A Robust Alternative to Correlation Networks for Identifying Faulty Systems

Patrick Traxler [1] and **Pablo Gómez**[2] and **Tanja Grill**[1]
[1]Software Competence Center Hagenberg, Austria
e-mail:patrick.traxler@scch.at
tanja.grill@scch.at
[2]Institute of Applied Knowledge Processing, Johannes Kepler University, Linz, Austria
e-mail: pablo.gomez@faw.jku.at

Abstract

We study the situation in which many systems relate to each other. We show how to robustly learn relations between systems to conduct fault detection and identification (FDI), i.e. the goal is to identify the faulty systems. Towards this, we present a robust alternative to the sample correlation matrix and show how to randomly search in it for a structure appropriate for FDI. Our method applies to situations in which many systems can be faulty simultaneously and thus our method requires an appropriate degree of redundancy. We present experimental results with data arising in photovoltaics and supporting theoretical results.

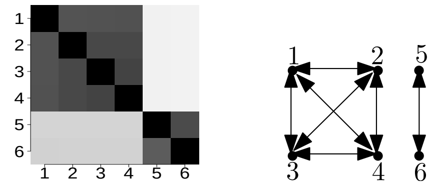
1 Introduction

The increasing number of technical systems connected to the Internet raises new challenges and possibilities in diagnosis. Large amount of data needs to be processed and analyzed. Faults need to be detected and identified. Systems exist in different configurations, e.g. two systems of the same type that have different sets of sensors. Knowledge about the system design is often incomplete. Data is often unavailable due to unreliable data connections. Besides these and other difficulties, the large amount of data also opens new possibilities for diagnosis based on machine learning.

The idea of our approach is to conduct fault detection and identification (FDI) by comparing data of similar systems. We assume to have data of machines, devices, systems of a similar type and want to know if some system is faulty and if so, to identify the faulty systems. This situation may deviate from classic diagnosis problems in that we just have limited information (e.g. sensor or control information) of system internals. Moreover, we may have incomplete knowledge about the system design. This makes manual system modeling hard or even impossible. The problem is then to compare the limited information of the working systems (perhaps only input-output information) to identify faulty systems.

In this work we tackle one concrete problem of this kind. It is motivated by photovoltaics. We describe it in more detail below. The problem that arises in our and other applications is that not every two systems can be compared. We thus need to learn relations between systems.

There are different approaches to learn structure, e.g. learning Bayesian networks, Markov random fields, or similar concepts. The concept that fits our needs are correlation networks. A correlation network is some structure in the correlation matrix, e.g. a minimum spanning tree or a clustering. In our application we have n variables which represent the produced energy per photovoltaic system. Given that a single system correlates strongly with enough other systems, we use this information for FDI via applying a median.



(a) Fitness matrix (b) Digraph

Figure 1: Learning relations between 6 systems. We draw an edge between two systems if there is a strong linear relation between them. First, we compute the fitness matrix, 1(a), our robust alternative to the sample correlation matrix. Darker colors mean a stronger linear relation. Going from Fig. 1(a) to 1(b) is a discretization step via thresholding. The digraph is the input for conducting FDI.

We can also think of correlation networks as a method for knowledge discovery. It has been applied in areas such as biology [18; 10] and finance [12] to analyze gene coexpression and financial markets. In our situation, the first step is to learn linear relations between systems. For learning we need historical data. A sample result of this step is depicted in Fig. 1. In Fig. 1(a) the fitness matrix, our robust alternative to the correlation matrix, is shown. It represents the degree of linearity between any two systems. For FDI, the second step of our method, we work with the result as depicted in 1(b) and current data. In the example, we derive for every of the six systems an estimation $\hat{m}_i$ of its current value y_i from its neighbors current values, e.g. for system 1 we get an estimate from the current values of the systems $2, 3, 4$ and for system 5 from system 6. Finally, we test for a fault by checking if $|\hat{m}_i - y_i|$ is large.

The major difficulty we try to tackle with this approach is the presence of many faults. Faults influence both the learning problem and the FDI problem. Robustness is an essential property of our algorithms. Our result can be seen as a robust structure learning algorithm for the purpose of FDI. Robustness is a preferable property of many learning and estimation algorithms. However, the underlying optimization problems unlike their non-robust variants are often NP-hard. This is for example the case for computing robust and non-robust estimators for linear regression, e.g. Least Median of Squares versus Ordinary Least Squares [16]. We avoid NP-hardness by a careful modeling of our problem. In particular, our algorithms are computationally efficient. Under some conditions, FDI can be done in (almost) linear time in the number of systems n.

