Analyzing Invariants in Cyber-Physical Systems using Latent Factor Regression

Marjan Momtazpour[1], Jinghe Zhang[2], Saifur Rahman[2],
Ratnesh Sharma[3], Naren Ramakrishnan[1]
[1]Discovery Analytics Center, Department of Computer Science, Virginia Tech
[2]Advanced Research Institute, Virginia Tech
[3]NEC Laboratories America, Inc.
[1]{marjan,naren}@cs.vt.edu,[2]{jing2014,srahman}@vt.edu,[3]ratnesh@nec-labs.com

ABSTRACT

The analysis of large scale data logged from complex cyber-physical systems, such as microgrids, often entails the discovery of invariants capturing functional as well as operational relationships underlying such large systems. We describe a latent factor approach to infer invariants underlying system variables and how we can leverage these relationships to monitor a cyber-physical system. In particular we illustrate how this approach helps rapidly identify outliers during system operation.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining

Keywords

Regression, Latent Factors, System Invariants, Outlier Detection.

1. INTRODUCTION

In recent years, with the rapid growth in data logged from modern devices in a distributed system, the need for having stronger knowledge discovery methods has attracted significant attention [28]. Concomitantly, the size and complexity of these systems have become a burden for administrators in detecting failures and repairing them [13, 25]. These challenges inspired us to characterize and track anomalies in cyber-physical systems by correlating all monitored data across the system.

According to [15], "*detecting anomalies that occur only within individual variables is often trivial, while detecting correlation anomalies is much harder and is practically important in fault analysis of complicated dynamic systems*". In a complex cyber-physical system, such as a smart grid (Fig.1), while some of the relationships between time series can be directly observed, other mutual dependencies are significantly complex to extract computationally. A typical cyber-physical system may include tens of time series with hundreds of mutual dependencies, where a large number of them are not directly observable. In the past, researchers have tried to infer existing linear relations using regression models [17] or by harnessing the structure of causal networks [4]. However, due to the complexity of modern systems, we must go beyond direct linear correlations in understanding them.

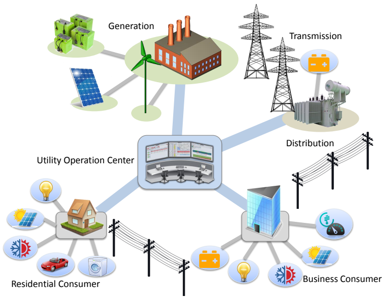


Figure 1: A typical example of a smart grid.

In this paper, we aim to use a more realistic approach to discover hidden patterns and indirect relationships among devices by employing latent variables in regression models. Specifically, we harness hidden factors derived by factor analysis and use them in regression models. We perform various experiments on synthetic and real datasets including wireless sensor networks and microgrid datasets. Furthermore, we use graph representations for better visualization of relationships which aids in discovering system-wide anomalies. Results show that the use of invariants derived with latent factors helps us to monitor large scale complex systems and discover outliers more precisely. We also propose a ranking method to score system-wide anomalies.

Our key contributions are thus:

- Proposing latent factor analysis regression to reveal hidden correlations among time series in a cyber-physical system.
- Summarizing the discovered invariants into an invariant graph of the system.
- Detecting system outliers based on the change in the graph of invariants and ranking time series for fault localization.

2. BACKGROUND

The high complexity of modern distributed cyber-physical systems urges us to enhance the self-management capabilities of these systems. Cyber-physical systems such as microgrid systems have a high degree of heterogeneity (in terms of shape, trend, and periodicity) that requires us to have a general tool to profile a variety of behaviors. Moreover, due to the nature of these systems, we may observe abrupt regime changes, seasonal patterns, and pairwise relationships among time series [11]. Ding et al. proposed an

KDD'15, August 11-14, 2015, Sydney, NSW, Australia.
© 2015 ACM. ISBN 978-1-4503-3664-2/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2788605.

ensemble of different approaches to tackle these problems in [11]. As stated in [28], traditional computational techniques cannot be used to model complex cyber-physical systems for data analytic purposes in a straightforward manner. There have been multiple research efforts to model complex dynamic systems such as inferring/visualizing the input-output relationships or predicting state switches/changes [28].

Guofei et al. [16] proposed a concept named *flow intensity* and used the ARX (autoregressive exogenous) model to quantify the relationship between each pair of flow intensities. If such a relationship holds all the time, they are considered as invariants of the underlying system. This model has been successful in characterizing complex systems and in supporting different system management tasks such as fault detection and localization. However, one of the main disadvantages of this method, as cited in [13], is that the complexity of algorithm in order to find all invariants is high. In this model, they look at two flow intensities (timeseries) where one of them is considered as input and the other one as output signal. Note that the differentiation of input and output time series are unknown and such labeling can occur only after examining both directions and evaluating which assignments lead to higher scores. The ARX model posits the following relationship between two flow intensities of y (output) and x (input):

$$\begin{aligned} y(t) + a_1 y(t-1) + ... + a_u y(t-u) \\ = b_0 x(t-l) + ... + b_v x(t-l-v) \end{aligned} \quad (1)$$

where u, v, and l are the order of the model and determine the number of previous steps that are affecting the current output. a_i's and b_j's are coefficient parameters that reflect how strongly a previous step is affecting the current output. Equation 1 can be solved using a least squares method (LSM) and the fitness score will indicate whether the model fits the observed data appropriately [16].

3. PROBLEM FORMULATION

Let us assume that we have observed a set of n time series, $\mathcal{D} = \{x_1(t), \ldots, x_n(t)\}$, measured at various points in one or more cyber-physical systems. For a time series $x_i(t)$, we represent the vector of samples at time steps t_k, ..., t_{k+w} as follows:

$$\mathcal{X}_i^{k:k+w} = \left[x_i(t_k), x_i(t_{k+1}) \ldots, x_i(t_{k+w})\right]^T. \quad (2)$$

Furthermore, we use X_i to represent the time series $x_i(t)$ as a random variable. In other words, $x_i(t)$ is a time series whose samples are drawn from a random distribution represented by random variable X_i.

In any type of cyber-physical system, there are various correlations and inter-dependencies among time series. In large cyber-physical systems, having sufficient level of knowledge about these inter-dependencies is crucial to preform accurate system management tasks. In the following definition, we formally define what we mean by dependency between two time series.

Definition 1. ***(Approximate Dependency):*** *At time step t_m, time series $x_j(t) \in \mathcal{D}$ approximately depends on $x_i(t) \in \mathcal{D}$, if and only if, there exists a function $f : \mathbb{R} \to \mathbb{R}$ that for appropriately small $\epsilon > 0$:*

$$\hat{x}_j(t_m) = f(\mathcal{X}_j^{1:m-1}, \mathcal{X}_i^{1:m}) \quad (3)$$

and

$$|x_j(t_m) - \hat{x}_j(t_m)| < \epsilon. \quad (4)$$

We depict this dependency by $x_j(t) \underset{\epsilon}{\to} x_i(t)\Big|_{t_m}$.

When the dependency between two time series does not change over time, we say that these two time series are system-invariants.

Definition 2. ***(System Invariants):*** *Two time series, $x_j(t) \in \mathcal{D}$ and $x_i(t) \in \mathcal{D}$, are system-invariant up to time T within range of ϵ if and only if at least one of the following rules satisfied:*

$$\exists f : \mathbb{R} \to \mathbb{R} \quad \textit{and} \quad \forall 0 \le t \le T \quad : \quad x_j(t) \underset{\epsilon}{\to} x_i(t)\Big|_{0 \le t \le T}$$

or

$$\exists f : \mathbb{R} \to \mathbb{R} \quad \textit{and} \quad \forall 0 \le t \le T \quad : \quad x_i(t) \underset{\epsilon}{\to} x_j(t)\Big|_{0 \le t \le T}.$$

We show invariant time series by $x_i(t) \overset{\epsilon}{\leftrightharpoons} x_j(t)$.

