Yes	1522.983
58 Stre Join	Yes	2	No	Nr, Sun, Mon, Tues, Wed, Thurs, Fri	1475.186	No	1	Yes	Nr, Sat	1486.895	No	1462.348
58 Stre Back	Yes	2	No	5	1439.488	No	1	No	Nr, Sat, 2, 5	1463.024	No	1428.737
60 Stre Abdo	Yes	3	Yes	Nr	900.25	Yes	7	No	Nr, Tues, Sat, 29	855.2571	No	903.296
60 Stre Tigh	Yes	1	Yes	Nr, 85	863.853	No	1	No	Nr, Tues, Fri, 15	942.1793	No	869.829
60 Stre Head	Yes	1	Yes	Nr	889.988	No	1	No	Nr, Tues, Fri	1038.374	No	960.950
63 Stre Musc	Yes	5	Yes	Sun, Tues, 14, 20, 47, 70, 74, 77, 85, 86	1307.663	No	3	Yes	Sun, 14, 20	1356.493	No	1336.695
63 Stre Abdo	Yes	3	Yes	Sun	1291.64	Yes	1	Yes	Sun	1309.721	No	1499.632
63 Stre Head	Yes	3	Yes	Sun	1434.434	Yes	3	Yes	Sun, Fri	1444.03	No	1551.692
64 Stre Abdo	Yes	2	Yes	–	1426.985	Yes	2	Yes	–	1442.928	No	1585.819
64 Stre Musc	Yes	1	Yes	Mon, Wed	1626.654	Yes	1	No	Tues, Fri, Sat, 8, 26, 42	1675.961	No	1635.979
64 Stre Ches	Yes	2	Yes	Thurs, 45, 57	1676.675	No	2	No	Nr, 8, 26	1752.614	No	1714.988

to applying a log transformation. Of the remaining 39 datasets where both approaches found a model, there are 34 instances (87%), where Autovar had a lower (better) BIC score than the manual approach, and five instances (13%) where the manual approach had the lower BIC score.

For the 27 datasets for which both Autovar and the manual approach found a valid model, there are three cases (11%) where Autovar favors a log-transformed model, while the manual approach favors a model without log transformation. Cases where a valid model of the manual approach favored a log transformation while Autovar did not, did not occur. For the remaining 24 datasets where both approaches used the same log-transformation setting, Autovar had the lower BIC score 22 times (91.7%) compared to two times (8.3%) for the manual approach. In both instances, where the manual approach had the lower BIC score, this was due to using a high lag (11) that is outside the search range of Autovar.

Surprisingly, in all five instances (18.5% of 27) where the lag order, log transform, and exogenous variables are identical for Autovar and the manual approach, Autovar still reached a lower BIC score because of a difference in the constraints used. In these cases, Autovar has one or two different constraints that result in a slightly lower BIC score. These results suggest that the added complexity of our constraint-finding method in practice may frequently result in better constraints. Another surprising result is that the built-in preference of Autovar for favoring models with fewer masked outliers did not result in significantly higher BIC scores on average.

While not shown in the results, we note that in 21 out of 27 cases (77.8%) where both Autovar and the manual approach find a valid model, Autovar also found a model at the same lag order and with the same log-transform setting as the manual model (with the only differences being in the exogenous variables and the constraints). One of these cases (42 Stre Head) was the only tested case where setting a constraint that invalidates the model would result in a valid model (with a lower BIC score than the solution of Autovar) by adding more constraints. This finding supports our implicit assumption that such constraint combinations occur infrequently in practice. Reasons for Autovar not finding certain models are due to the manual approach using higher lags or different outliers (i.e., there is one instance where a mistake was made in calculating the set of outliers in the manual approach which resulted in a valid model). The number of valid models missed because of constraining only valid models is not reflected in these results, as both approaches applied constraints only to valid models.

8) *Clinical Interpretation:* As an example of the clinical interpretation of the learned models, we take a closer look at one of the datasets used in our evaluation, 57 Stre Bowe. The most interesting part for clinicians would be the Granger causality summary as reported by Autovar. For this particular dataset, it shows the following as part of its output:

Granger causality summary of all 12 valid models :

66.67% Bowe Granger causes Stre (8 models)

33.33% < None > (4 models)

This output indicates that in 8 of the 12 valid models Autovar found, there is one Granger causality relation found. This gives reasonable confidence to assume that this relation is indeed present in the dataset. These summary results are more reliable than deriving conclusions from any individual model, since any individual model may have passed the tests by chance.