To summarize our contributions, we introduce a novel alternative to the sample correlation matrix and present a first use of it to discover structure appropriate for general FDI and in particular for identifying faulty photovoltaic systems (PV). Our method works in the presence of many faults. Our algorithms are computationally efficient. Our method incorporates a couple of techniques from machine learning and statistics: (Repeated) Theil-Sen estimation for robust simple linear regression. Trimming to obtain a robust fitness measure. Randomized subset selection for improved running time. And a median mechanism to conduct FDI.

In Sec. 2 we discuss our method. In Sec. 3 we present experimental and theoretical results.

1.1 Motivating Application: Identifying Faulty Photovoltaic Systems

Faults influence the performance of photovoltaic systems (PV). PV systems produce less energy than possible if faults occur. We can distinguish between two kinds of faults. Faults caused by an exogenous event such as shading, (melting) snow, and tree leafs covering solar modules. And faults caused by endogenous events such as module defects and degradation, defects at the power inverter, and string disconnections.

We are going to detect faults by estimating the drop in produced energy. Most of the common faults result in such a drop. The particular problem is given by the sensor setup. We just assume to know the produced energy and possible but not necessarily the area (e.g. the zip code) where the PV system is located.

We apply our method to PV system data. Difficulties in the application are different system types and deployments of systems. For example, different number of strings and modules per string and differing orientation (north, west, south, east) of the modules; see Fig. 2. Moreover, the lack of information due to the lack of sensors and incomplete data due to unreliable data connections. Faults occur frequently, in particular exogenous faults during winter.

The novelty of our work in the context of photovoltaics is that it works in an extremely restrictive (only power measurement) sensor setting. To the best of our knowledge, we are the first to consider this restrictive sensor setting. We only need to know the produced energy of a PV system. There is also the implicit assumption, which is tested by the learning algorithm, that the systems are not too far from each other so that we can observe them in similar working (environmental) conditions. Distances of a couple of kilometers are possible. Systems which are very close to each other and have the same orientation such as systems in a solar power plant yield the best results. Other approaches assume the presence of a plane-of-array-irradiance sensor which are mostly deployed for solar power plants. Irradiance estimations via satellite imaging are usually not accurate enough.

1.2 Related Work

Correlation networks have applications in biology and finance. See e.g. [12; 18; 10] and the references therein. In biology [18; 10], they are applied to study gene interactions. The correlation matrix is the basis for clustering genes and the identification of biologically significant clusters. In [18; 10], a scale-free network is derived via the concept of topological overlap. Scale-free networks tend to have few nodes (genes) with many neighbors, so called hubs.

Correlation networks are primarily used for knowledge discovery. In particular, concepts such as clusters, hubs, and spanning trees are interpreted in the context of biology and finance. In our work, we introduce a robust alternative to correlation networks.

Other structural approaches, i.e. approaches based on graphical models, are based on Bayesian networks, Markov random fields and similar concepts. Gaussian Markov random fields are loosely related to correlation networks. Their structure is described by the precision matrix, the inverse covariance matrix (ch. 17.3, [9].)

Another structural approach is FDI in sensor networks [7; 4; 19; 20]. The current approach [7; 4; 19] mainly deals with wireless sensor networks. The algorithms usually use the median for FDI such as we do. The difference is that FDI in wireless sensor networks uses a geometric model similar to interpolation methods. It requires the geographic location of the sensors. It is assumed that two sensors close to each other have a similar value. This cannot be assumed in general. To overcome these problems of manual modeling, we apply machine learning techniques.

Models for PV systems are compared in [14]. All these models require the plane-of-array irradiance. Fault detection of PV systems is the topic of e.g. [3; 8; 5; 2; 17]. Firth et al. [8] consider faults if the PV system generates no energy. Another type of fault occurs if the panels are covered by snow, tree leaves, or something else. In this case, we can observe a drop in energy. It is considered e.g. in [5]. The fraction of panel area covered is a crucial parameter. All these approaches [3; 8; 5; 2; 17] require at least the knowledge of the plane-of-array irradiance, i.e. it requires an irradiance sensor installed. We do make this assumption.

The median is common in fault detection and identification. One reason for this circumstance is its optimal breakdown point [16]. We also make use of (repeated) Theil-Sen algorithms [6; 15] for learning. An ingredient of our fault identification algorithm is the algorithm for median selection [1] and an algorithm for generating uniform subsets efficiently (see e.g. the Fisher-Yates or Knuth shuffle in [13].) In our algorithm analysis we derive bounds for a partial Cauchy-Vandermonde identity (pg. 20 in [11]).