Based on the nature of the system, dependencies between time series can be linear or nonlinear and this is modeled by the function f. In complex cyber-physical systems, when we have a large number of time series, it is appropriate to represent the invariants in the form of a graph.

Definition 3. ***(Invariant Graph):*** *Graph $G = (V, E)$, with the set of vertices $V = \{v_1, \ldots, v_n\}$ and the set of edges $E = \{e_1, \ldots, e_m\}$, is called an invariant graph of a system with observed time series $\mathcal{D} = \{x_1(t), \ldots, x_n(t)\}$, where $e = (v_i, v_j) \in E$ if and only if $x_i(t) \overset{\epsilon}{\leftrightharpoons} x_j(t)$.*

From Definition 3 it is obvious that the vertex v_i is equivalent to the time series $x_i(t)$. It should be noted that system invariants and invariant graph represent features of a system under its normal condition. However, in the presence of anomalies, when the behavior of system deviates from its normal condition, these dependencies may disappear. In other words, while two times series, $x_i(t)$ and $x_j(t)$, may be invariant under normal conditions, the invariant feature may not hold when an anomaly condition appears in the system.

Definition 4. ***(Broken Invariants):*** *We say that system invariant $x_i(t) \overset{\epsilon}{\leftrightharpoons} x_j(t)$ is broken at time $T = t_m$, if and only if, time series $x_i(t)$ and $x_j(t)$ satisfy the following conditions:*

$$\exists f : \mathbb{R} \to \mathbb{R} \quad \textit{and} \quad \forall 0 \le t < T = t_m :$$

$$\left(x_j(t) \underset{\epsilon}{\to} x_i(t)\Big|_{t<T} \quad \wedge \quad \left|x_j(t_m) - f(\mathcal{X}_j^{1:m-1}, \mathcal{X}_i^{1:m})\right| \ge \epsilon \right)$$

or

$$\left(x_i(t) \underset{\epsilon}{\to} x_j(t)\Big|_{t<T} \quad \wedge \quad \left|x_i(t_m) - f(\mathcal{X}_i^{1:m-1}, \mathcal{X}_j^{1:m})\right| \ge \epsilon \right).$$

In some cases, the existence of unseen factors has an impact on the observed values of the system which cause them to have a specific behavior. However, uncovering those hidden factors behind all the underlying electro-mechanical devices is a challenging task. Characterizing these factors can help us to reveal the hidden relationships between potential time series whether they have indirect or complex relationships. Figure 2 (a) shows an example of relationships among a set of time series, $(x_1, x_2, \cdots, x_n)$. In reality, the relationships can be direct (solid lines) or indirect (dashed lines). Previous works tried to reveal the direct relationships among time series (which is shown in Figure 2 (b)). However, despite the simplicity of these linear methods, sometimes they result in a sparse graph of invariants where tracking all time series is impossible. Moreover, these methods are not able to capture the underlying hidden relationships and results in poor detection of system outliers. In this paper, we aim to uncover those hidden relationships

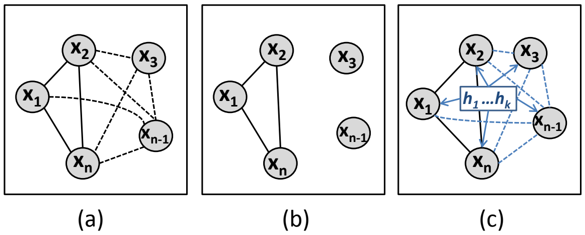


Figure 2: Different models of relationships: (a) Relationships in reality, (b) Relationships in the ARX model and (c) Relationships in ARX with a latent model.

with the help of hidden factors as latent variables. Hidden factors, $(f_1, f_2, \cdots, f_n)$, are considered as a higher level in hierarchy of the system and have an impact on the whole observed variables. An example of relationships in a system with hidden factors is illustrated in Figure 2 (c).

Definition 5. ***(Latent Variable):*** *In a cyber-physical system with the set of observed time series $\mathcal{D} = \{x_1(t), \ldots, x_n(t)\}$, an unobserved time series $h(t)$ is a latent variable when two or more observed time series are functions of $h(t)$. In other words,*

$$\begin{aligned} &\exists \mathcal{D}' \subset \mathcal{D} \quad where \quad \forall x(t) \in \mathcal{D}', \quad \exists g_x : \mathbb{R} \to \mathbb{R} \quad : \\ &\quad \forall t_m \geq 0, \quad x(t_m) = g_x(\mathcal{H}^{1:m}) \end{aligned} \tag{5}$$

where similar to Eq. 2, $\mathcal{H}^{1:m}$ is defined as follows:

$$\mathcal{H}^{1:m} = [h(t_1), h(t_2) \ldots, h(t_m)]^T .$$

It should be noted that each cyber-physical system may have more than one latent variable. Also, existence of a latent variable does not mean that all the observed time series should be directly related to that variable.

4. INVARIANT DISCOVERY

In this section, we describe a framework for invariant graph discovery and anomaly detection. For this purpose, we first extract latent variables using factor analysis and incorporate hidden factors into the regression model. Then we construct the invariant graph using a search algorithm. Finally, we use the constructed graph as a normal invariant graph and deploy it for the purpose of anomaly detection in the system. By discovering the broken invariants and ranking them, one may be able to find fault(s) and localize them.

4.1 Factor Analysis

Let us assume that we have a set of n random variables (input variables), denoted by X_1, ..., X_n. Also, assume that there are k hidden (latent) factors in the system, denotes by H_1, ..., H_k. Furthermore, assume that the observed variables are modeled as linear combinations of latent variables. Then we derive latent variables using the *factor analysis* method. Factor analysis is a well-studied field and is used to determine the main latent sources behind the observed data variation [9]. Although factor analysis is similar to principal component analysis (PCA), it is used more in predictive models due to its generalizability (e.g., factor loadings can remain consistent for different subsets of variables) [29]. Some might think of the factor model as generative models where the data is produced based on factors.

In factor analysis, for one sample of data extracted from random variable distributions, we have:

$$X_i - \mu_i = \lambda_{i1} H_1 + ... + \lambda_{ik} H_k + \zeta_i \tag{6}$$

where μ_i is the expected value of X_i, H_j's are unobserved random variables and λ_{ij}'s are unknown constants ($i \in 1, \cdots, n$ and $j \in 1, \cdots, k$ where $k < n$). Also, ζ_i's are independently distributed error terms with zero mean and finite variance ($Var(\zeta_i) = \psi_i$). In other words, by Eq. 6, each of the X_i's random variables is related to k hidden random variables, known as latent factors.

In matrix notation, we have:

$$\mathbf{X} - \mu = \mathbf{\Lambda H} + \mathbf{Z} \tag{7}$$

where $\mathbf{X} = (X_1, \cdots, X_n)^T$ is a data sample vector, μ is the expected values of data samples, $\mathbf{\Lambda}$ is an $n \times k$ matrix named as loading matrix, $\mathbf{H} = (H_1, \cdots, H_k)^T$ is a vector of latent factors, and $\mathbf{Z} = (\zeta_1, \cdots, \zeta_n)'$ is the vector of error.