The above information can be interpreted as stating that an increase/decrease in bowel symptoms causes an increase/decrease in stress. Since by the validity assumptions, we know that this relation is significant, our Autovar output could suggest to a clinician that it is more effective to treat for bowel symptoms than to treat for stress symptoms, since the latter are at least partially caused by the former. Without having a tool to find these relations, it would be very hard for a clinician, or very time costly for a statistician, to deduce the right conclusions from the electronic diary data of a patient.

B. Performance

Next, we consider aspects of time complexity, memory complexity, and scalability of our approach.

1) *Time Complexity:* The minimum number of models evaluated by Autovar is $O(4l)$, where l is the number of lags to consider, i.e., $\max_lag - \min_lag + 1$. The factor $4 = 2^2$ follows from considering at most 2 options for applying a log transformation and 2 options for including weekday dummy variables. In the worst case, if the stability test fails for all initial models, we need to evaluate twice this number of models. In addition, for each of the stable models, we may need to evaluate an additional set of models depending on the outcome of the residual diagnostic tests. Thus, the total number of models that is evaluated is $O(4l + 4l4^k)$, with k the number of endogenous variables, since we may need to consider all possible subsets of outliers for up to three iterations. Since we are estimating a VAR model in every step, which is a costly operation, k cannot be too large. Adding one endogenous variable to the system will cause Autovar to take about four times as long to evaluate all models. We have tested Autovar with $k = 2$ and $k = 3$, and it typically runs between 1 and 3 s for $k = 2$ and up to a minute for $k = 3$, measured as a single-threaded run time on an i7 PC at 3.5 GHz. We have not tested Autovar with $k \geq 4$.

The maximum number of valid models returned by Autovar is $O(8l4^k)$. The derivation of this bound follows the reasoning above, and taking into account that for every valid model, we also return a constrained version. The different iterations of outliers are often mutually exclusive, so the full 4^k subsets of models will rarely, if ever, all be estimated. In practice, we of course find that the number of valid models returned by Autovar is far lower. For example, for the datasets shown in Table I, where $k = 2$ and $l = 8$, the average number of valid models returned by Autovar per dataset is 8.07 with a std of 5.17 and a maximum of 27.

A significant portion of the running time is spent on finding constraints for the valid models found. Following the above reasoning, we find that an upper bound on the number of models to be restricted is $O(4l4^k)$. Recall from Section III-E that the constraint-setting procedure has $O(n^2)$, with n the number of

Fig. 6. Encoding a model configuration as an integer number. A total of $2k + 8$ bits (with k the number of endogenous variables) is required to distinguish between all possible model configurations.

terms in the equations. Since there are k equations, the number of terms in the equations is k times the number of terms in one equation. In the unconstrained models, each of the k endogenous variables appears with all its l lags in each equation. It follows that the total number of terms in an unconstrained model is $O(lk^2)$. With an $O(n^2)$ complexity for setting constraints, in the worst case, we perform $O(l^2k^4)$ the full VAR model estimations for every valid unconstrained model. To put these numbers in perspective, for, e.g., a model with $k = 3$, $l = 6$, and having found three valid unconstrained models, we spend around half the running time on constraining the three valid models found and the other half on assessing the validity of all models under consideration.

2) Memory Complexity: Our approach requires the implementation to retain a list of all model configurations in memory. We need to distinguish between two options for applying a log transformation, two options for including weekday dummy variables, two options for applying restrictions, and two options for trend variable inclusion. In addition, we need to encode the lag order of the system, and the iterations for masking outliers for the k endogenous variables.

Fig. 6 shows how model configurations can be represented as integer numbers. The iterations for masking outliers for the different equations can be encoded as a 2 bit number because the iterations range from 0 to 3, inclusive. For a system with two variables, we find that $2 \cdot k + 8 = 2 \cdot 2 + 8 = 12$ bits are needed to represent each possible model configuration. If we encode model configurations as numbers indexing into a Boolean array, this array would need to have a size of $2^{12} = 4096$. If we assume that 1 B of memory is used per element in a Boolean array, when $k = 2$, retaining the “processed” state of all model configurations requires 4 kB of memory. However, to accommodate debugging, our implementation in R is less space efficient.