It is assumed that $\mathbf{H}$ and $\mathbf{Z}$ are independent, and $E(\mathbf{Z}) = \mathbf{0}$, $E(\mathbf{H}) = \mathbf{0}$, $Cov(\mathbf{Z}) = Diag(\psi_1, ..., \psi_n) = \psi$, and $E(\mathbf{HH^T}) = \mathbf{\Phi}$. Furthermore, it is assumed that the data has a multivariate normal distribution, $\mathbf{X} = \mathcal{N}(\mu, \mathbf{\Sigma})$. Based on these assumptions, we will have:

$$\mathbf{\Sigma} = \mathbf{\Lambda \Phi \Lambda^T} + \psi. \tag{8}$$

Since $\mathbf{X}$ has a multivariate normal distribution, the actual distribution function of elements of sample covariance matrix, $\mathbf{S}$, can be expressed as a Wishart distribution with $m - 1$ degrees of freedom, $m\mathbf{S} \sim \mathcal{W}_n(\mathbf{\Sigma}, m-1)$, where m is the number of samples.

The log-likelihood of the Wishart distribution can be expressed as follows:

$$log\, L = -\frac{m-1}{2}\big(log|\mathbf{\Sigma}| + tr(\mathbf{S\Sigma^{-1}})\big) \tag{9}$$

where the terms independent of $\mathbf{\Sigma}$ are dropped.

It is obvious that maximization of L is equivalent to minimizing the following function:

$$Q = log\,|\mathbf{\Sigma}| + tr(\mathbf{S\Sigma}^{-1}). \tag{10}$$

One can find the latent variables by taking the partial derivatives of Eq. 10 with respect to the elements of loading matrix and errors constrained by Eq. 8. For simplicity, it is convenient to assume that $\mathbf{\Phi} = \mathbf{I}$ and $\mathbf{\Lambda^T \Psi^{-1} \Lambda}$ is diagonal.

There are different types of criteria to determine the number of factors such as criteria based on eigenvalues, discrepancy of approximation, or overall discrepancy [24]. Here, we use the Kaiser criterion which drops those with eigenvalues of less than 1. Indeed, the number of factors, must be lower than the number of observed variables, $k < n$. More details can be found in [14, 19].

4.2 Latent Factor Auto Regression with Exogenous input (LFRX)

Having n time series, $\mathcal{D} = \{x_1(t), \cdots, x_n(t)\}$, related to a cyber-physical system, similar to the ARX model [16], we can rewrite Eq. 1 as:

$$\hat{x}_j(t) = \sum_{p=1}^{u} a_p x_j(t-p) + \sum_{p=0}^{v} b_p x_i(t-l-p) \tag{11}$$

where $x_i(t), x_j(t) \in \mathcal{D}$.

As acknowledged widely [16, 13, 6, 5, 27], a drawback of ARX is that relationship discovery is done based on the existence of direct linear relationships between two observed time series. In other words, at each time ARX considers a pair of time series without considering the underlying relationships and hidden patterns. To address this issue, we deploy latent factors in the ARX model to recover the complex relationships. If we use latent factors in the

Algorithm 1: Invariant Search Algorithm

```
   Input: $x_i,\ i \in \{1,..,n\}$: set of time series, $\Delta$: ARX
          superiority threshold, $\tau$: minimum acceptable score, $t_s$
          and $t_e$: start and end time of training dataset.
   Output: $G$: Invariant Graph.
1  $S_{ARX} = \{\}$;
2  $S_{LFRX} = \{\}$;
3  for $i = 1$ to $n$ do
4      for $j = 1$ to $n$ do
5          if $i == j$ then
6              Continue;
7          end
8          foreach $t_s \leq t \leq t_e$ do
9              Learn an ARX model, $\theta_{ji}^{ARX}$, using Eq. 11;
10             Calculate $\hat{x}_j^{ARX}(t)$ using $\theta_{ji}^{ARX}$;
11             Compute $\mathcal{F}_{ji}^{ARX}(t)$ with Eq. 15;
12             Learn an LFRX model, $\theta_{ji}^{LFRX}$, using Eq. 12;
13             Calculate $\hat{x}_j^{LFRX}(t)$ using $\theta_{ji}^{LFRX}$;
14             Compute $\mathcal{F}_{ji}^{LFRX}(t)$ with Eq. 15;
15         end
16         if $\left(\sum_{t=t_s}^{t_e} \mathcal{F}_{ji}^{ARX}(t) \geq \sum_{t=t_s}^{t_e} \mathcal{F}_{ji}^{LFRX}(t) - \Delta\right)$ and
           $\left(\min_t(\mathcal{F}_{ji}^{ARX}(t)) \geq \tau\right)$ then
17             $S_{ARX} = S_{ARX} \cup \{x_i \leftrightharpoons x_j\}$;
18         end
19         if $\left(\sum_{t=t_s}^{t_e} \mathcal{F}_{ji}^{LFRX}(t) > \sum_{t=t_s}^{t_e} \mathcal{F}_{ji}^{ARX}(t) + \Delta\right)$ and
           $\left(\min_t(\mathcal{F}_{ji}^{LFRX}(t)) \geq \tau\right)$ then
20             $S_{LFRX} = S_{LFRX} \cup \{x_i \leftrightharpoons x_j\}$;
21         end
22     end
23 end
24 Construct Graph, $G = (V, E)$, using $S_{ARX}$ and $S_{LFRX}$;
25 return $G$;
```

above regression model, we will have:

$$\hat{x}_j(t) = \sum_{p=1}^{u} a_p x_j(t-p) + \sum_{p=0}^{v} b_p x_i(t-l-p) + \sum_{p=0}^{w} \sum_{q=1}^{k} c_{pq} h_q(t-p) \tag{12}$$

where $h_p(t)$'s are the latent factor time series that have been built based on the latent factor random variables, as discussed in the previous subsection (Eq. 6). Also, a_p's, b_p's, and c_{pq}'s are the regression weights that are determined in the learning phase. Note that in Eq. 12, in addition to the regression weights, latent factors are also unknown and should be estimated in the learning phase.

It should be noted that here we incorporate the previous values of $x_j(t)$ as well as values of exogenous variable, $x_i(t)$, and hidden variables, $h_q(t)$'s, to estimate new value of $x_j(t)$. In matrix notation, Eq. 12 will change to:

$$\hat{x}_j(t) = \mathbf{A}^T \mathcal{X}_j^{t-u:t-1} + \mathbf{B}^T \mathcal{X}_i^{t-l-v:t-l} + \mathrm{Tr}(\mathbf{C}^T \boldsymbol{\mathcal{H}}) \tag{13}$$

where $\mathbf{A}_{u\times 1}$, $\mathbf{B}_{(v+1)\times 1}$, $\mathbf{C}_{(w+1)\times k}$ are matrices of coefficients. Also, $\boldsymbol{\mathcal{H}}_{(w+1)\times k}$ is a matrix that represents all the latent factors, i.e. $\boldsymbol{\mathcal{H}} = \left[\mathcal{H}_1^{t-w:t} \cdots \mathcal{H}_k^{t-w:t}\right]$. In our experiments, we assume $u = v = w$ and their values are estimated using cross-validation. Also, due to the lack of delay in our datasets, we assume l is zero. In order to solve Eq. 13, first we derive latent factors, $\boldsymbol{\mathcal{H}}$, using factor analysis of Subsection 4.1 and then we incorporate them into the regression model to estimate the weights.

4.3 Invariant Graph Construction

Based on Definition 2, in order to discover system invariants we need to identify time series that have persistent approximate dependencies. While time series may have nonlinear dependencies, in this paper we consider linear relationships and use ARX and LFRX for this purpose.

The search algorithm that extracts system invariants is shown in Algorithm 1. In this algorithm, for each pair of time series, we first assume that they have a direct linear relationship and we fit them using an ARX model (Eq. 11). The ARX model for time series $x_i(t)$ and $x_j(t)$ is illustrated by θ_{ij}^{ARX}. Then, we assume that there might be an indirect relationship through latent variables and hence, we use LFRX model to learn θ_{ij}^{LFRX}. As defined in Definition 1, to determine if $x_j(t)$ depends on $x_i(t)$, we need to compare the estimation error with an acceptable threshold, ϵ. However since in a specific cyber-physical system different time series have different range of values, it is more appropriate to use normalized error measurements. For this purpose, when we estimate $x_j(t)$ based on $x_i(t)$, we can evaluate the relative absolute error (RAE) defined by the following equation:

$$e_{j,i}^{RAE}(t) = \frac{|\hat{x}_j(t) - x_j(t)|}{\sum_{t=t_s}^{t_e} |\overline{x}_j(t) - x_j(t)|} \tag{14}$$

where $x_j(t)$ is the observed value, $\hat{x}_j(t)$ is the estimated value based on $x_i(t)$, and $\bar{x}_j$ is the sample mean of observed values.

According to [17] for each pair of time series, $x_i(t)$ and $x_j(t)$, we calculate a score to measure their dependencies. The following normalized score may be used for the evaluations:

$$\mathcal{F}_{j,i}(t) = 100(1 - e_{j,i}^{RAE}(t)). \tag{15}$$

A higher score indicates stronger dependency between the time series. It should be noted that RAE is a specific example of normalized error measurement and one can easily extend the algorithm to use other error measurement approaches including RMSE, specifically when time series have the same range of variations.

In Algorithm 1, lines 8 to 15 are dedicated to estimate individual values of $x_j(t)$ based on $x_i(t)$ and calculate the scores for ARX and LFRX models. In order to discover invariants and choose between direct or indirect relationships, we consider the following criteria:

- For all the time steps, score should be greater than or equal to a specific threshold. We name this threshold as *minimum acceptable score* and denote it by τ. Then, we should have:

$$\min_t(\mathcal{F}_{ji}(t)) \geq \tau. \tag{16}$$

- Since higher score depict stronger relationships, in choosing between ARX and LFRX, the one with the better overall score is chosen.
- Since linear invariants represent simpler relationships, higher priority is given to ARX-based invariants using a guard bound, Δ, which we name as the *ARX superiority threshold.* In other words, ARX-based invariants are selected when:

$$\sum_{t=t_s}^{t_e} \mathcal{F}_{ji}^{ARX}(t) \geq \sum_{t=t_s}^{t_e} \mathcal{F}_{ji}^{LFRX}(t) - \Delta. \tag{17}$$

When calculated scores satisfy Eq. 16, we will say that $x_i(t)$ and $x_j(t)$ are invariant and based on Eq. 17, the type of invariant is chosen to be direct (ARX) or indirect (LFRX). The resulted invariants are added to the sets of ARX and LFRX invariants denoted by S_{ARX} and S_{LFRX}, respectively. In Algorithm 1, lines 16 to 21 are dedicated to this process.

Algorithm 2: Alerting Algorithm

```
Input: $x_i(t)$, $i \in \{1,..,n\}$: set of time series,
       $G = (V, E)$:Invariant Graph, $t_e$: start time of
       monitoring, $\alpha$: alerting threshold.
foreach $t > t_e$ do
    foreach $e_{i,j} \in E$ do
        Use Definition 4 to check if $e_{i,j}$ is broken;
        if $e_{i,j}$ is broken then
            $cnt_{ij} \leftarrow cnt_{ij} + 1$;
        else
            $cnt_{ij} \leftarrow 0$;
        end
        if $cnt_{ij} > \alpha$ then
            Invoke an alert;
            $cnt_{ij} \leftarrow 0$;
        end
    end
end
```

After finding system invariants, the final step (line 24 in Algorithm 1) is to construct the invariant graph, $G = (V, E)$. This is a straightforward task which is performed based on Definition 3. The total number of iterations of this algorithm is $O(tn^2)$ where t is the length of time series. At each iteration (lines 9 to 14), models are learned with a time complexity which is a function of t^2 and various constants $(w, v, u, \cdots)$. This results in an overall complexity of $O(Cn^2t^3)$.

4.4 Outlier Detection using Broken Invariants

After constructing the invariant graph (in Subsection 4.3), we can use this graph for detecting abnormalities in the system. For this purpose, using Definition 4, at each time step we check whether each of the graph edges is broken or not. We then rank the time series in order to localize the source of abnormality. In what follows we first describe the alerting algorithm, followed by a metric for alerting threshold estimation and finally the ranking method for fault localization.

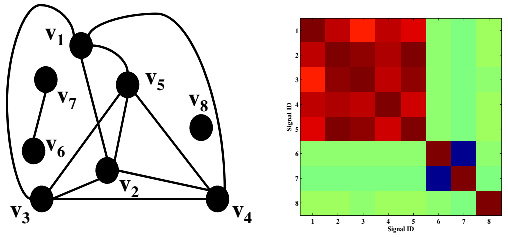
Anomaly alerting algorithm: The alerting algorithm is illustrated in Algorithm 2. In this algorithm, in order to prevent generations of multiple alerts consecutively, we use an alert filtering mechanism by imposing a counting strategy with alert threshold of α. When the number of consecutive violations of a specific invariant goes beyond α, the algorithm invokes an alert to the system administrator, who may use this for further investigations. Time complexity of this algorithm is $O(|E|)$ at each time-step.

Anomaly detection threshold: According to model-based FDI methods used in control theory and similar to [27], in order to reduce false alarms the following approach is used for detection of broken invariants. The difference between the predicted value, $\hat{x}_j(t)$, and the actual value, $x_j(t)$, is recorded and whenever this difference deviates more than a predetermined threshold, ϵ_0, an invariant will be broken:

$$|\hat{x}_j(t) - x_j(t)| > \epsilon_0 \tag{18}$$

The threshold ϵ_0 can be estimated based on the observed values in the training period. According to [27], ϵ_0 is assumed to be 10% larger than the tolerance of deviations from the actual values:

$$\epsilon_0 = 1.1 * arg_r \{Prob(|\hat{x}_j - x_j| < r) = 0.995\} \tag{19}$$



(a) (b)

Figure 3: (a) Invariant graph of synthetic data (b) Correlation matrix of synthetic time series.

where r is greater than 99.5% of the residuals observed in the training data.

Ranking time series for fault localization: In complex cyber-physical systems with a large number of invariants, one single fault in the system may lead to a large number of broken invariants. Hence, for fault localization we need to rank the invariant graph vertices according to the number of their broken edges. Similar techniques have also been used in [12]. For this purpose, we use the following score to rank the vertices after the occurrence of an alarm. Assuming that an alarm is generated at time t, for each vertex, v_j, we calculate the following score:

$$\rho_j = \frac{d_j^{normal} - d_j(t)}{d_j^{normal}} \tag{20}$$

where d_j^{normal} is the degree of v_j in normal condition and $d_j(t)$ is the degree of v_j after alarm generation at time t. It is obvious that higher value of ρ_j indicates v_j has lost more edges which may potentially be due to the occurrence of a fault at $x_j(t)$.

5. EXPERIMENTAL RESULTS

We perform our experiments on several datasets. We aim to show how our method (ARX + LFRX) can discover the invariants, how it can improve the accuracy of system, and how it can find the anomalies happening throughout the network. First, we perform our analysis on a synthetic dataset to recover indirect invariants. Next, we use two datasets from real cyber-physical systems: a wireless sensor network and a microgrid system. In these datasets, there are multiple factors and measurements with various temporal and spatial dependencies.