To generate its output, our approach also needs to retain the valid VAR model estimations in memory. From the time complexity analysis, we know that our approach finds $O(8l^4k)$ valid models. The size of the estimated models is implementation dependent and varies in practice, but includes at least the coef-

ficients of the terms of the formula. On the assumption that the storage size for a model estimation grows linearly in relation to the number of coefficients in the model, the memory size for a model estimation scales with $O(lk^2)$.

3) Scalability: For finding and outputting models for all 60 datasets of Section IV-A on an i7 PC at 3.5 GHz, Autovar required around 25 min single-threaded execution time in total. This does not include the approximate 10 min that the authors needed to write an R script to process all datasets in sequence using Autovar. In comparison, the analysis of the experts working manually required several working days.

While not exploited in the current implementation of Autovar, our approach for constructing and evaluating VAR models (see Algorithm 1) allows for parallelization. The conditions are that all access to queue Q and result list R must be synchronized by mutual exclusion. If each initial model configuration and the variations thereof were to be executed in parallel (requiring at least $4l$ processors), then assessing the validity of all models takes $O(1 + 4^k)$ time. If we may assume that $k \leq 3$, assessing the validity of all models can be performed in constant time, with a constant factor of at most 65 VAR model estimations per processor. However, reducing the complexity of or introducing parallelization to the constraint-setting procedure is more difficult and remains a bottleneck in our approach. Even if all valid models were constrained on different processors, each processor would still have to perform an $O(l^2k^4)$ full VAR model estimations.

V. RELATED WORK

The findings of the current study are consistent with those of Hendry and Krolzig [5], who found that automatic modeling techniques can perform on a competitive level with experts working manually. However, previous work warns for an approach based on “data mining” for models as it could potentially lead to random models passing tests by chance [26]. This issue applies to Autovar as well. However, the relatively low number of models that Autovar evaluates on average combined with the low probability of a random model passing all three tests render it unlikely that any random models passed the tests for the datasets we tested on. Autovar performs three tests at a 0.05 significance level, and if we were to assume that all three tests are independent, then there is a probability of $0.05^3 = 0.0125\%$ of a model randomly passing all three tests. That translates into evaluating 8000 models on average before we expect to see one random model passing all tests. For the datasets of Table I, the maximum number of distinct models we tested for any particular dataset was 237 (with an average of 63.8). However, if we assume a worst-case scenario in which two of the three tests are fully statistically dependent, the probability of a model passing all tests randomly becomes $0.05^2 = 0.25\%$ or 1 in 400 models, which makes the event more probable. This is one of the reasons why Autovar returns not one best model, but all valid models found, along with summary statistics to show the user which model configuration settings are common among the valid models. Returning multiple valid models instead of just one is one of the main distinctions between Autovar and other

approaches to automated model selection. We consider it to be one of its main contributions because a list of all valid models found for a dataset grants more insight into the properties of the valid models than a single model does. For example, if we want to determine whether a certain Granger causality is present in a dataset, an approach that returns a single model could only base its answer on the relations found in that model, while Autovar can average over all valid models found and answer in the form of a probability.

A. PcGive

Here, we present a comparison of the functionality of Autovar to that of PcGive. To the best of our knowledge, PcGive (previously PcGets [26]) is currently the only other software that can perform a fully automated VAR model fitting. RETINA [9] is another known implementation for automated model selection but is not suited for VAR. Other software exists for modeling VAR, e.g., Eviews, Mathematica, MATLAB, TSP, GAUSS, gretl, SHAZAM, R (also available in sage and S-PLUS), LIMDEP and NLOGIT, Stata, RATS, and Microfit, but these programs do not feature an automated model selection. There are, however, frameworks that provide a theoretical basis for an automated approach to model selection. Pesaran and Timmermann [27] describe a nonsequential approach with specific-to-general aspects [26], and Phillips provides the basis for a Bayesian framework for an automated model selection [28].

We compare the functionality of Autovar and PcGive in Table II. This table compares features and functionality (left column) of Autovar (middle column) to those of PcGive (right column). The automatic data imputation for missing values are described in Appendix A.