5.1 Synthetic Data

Dataset Description: At the first step, we perform our experiment on a synthetic dataset to verify our method for the discovery of indirect hidden relationships. For this purpose, we generate eight signals and compare the results of ARX with our method (integration of ARX and LFRX). In this experiment, we add a Gaussian noise with zero mean and standard deviation of 0.1 to one of the time series in order to test the invariant graph under abnormalities. The ground truth graph and its corresponding correlation matrix are shown in Fig. 3 (a) and (b), respectively. As Fig. 3 (a) shows, V_6 and V_7 are correlated to each other. V_8 is isolated and all the remaining nodes are correlated to each other. However, the hidden relationship between signals is not observable in Fig. 3 (a). In fact, V_3, V_4, V_5 are generated using V_1 and V_2. The relationship between signals is given in the following equations:

$$V_1(t) = 0.9V_1(t-1) - 0.02V_1(t-2) - 0.01V_1(t-3) + 0.09 + \eta$$

$$V_2(t) = 2(V_1(t-1) - V1(t-2)) + 0.5(V_2(t-1) + V2(t-2))$$

$$V_3(t) = V_1(t-1)+V_2(t)-V_1(t),\ V_4(t) = 3V_1(t-1)+V_2(t-1)$$

$$V_5(t) = 3V_1(t-1) - V_2(t-1)$$

$$V_6(t) = 1 + 0.01R(t, 100),\ V_7(t) = 1 - 0.01R(t, 100)$$

$$V_8(t) = 2e^{10^{-4}t} + R(t, 600)e^{-10^{-4}t}$$

where $R(t, T)$ is a rectangular function of t, oscillating between -1 and 1 with period of T and η is a Gaussian noise with zero mean and standard deviation of 0.01.

In order to consider various situations, with and without presence of hidden relationships, we perform multiple experiments with different subset of the above signals.

Results and Discussion: Recovered graph for both methods in normal and abnormal condition are shown in Figure 4. In this figure, each row denotes an experiment involving a subset of synthetic time series, where white nodes represent the one with injected noise. As it is shown, in all cases the ARX + LFRX method has recovered the planted invariants and the recovered graph matches the ground truth. In both methods, in the presence of an anomaly, the invariants attached to the corrupted signal (white node) are broken. However, in some cases such as (a) and (b) where the ARX method cannot recover the existing relationships, at the time of anomaly, it was not able to detect it correctly. In figures (a) to (d), time series V_1 and V_2 are not measured and hence, time series V_3, V_4, and V_5 have indirect relationships. It is obvious from Fig. 4 that the proposed method (ARX + LFRX) is able to discover the corresponding invariants while ARX, with the same parameter settings, has failed to discover them.

5.2 Sensor Motes

Dataset Description: The sensor motes dataset contains measurements from wireless sensors at Intel Berkeley Research lab. There are a total of 54 sensors located at a lab measuring temperature, humidity, light, and voltage between February 28th and April 5th, 2004 [10]. Each sensor was able to record different variables every 31 seconds. Fig. 5 (a) shows the location of each node as well as different part of the lab.

Results and Discussion: Fig. 5 (b) shows the clustering of sensors in the loading matrix (i.e. Λ in Eq. 7) of light measurements. For this purpose, we used a k-means algorithm with $k = 6$. It is interesting to note that the sensors are clustered in a way that reflects their spatial distributions.

We performed invariant graph analysis on each variable (light, temperature, humidity, and voltage), separately. Overall results are illustrated at Table 1. It is obvious from this table that the proposed method results in lower average estimation error on test datasets, compared to the ARX approach. Also, the number of discovered invariants using the proposed method is higher than the one using ARX. This is due to the deployment of latent factors in LFRX method which is beneficial in anomaly detection. In fact, for the purpose of anomaly detection using invariant graphs, anomalies in vertices with a small number of edges cannot be discovered easily. One might think that by increasing the value of ϵ (in Eq. 4) at the time of invariant discovery, we can find larger number of invariants. However, by increasing ϵ, the estimation accuracy decreases dramatically which results in having false invariants. Table 1 shows that the proposed method discovers more invariants with higher accuracy.

Fig. 6 shows an invariant graph of temperature under the normal condition and in the presence of abnormalities. In this figure, direct and indirect invariants are illustrated by blue and red edges, respectively. Fig. 6 (a) shows the invariant graph under the normal condition with 183 edges where 140 of them are derived using LFRX. As we expected, geographic placement of sensors has an effect on the result. Fig. 6 (b) shows the invariant graph at the presence of anomalies. In this figure, the top ten sensors based on the ranking of Eq. 20 are highlighted with red circles. Larger circles represents higher rank of vertices. As this figure shows, variations of environmental temperature in the lab result in distortions in nearby sensors. As an example, sensors 12 to 17 are in the top ten ranking list. Sensors 16 and 37 are both among the highly ranked ones that are susceptible to be the source of anomaly. In Fig. 6(b), we can observe that the LFRX edge between these two vertices is broken. Fig. 7 depicts the corresponding time series of these two sensors. We can easily observe that these two time series have almost similar behavior and are expected to be system invariant. As this figure illustrates, the relationship between these sensors is broken at time $t = 1300$. Abnormal conditions are shown in darker colors.

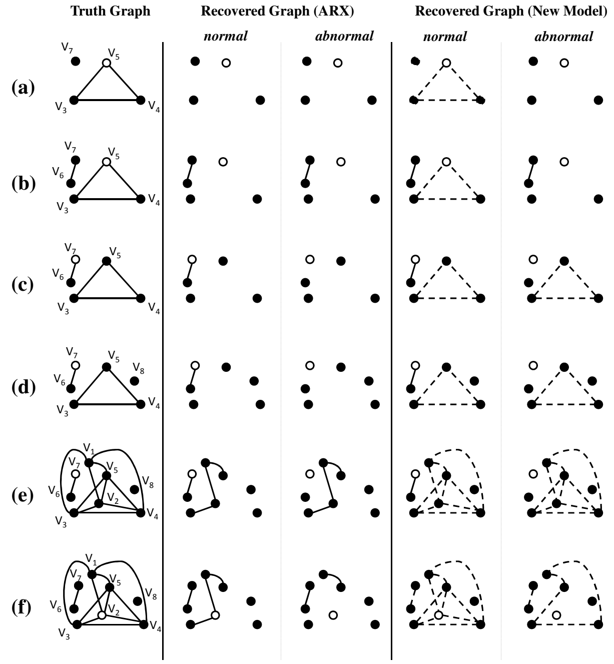
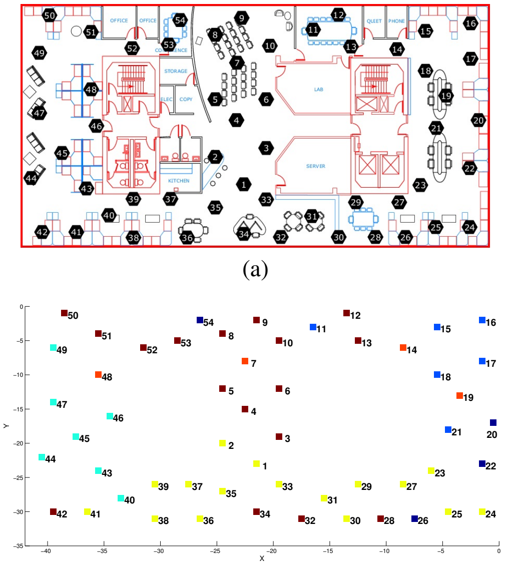


Figure 4: Invariant graphs discovered using ARX and the proposed method (ARX + LFRX) under normal and abnormal conditions. First column shows the ground truth. In the abnormal condition, an anomaly is injected into each graph at one variable (white node). Rows (a) to (f) shows different combinations of time series in Fig. 3. Direct invariants are shown in solid lines and indirect invariants are shown in dashed lines.

Table 1: Performance evaluation result of ARX and (ARX + LFRX) for the Sensor Motes dataset.