B. Discussion

From Table II, we see that PcGive is a more extensive software suite. It supports not only VAR modeling but various other statistical models as well. Furthermore, it not only finds models but can also apply them, for example, in forecasts and impulse response simulations. Autovar, on the other hand, is easier to use and incorporates more automation. It features automatic creation and inclusion of seasonal dummy variables for weekdays and day segments, of trend variables, of log transformations of the data, and of constraints specific per VAR equation. These aspects of automation make Autovar easier to use because in most cases a user can just access the web application, upload a dataset, select the VAR columns and click “Run.” With respect to configurability, PcGive favors an approach of extensive configurability that relies on the expertize of the user in specifying the proper settings, while Autovar prefers an approach of automatically trying to determine which settings to use for a dataset, having embedded the expertize in its algorithms for finding models. Another important distinction is that Autovar discards any models that fail any of the tests, while PcGive always finds and returns a best model, even when it is not valid.

TABLE II
COMPARING THE FUNCTIONALITY OF AUTOVAR AND PCGIVE

	Autovar	PcGive 14
Approach	Exhaustive search restricted by statistical tests.	General-to-specific modeling strategy.
Model-selection results	Multiple valid models.	A single best model.
Additional results	Granger causality summary, Contemporaneous correlation summary, model configuration summary statistics, plots of input variables, test results.	Test results for the model returned, plots of input variables, forecasts, simulation and impulse response, dynamic analysis, cointegration tests.
Max. lag setting	Yes	Yes (set per variable)
Zero-order lag models	Yes	Yes
Outlier detection	Large residuals.	Large residuals, impulse indicator saturation, or step indicator saturation.
Automatic outlier variables	Yes	Yes (with linear combinations)
Automatic weekday variables	Yes	No
Automatic day-segments variables	Yes	No
Automatic trend inclusion	Yes (by Phillips–Perron test)	No
Automatic log-transforms	Yes	No
Automatic constraints	Yes (equation specific)	Yes
Portmanteau test	Yes	Yes
Homoskedasticity test	Yes	Yes
Normality test	Yes	Yes
Chow test	No	Yes
Stability test	Yes	No
Validity test inclusion	Not configurable	Configurable
Automatic data imputation	Very limited	No
Scripting support	Yes (R script)	Yes (OxMetrics batch language)
Modeling non-VAR systems	Not supported	Supported
Data input formats supported	STATA, SPSS	STATA, Excel, *.csv

VI. CONCLUSION

With the recent developments of widespread portable consumer electronics devices being used as a means of data collection in healthcare, we investigated whether a fully automated approach to VAR is possible that does not require statistical expertize to operate, while still closely resembling the logic and decision making of statisticians working manually. The existing alternative follows a general-to-specific [5] approach that is different from the approach implemented in Autovar, and it does not automate some of the key operations that a statistician might perform when working manually (e.g., log transforming a dataset or including dummy variables for weekdays). Autovar leverages the power of automation to consider more potential models, and to improve on the manual process by developing a novel way for finding better constraints. Autovar serves as a proof of concept, and in this paper, we compared its performance against experts working manually, and its features against those of commercially available software (PcGive).

The results need to be interpreted with caution because the performance does not necessarily generalize to other manual analyses or datasets. Autovar needs to undergo simulation studies and statistical evaluation in order to assess the properties of the approach, and to determine whether the approach is useful outside the context of patient diary data. Also note that, for patient diary data in particular, VAR analysis may not be accurate when measurements are obtained at unequal intervals. Autovar currently has no functionality to preprocess the data to account for unequal intervals, and only very limited support for imputing missing values. With regards to the comparisons performed in the current study, we note that AIC/BIC scores are not the only measure of fit for ranking models. For example, the model with the best predictions is not necessarily the model that has the best fit on the current data [7, p. 62]. Moreover, in practice, a model that does not pass all validity tests can still be useful if it is reasonably close to passing those tests. These considerations are often taken into account by human experts because Autovar discards any models that fail any of the tests, its performance depends on the particular set of validity tests chosen, and since this set is not configurable, the flexibility of the approach is heavily limited.