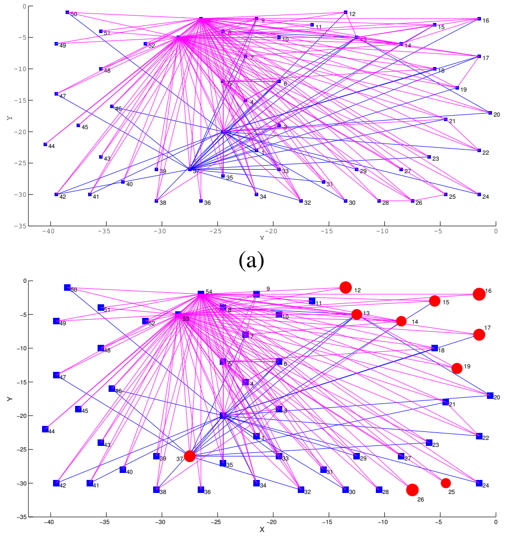
Metric	Avg. Error ARX+LFRX	Avg. Error ARX	Edges ARX+LFRX		Edges ARX
			Total	Latent	
Voltage	0.0056	0.0070	441	284	170
Temperature	0.3658	0.4971	183	43	145
Humidity	0.6262	0.8632	1142	866	764
Light	81.8377	93.1723	539	269	463



(a)

(b)

Figure 5: (a) Geographical location of wireless sensors (taken from [1]). (b) Clustering of sensors based on latent factors of light measurement indicating a high degree of spatial correlation.



(a)

(b)

Figure 6: (a) Invariant graph of sensors based on temperature at normal condition (with 183 edges). ARX-based and LFRX-based invariants are shown with red and blue edges, respectively. (b) Invariant graph with broken edges in the presence of an anomaly (162 edges). Top ten sensors based on the ranking of Eq. 20 are shown with red circles.

5.3 Microgrid

Dataset Description: We performed our experiments on a microgrid system where several devices are operating in a distributed setting (Fig. 8). In this setting, the control unit tries to minimize the amount of energy based on various criteria and hence the microgrid shows a complex behavior in the logged measurements. This dataset which is provided by NEC labs contains logged data from multiple sources such as loads (primary, secondary), battery, PMU

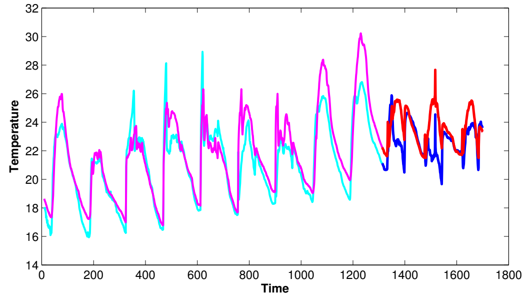


Figure 7: Temperature of sensors 16 (blue) and 37 (red). Outliers are shown in darker color.

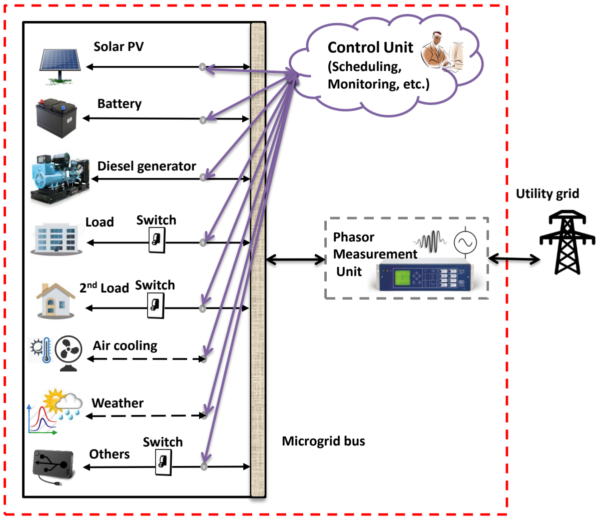


Figure 8: Schematic view of the NEC microgrid setup.

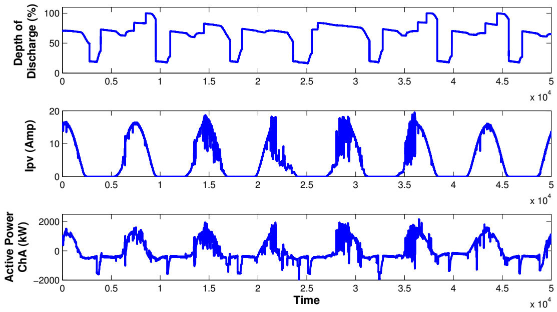
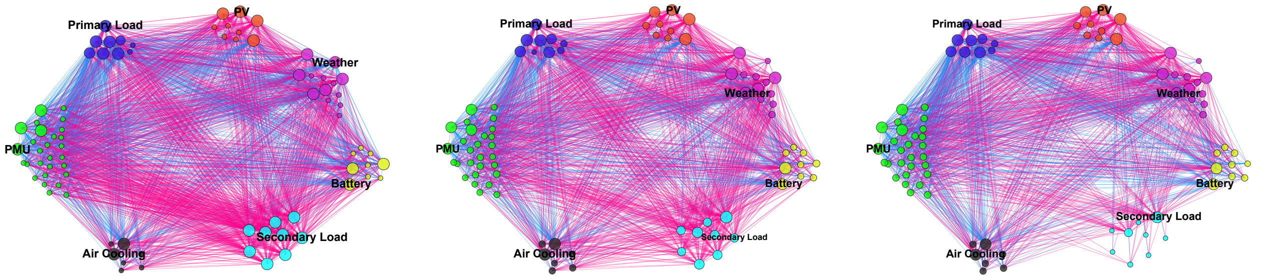


Figure 9: Three different time series of NEC microgrid dataset during one week.

(measurement unit outside of the microgrid), solar system (PV), weather (inside and outside parameters), and air cooling unit. There are total of 84 features measured from July 7th to August 7th, 2014. Due to the different sampling rates of each device, time series are re-sampled with a unique rate to be aligned to a specific window-time. Figure 9 shows a sample plot of three time series from different units during one week.

Results and Discussion: The invariant graph derived by our proposed method is represented in Fig. 10 (a). In this figure, each node represents one of the features and the set of features that belong to



(a) (b) (c)

Figure 10: (a) Invariant graph of microgrid under normal condition (ARX + LFRX) (b) Outlier happens when an additional device is switched off/on in the system. (c) Outlier happens when the secondary load is disconnected.

Table 2: Performance evaluation result of ARX and (ARX + LFRX).

Device (No. of Signals)	Avg. Error (ARX + LFRX)	Avg. Error (ARX)	Intra-Device Edges (ARX + LFRX)		Intra-Device Edges (ARX)	Inter-Device Edges (ARX + LFRX)		Inter-Device Edges (ARX)
			Total	Latent		Total	Latent	
PMU (28)	0.0028	0.1346	102	73	102	810	265	755
PV (9)	0.0119	0.1146	26	5	26	187	63	175
Battery (9)	3.7×10^{-14}	0.2764	21	12	21	239	81	226
Primary LD (9)	2.2×10^{-4}	0.1460	35	22	29	235	66	236
Secondary LD (9)	2.3×10^{-6}	0.0434	36	0	36	53	19	52
Weather (13)	0	0.1342	57	17	57	260	94	244
Air Cooling (7)	0.2852	0.2903	11	5	11	213	75	203

a specific device are illustrated using the same color. Also, the size of each node is proportional to its degree. Furthermore, invariants derived by ARX and LFRX methods are shown by red and blue edges, respectively. The total number of invariants that ARX + LFRX discovered is 2285 where 797 of them are indirect and 1488 of them are direct invariants.

The average estimation errors of traditional ARX and the proposed method are compared in Table 2. As this table shows, the average error of the proposed method for each device is dramatically lower than the error resulted by ARX approach. This means that invariants are selected with higher accuracy using ARX + LFRX. Also, Table 2 compares the proposed method with ARX in terms of the number of invariants between devices (Inter-Device) and within each device (Intra-Device). Inter-Device edges are visualized in Fig. 11(a) where edge thickness represents the total number of invariants between devices. From this figure, we can observe the high complexity of inter-dependencies between measurements of devices. For example, energy produced by photovoltaics (PV) has effect on battery, PMU, loads, and the temperature of environment.

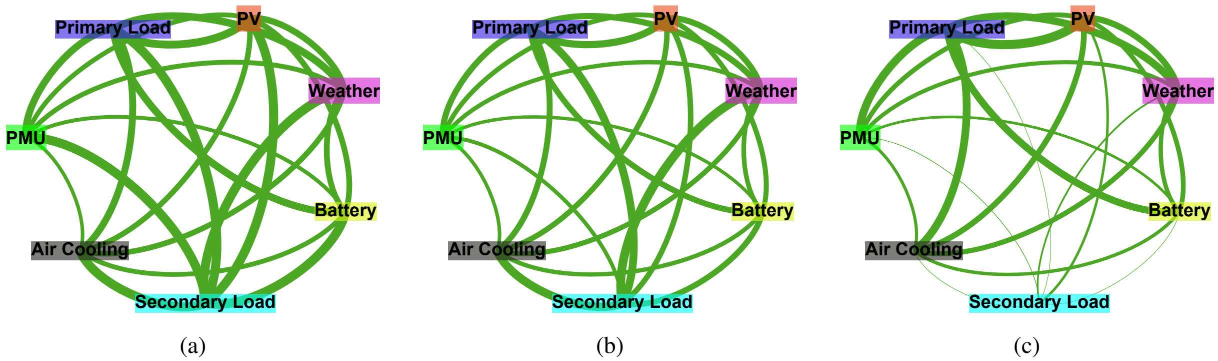
After occurrence of an abnormal behavior, the topology of the invariant graph changes (i.e., depending on the nature of anomaly, some edges are removed from the graph). By comparing consecutive graphs, one is able to detect outliers in the system. We detect changes in the invariant graph in two different situations: when an additional device is switched off/on and when secondary load is turned off. Results are shown in Fig. 10 (b) and (c), respectively. As we may observe from Fig. 10 (b), two edges connecting the secondary load to PMU and PV are broken. This is due to the change in the energy consumption behavior of the system. On the other hand, as Fig. 10 (c) depicts, a large number of invariants between secondary load and other devices are broken. This is due to the disconnection of the secondary load. It should be noted that measurements of the secondary load were among the top five anomalies returned by our outlier ranking method. Also, the number of remaining invariants between devices are shown in Fig. 11 (b) and (c).

Figure 12 shows some examples of anomalies in the microgrid system. Each figure shows a pair of time series that under normal condition are invariant. Abnormal conditions are shown in darker colors. The gap between occurrence of anomalies and normal time series depicts the time difference between them. As an example, Fig. 12 (a) shows the detection of sudden change in the red curve which is the State of Health (SOH) of the battery. This change is detected using the broken link between SOH and Reactive power in Channel C of the PMU. Detecting such anomalies is crucial for microgrid operators.

It should be noted that since we do not know the labeling of real dataset, we are unable to evaluate our method using precision and recall metrics. Nevertheless, we calculated precision and recall under different scenarios. As an example, when 10 nodes have random injected noise, by looking at the top 10 ranked results, precision and recall were equal. This value is 0.51 for the ARX method, whereas for ARX + LFRX, it is 0.68.

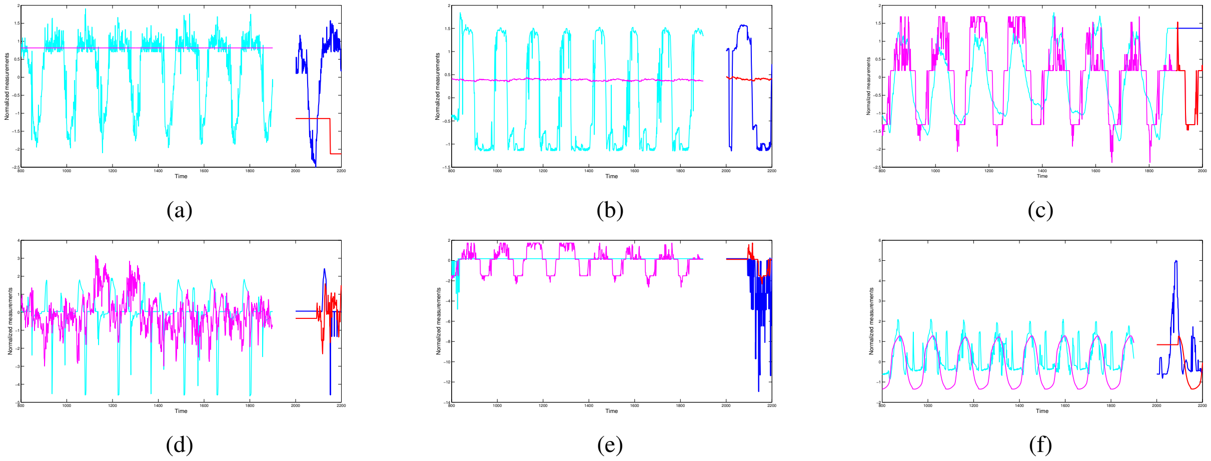
6. RELATED WORK

Smart Grid and Power System Analytics: Power grids comprise a large number of elements and processes that are highly dynamic and complex. Traditionally power system operational studies are primarily based on a quasi-steady-state assumption, with static and explicit models that largely ignores dynamic characteristics of loads and control devices. The classic weighted least square (WLS) estimator, combined with methods such as largest normalized residual test and hypothesis testing identification, is extensively used for system diagnosis and outlier identification [2]. Recent developments in smart grids have revealed to us insight into stochastic operating behaviors and dynamics that we were never able to observe before. In particular, the widespread deployment of smart meters, renewable generation, smart load controls, energy storage, and plug-in hybrid vehicles will require fundamental



(a) (b) (c)

Figure 11: (a) Invariant graph of inter-devices of microgrid (ARX + LFRX) (b) Outlier happens when additional device is switched off/on in the system (c) Outlier happens when secondary load is disconnected.



(a) (b) (c)

(d) (e) (f)

Figure 12: Invariant time series at normal and abnormal conditions: (a) Reactive power Channel C vs. State of Health of Battery (b) Power factor vs. Voltage of channel A in PMU (c) Outside temperature vs. Power Factor of primary load (d) Battery Current vs. Inverter output voltage (e) Peak voltage of secondary load vs. Power factor (f) Current magnitude of Channel A vs. Peak power of primary load.

changes in the operational concepts and principal components of the grid, in order to achieve real-time operation and control.

Fraud detection and particularly detection of energy theft is one of most important concerns in the smart grid [18, 22]. Data analytic methods can play an important role in identifying abnormal consumption trends and possible malicious activities in such systems. Daisuke et al. [22] used ARMA and LOF methods in an adversarial environment to detect attacks in data collected using advanced meter infrastructure (AMI). Rong et al. [18] compared classification-based, state-based, and game theory-based methods in energy-theft detection schemas.

One area that has witnessed significant developments is in the use of phasor measurement units (PMUs). Chen et al. [7] use PCA for online monitoring of PMU data for the purpose of early event detection. Khan et al. [20] proposed a parallel fluctuation approach using MapReduce techniques. At the lower level, Momtazpour et al. [23] proposed an integrated data-driven framework to study the behavior of battery systems in microgrids using clustering, regression, and spectral clustering of time series for the purposes of high level characterization of usage behavior and online parameter estimation.