We conclude by stating that the most important implication of Autovar is in making VAR feasible on a large scale. Current manual approaches may require days for analyzing a single dataset, i.e., they function on a small scale only. Likewise, other automated approaches work only on a small scale because their operation still requires a background in statistics. This is because applying and determining the applicability of certain actions, such as log transforming the data, including a trend or creating seasonal dummy variables, is not covered by automation in other automated approaches. Scaling any of the current alternatives, including any manual approach, to process multiple datasets in parallel would require employing multiple statisticians, which is expensive. Autovar, on the other hand, can perform the same tasks in minutes and does not require statistical expertise because its operation can be fully automated with trivial efforts (e.g., a line of R code to call Autovar with a filename). Thus, Autovar can work on a large scale at merely the cost of hardware.

Autovar is a demonstration of an exhaustive approach for a VAR model selection that is relatively safe to use. The requirement is that there is enough logic implemented to restrict the search space for models to the extent where the possibility of random models passing tests by chance is virtually nil. Under this assumption, performing a large-scale VAR model analysis without a background in statistics appears feasible, and a widespread application of fast and easy automated VAR analysis in healthcare could benefit more patients.

ACKNOWLEDGMENT

The authors would like to thank E. Wit and S. de Vos for help with mathematical proofs, A. van Gils for the manual analyses, C. Burton for providing datasets, F. Blaauw for proofreading, and J. Rosmalen for fruitful discussion on VAR.

REFERENCES

- [1] T. J. Sargent, "Estimating vector autoregressions using methods not based on explicit economic theories," *Fed. Reserve Bank Minneapolis Quart. Rev.*, vol. 3, no. 3, pp. 8–15, 1979.
- [2] G. E. Primiceri, "Time varying structural vector autoregressions and monetary policy," *Rev. Econ. Stud.*, vol. 72, no. 3, pp. 821–852, 2005.
- [3] B. Wild, M. Eichler, H.-C. Friederich, M. Hartmann, S. Zipfel, and W. Herzog, "A graphical vector autoregressive modelling approach to the analysis of electronic diary data," *BMC Med. Res. Method.*, vol. 10, no. 28, pp. 1–13, 2010.
- [4] M. Oorschot, T. Lataster, V. Thewissen, M. Wichers, and I. Myin-Germeys, "Mobile assessment in schizophrenia: A data-driven momentary approach," *Schizophrenia Bull.*, vol. 38, no. 3, pp. 405–413, 2012.
- [5] D. F. Hendry and H.-M. Krolzig, *Automatic Econometric Model Selection Using PcGets 1.0*. Timberlake Consultants, 2001.
- [6] C. W. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica, J. Econometric Soc.*, vol. 37, no. 3, pp. 424–438, 1969.
- [7] H. Lütkepohl, *New Introduction to Multiple Time Series Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
- [8] J. G. Rosmalen, A. M. Wenting, A. M. Roest, P. de Jonge, and E. H. Bos, "Revealing causal heterogeneity using time series analysis of ambulatory assessments: Application to the association between depression and physical activity after myocardial infarction," *Psychosomatic Med.*, vol. 74, no. 4, pp. 377–386, 2012.
- [9] T. Perez-Amaral, G. M. Gallo, and H. White, "A flexible tool for model building: The relevant transformation of the inputs network approach (RETINA)," *Oxford Bull. Econ. Stat.*, vol. 65, no. s1, pp. 821–838, 2003.
- [10] C. R. Nelson and C. R. Plosser, "Trends and random walks in macroeconomic time series: Some evidence and implications," *J. Monetary Econ.*, vol. 10, no. 2, pp. 139–162, 1982.
- [11] P. C. Phillips and P. Perron, "Testing for a unit root in time series regression," *Biometrika*, vol. 75, no. 2, pp. 335–346, 1988.
- [12] S. Stigler, "Fisher and the 5% level," *Chance*, vol. 21, no. 4, pp. 12–12, 2008.
- [13] J. D. Hamilton, *Time Series Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1994, vol. 2.
- [14] G. E. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*. San Francisco, CA, USA: Holden-Day, 1976.
- [15] F. X. Diebold, *Elements of Forecasting*. Cincinnati, OH, USA: South-Western, 1998.
- [16] G. M. Ljung and G. E. Box, "On a measure of lack of fit in time series models," *Biometrika*, vol. 65, no. 2, pp. 297–303, 1978.
- [17] H. White, "A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity," *Econometrica, J. Econometric Soc.*, vol. 48, no. 4, pp. 817–838, 1980.
- [18] C. W. J. Granger and A. P. Andersen, *An Introduction to Bilinear Time Series Models*. Göttingen, Germany: Vandenhoeck und Ruprecht Göttingen, 1978.
- [19] C. M. Jarque and A. K. Bera, "Efficient tests for normality, homoscedasticity and serial independence of regression residuals," *Econ. Lett.*, vol. 6, no. 3, pp. 255–259, 1980.
- [20] D. A. Belsley, E. Kuh, and R. E. Welsch, *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity* (Wiley Series in Probability and Statistics). New York, NY, USA: Wiley, 2004, vol. 546.
- [21] D. Cousineau and S. Chartier, "Outliers detection and treatment: A review," *Int. J. Psychol. Res.*, vol. 3, no. 1, pp. 58–67, 2010.
- [22] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Dec. 1974.
- [23] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 1978.
- [24] C. Burton, D. Weller, and M. Sharpe, "Functional somatic symptoms and psychological states: An electronic diary study," *Psychosomatic Med.*, vol. 71, no. 1, pp. 77–83, 2009.
- [25] A. van Gils, C. Burton, E. Bos, K. Janssens, R. Schoevers, and J. Rosmalen, "Individual variation in temporal relationships between stress and functional somatic symptoms," *J. Psychosomatic Res.*, vol. 77, no. 1, pp. 34–39, 2014.
- [26] P. D. Owen, "General-to-specific modelling using PcGets," *J. Econ. Sur.*, vol. 17, no. 4, pp. 609–628, 2003.
- [27] M. H. Pesaran and A. Timmermann, "A recursive modelling approach to predicting uk stock returns," *Econ. J.*, vol. 110, no. 460, pp. 159–191, 2000.
- [28] P. C. Phillips, "Econometric model determination," *Econometrica, J. Econometric Soc.*, vol. 64, no. 4, pp. 763–812, 1996.