Invariant Discovery and Structure Learning: Sharma et al. [27] used ARX for invariant discovery in distributed systems and discussed the challenges in fault localization for data centers. Shan et al. [26] have extracted overlay invariants based on pairwise invariant networks for fault detection and capacity planning in distributed systems. Due to the time complexity of invariant discovery of large scale systems, Ge et al. [13] developed an effective pruning techniques based on the identified upper bounds. In some applications, the existence of anomalies in invariant graphs yields many broken links which makes it difficult for a system expert to manually inspect each broken link. Hence, Ge et al. in [12] proposed two different methods of ranking metrics according to the anomaly levels occurring in invariant networks.

In a closely related area, viz. causal modeling of time-series data, Arnold et al. [4] used the concept of Granger causality to infer the structure of the causal network given set of time series. These authors compared performance of the exhaustive Granger method and a Lasso-Granger method with benchmark methods including the VAR and SIN methods. However, in [4], the main goal was to construct causal graphs instead of addressing data with correlated variables. Subsequently, Liu et al. [21] used a hidden Markov random field regression framework to infer temporal causal structures.

Cheng et al. [8] use time order relationships to capture temporal dependence structures underlying multivariate time series.

Anomaly Detection in Graphs: Akoglu et al. [3] provide an extensive survey of anomaly detection methods in graphs spanning different settings: unsupervised, (semi-) supervised approaches, static, dynamic, attributed, and plain graphs. In dependency graphs, for the purpose of anomaly detection, Ide et al. [15] used sparse structure learning to compute correlation anomaly scores of each variable using neighborhood selection approaches.

7. CONCLUSION

Invariant discovery is an exciting research field which aims to discover underlying relationships in cyber-physical systems. We used latent factor regression analysis and combined it with the ARX model (ARX + LFRX) to recover underlying direct and indirect relationships. These invariants are helpful in decision making and monitoring processes such as outlier detection. We tested our models on several datasets and results showed that with the help of latent factors, the accuracy of discovered invariants was higher than traditional methods. Investigating other topologies involving latent variables (such as a mesh network) and heuristic search algorithms to reduce the computational complexity are some of the directions for future research.

8. REFERENCES

[1] Intel Lab Data. `http://select.cs.cmu.edu/data/labapp3/index.html`, 2008. [Online; accessed 03-June-2015].

[2] A. Abur and A. G. Exposito. *Power system state estimation: Theory and implementation*. Marcel Dekker, 2004.

[3] L. Akoglu, H. Tong, and D. Koutra. Graph-based anomaly detection and description: A survey. *Data Min. and Knowl. Disc.*, 28(4), 2014.

[4] A. Arnold, Y. Liu, and N. Abe. Temporal causal modeling with graphical granger methods. In *Proc of KDD'07*, pages 66–75. ACM, 2007.

[5] H. Chen et al. Exploiting local and global invariants for the management of large scale information systems. In *Proc ICDM'08*, pages 113–122, 2008.

[6] H. Chen, G. Jiang, K. Yoshihira, and A. Saxena. Invariants based failure diagnosis in distributed computing systems. In *Proc. IEEE Symp on Reliable Distributed Systems*, pages 160–166, 2010.

[7] Y. Chen, L. Xie, and P. Kumar. Dimensionality reduction and early event detection using online synchrophasor data. In *Power and Energy Society General Meeting (PES), 2013 IEEE*, pages 1–5, July 2013.

[8] D. Cheng, M. T. Bahadori, and Y. Liu. Fblg: A simple and effective approach for temporal dependence discovery from time series data. In *Proc KDD'14*, pages 382–391, 2014.

[9] J. C. F. De Winter and D. Dodou. Common factor analysis versus principal component analysis: a comparison of loadings by means of simulations. *Communications in Statistics - Simulation and Computation*, June 2014.

[10] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *Proc. Int Conf on VLDB*, 2004.

[11] M. Ding, H. Chen, A. Sharma, K. Yoshihira, and G. Jiang. A data analytic engine towards self-management of cyber-physical systems. In *Proc IEEE ICDCSW'13*, pages 303–308, 2013.

[12] Y. Ge, G. Jiang, M. Ding, and H. Xiong. Ranking metric anomaly in invariant networks. *ACM Trans. Knowl. Discov. Data*, 8(2):8:1–8:30, June 2014.

[13] Y. Ge, G. Jiang, and Y. Ge. Efficient invariant search for distributed information systems. In *Proc. ICDM'13*, pages 1049–1054, 2013.

[14] H. Harman. *Modern Factor Analysis*. University of Chicago Press, Chicago, 3rd ed. edition, 1976.

[15] T. Ide, A. C. Lozano, N. Abe, and Y. Liu. Proximity-based anomaly detection using sparse structure learning. In *Proc SDM'09*, 2009.

[16] G. Jiang, H. Chen, and K. Yoshihira. Discovering likely invariants of distributed transaction systems for autonomic system management. *Cluster Computing*, 9(4):385–399, Oct. 2006.

[17] G. Jiang, H. Chen, and K. Yoshihira. Efficient and scalable algorithms for inferring likely invariants in distributed systems. *IEEE Trans. on Knowl. and Data Eng.*, 19(11):1508–1523, Nov. 2007.

[18] R. Jiang, R. Lu, Y. Wang, J. Luo, C. Shen, and X. S. Shen. Energy-theft detection issues for advanced metering infrastructure in smart grid. *Tsinghua Science and Technology*, 19(2):105–120, April 2014.

[19] K. Joreskog. Some contributions to maximum likelihood factor analysis. *Psychometrika*, 38(2):183–202, 1967.

[20] M. Khan, M. Li, P. Ashton, G. Taylor, and J. Liu. Big data analytics on pmu measurements. In *Proc Int Conf on FSKD'14*, pages 715–719, Aug 2014.

[21] Y. Liu, A. Niculescu-mizil, A. C. Lozano, and Y. Lu. Learning temporal causal graphs for relational time-series analysis. In *Proc ICML'10*, pages 687–694, 2010.

[22] D. Mashima and A. A. Cardenas. Evaluating electricity theft detectors in smart grid networks. In *Research in Attacks, Intrusions, and Defenses*, volume 7462 of *Lecture Notes in Computer Science*, pages 210–229. 2012.

[23] M. Momtazpour, R. Sharma, and N. Ramakrishnan. An integrated data mining framework for analysis and prediction of battery characteristics. In *Proc IEEE ISGT Asia'14*, pages 774–779, 2014.

[24] K. J. Preacher, G. Zhang, C. Kim, and G. Mels. Choosing the optimal number of factors in exploratory factor analysis: A model selection perspective. *Multivariate Behavioral Research*, 48:28–56, 2013.

[25] I. Shafer, K. Ren, V. N. Boddeti, Y. Abe, G. R. Ganger, and C. Faloutsos. Rainmon: An integrated approach to mining bursty timeseries monitoring data. In *Proc KDD'12*, pages 1158–1166, 2012.

[26] H. Shan, G. Jiang, and K. Yoshihira. Extracting overlay invariants of distributed systems for autonomic system management. In *Proc IEEE Int Conf on Self-Adaptive and Self-Organizing Systems*, pages 41–50, 2010.

[27] A. Sharma, H. Chen, M. Ding, K. Yoshihira, and G. Jiang. Fault detection and localization in distributed systems using invariant relationships. In *Proc IEEE Int Conf on DSN'13*, pages 1–8, 2013.

[28] A. B. Sharma, F. Ivancic, A. Niculescu-Mizil, H. Chen, and G. Jiang. Modeling and analytics for cyber-physical systems in the age of big data. *SIGMETRICS Perform. Eval. Rev.*, 41(4):74–77, Apr. 2014.

[29] D. Suhr. Principal component analysis vs. exploratory factor analysis. In *Proceedings of SUGI 30*, pages 203–30, 2005.