- [29] (2015, Jan. 22). The R project for statistical computing. [Online]. Available: <http://www.r-project.org>
- [30] (2015, Jan. 22). Autovar: GitHub repository. [Online]. Available: <https://github.com/roqua/autovar>
- [31] B. Pfaff, "VAR, SVAR and SVEC models: Implementation within R package vars," *J. Stat. Softw.*, vol. 27, no. 4, pp. 1–32, 2008.
- [32] (2015, Jan. 22). Foreign: Read data stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase. [Online]. Available: <http://cran.r-project.org/web/packages/foreign/>
- [33] (2015, Jan. 22). Urca: Unit root and cointegration tests for time series data. [Online]. Available: <http://cran.r-project.org/web/packages/urca/>
- [34] (2015, Jan. 22). Twitter Bootstrap. [Online]. Available: <http://getbootstrap.com>
- [35] (2015, Jan. 22). The Apache software foundation. [Online]. Available: <http://www.apache.org/>
- [36] (2015, Jan. 22). OpenCPU: Scientific computing in the cloud. [Online]. Available: <https://www.opencpu.org/>
- [37] (2015, Jan. 22). knitr: A general-purpose package for dynamic report generation in R. [Online]. Available: <http://cran.r-project.org/web/packages/knitr/>
- [38] (2015, Jan. 22). Markdown: Markdown rendering for R. [Online]. Available: <http://cran.r-project.org/web/packages/markdown/>
- [39] H. Wickham, *Ggplot2: Elegant Graphics for Data Analysis*. New York, NY, USA: Springer, 2009.
- [40] R. B. D'Agostino, A. Belanger, and R. B. D'Agostino, Jr., "A suggestion for using powerful and informative tests of normality," *Amer. Stat.*, vol. 44, no. 4, pp. 316–321, 1990.
- [41] P. Royston, "Comment on sg3. 4 and an improved D'Agostino test," *Stata Techn. Bull.*, vol. 1, no. 3, pp. 23–24, 1992.
- [42] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. Ser. B, Method.*, vol. 39, no. 1, pp. 1–38, 1977.

Authors' photographs and biographies not available at the time of publication.