2 Method

2.1 Data Model for Incomplete Data

We have data from n systems and one data stream per system. A data stream for system $i \in \{1, \ldots, n\}$ is given by a set $N_i \subseteq \{1, \ldots, N\}$ of available data and values $x_{i,t} \in \mathbb{R}$ with $t \in N_i$. We can think of the parameter t as discrete time. With N_i, we explicitly model data availability. Incomplete data is a common problem in our situation. Causes in practice are unreliable data connections or unreliable sensors. We call $D := \{(x_{i,t})_{t \in N_i} : i \in \{1, \ldots, n\}\}$ a *data* *set.* Sets of historical and current data are the inputs to our algorithms.

2.2 Fitness Matrix – Definition and Robustness

The fitness matrix is intended as a robust replacement for the sample correlation matrix. The sample correlation coefficient such as the sample covariance is well known to be sensitive to faults (outliers) [16]. As an example, we generated the data for Fig. 1 with faults. The non-robust sample correlation matrix would have yield a digraph without edges instead of the digraph in Fig. 1(b).

A fault can be an arbitrary corruption of a single data item $x_{i,t}$. That is, $x_{i,t} = \tilde{x}_{i,t} + \Delta$, $\Delta \neq 0$, where Δ is the fault. We think of $\tilde{x}_{i,t}$ as the actual or true but unobserved value.

We do not make any assumptions on faults themselves but only on their number. This is at core of the definition of the breakdown point. This statistical concept is defined for a particular estimation or learning problem. In our case for simple linear regression.

Linear regression is closely related to the correlation coefficient. For simple linear regression – a regression model with one independent and one dependent variable – the correlation coefficient can be seen as a fitness measure of the line which fits the data best w.r.t. vertical squared distances. See e.g. [16]. However, the corresponding estimator, namely ℓ_2-regression a.k.a. ordinary least squares, is known to be sensitive to outliers [16]. On the other hand, there are estimators for simple linear regression which are robust to a large number of faults, i.e. they have a large breakdown point.

The idea underlying the fitness matrix is thus to replace the correlation coefficient (and ℓ_2-regression) by a robust notion of fitness based on robust linear regression. In the remainder of this section we recall the definition of the breakdown point following [16], pg. 9, and we are going to formalize the notion of fitness matrices.

We define the breakdown only for simple linear regression. We fix two systems $i, j \in \{1, \dots, n\}$ and define $Z := Z_{i,j} := \{(x_{i,t}, x_{j,t}) : t \in N_i \cap N_j\}$. Let T be a regression estimator, i.e. $T(Z_{i,j}) = \hat{\theta} \in \mathbb{R}^2$ is the intercept and slope for the data set $Z_{i,j}$. For Z, we define Z' as Z with m data points arbitrarily corrupted. Define

$$\mathsf{bias}(m; T, Z) := \sup_{Z'} \|T(Z) - T(Z')\|.$$

If $\mathsf{bias}(m; T, Z)$ is infinite, then m faults (outliers) have an arbitrarily large effect on the estimate $T(Z')$. The *(finite sample) breakdown point* of T and Z is defined as

$$\varepsilon^*(T, Z) := \min\left\{\frac{m}{|Z|} : \mathsf{bias}(m; T, Z) = \infty\right\}.$$

To explain this notion, we consider four typical examples. The breakdown point $\varepsilon^*(T_{\ell_2}, Z)$ is $1/n$ for ℓ_2-regression. This holds for any Z. The situation is different for ℓ_1-regression in that $\varepsilon^*(T_{\ell_1}, Z) = 1/n$ for some Z.

In this work we are going to use the Theil-Sen estimator[1] T^{TS} a.k.a. median slope selection. The reason is its breakdown point of at least $1 - \frac{1}{\sqrt{2}} \geq 0.292$ (see e.g. [6]) and the wide availability of efficient implementations of near-linear time algorithms. There is also a variant of T^{TS}, called the repeated Theil-Sen estimator, which has a breakdown point of 0.5, but less efficient implementations. The concrete definition of T^{TS} can be found e.g. in [6]. It is however not important for our application, only its robustness property and the existence of efficient implementations are.

To define the breakdown point of a fitness matrix, let f be a real-valued function defined on any finite data set. We define the *fitness matrix* as

$$F_{ji} := f(Z_{i,j})$$

and its breakdown point as

$$\varepsilon^*(F) := \min_{i,j} \varepsilon^*(f, Z_{i,j}).$$

Next, we provide the fitness matrix we are going to use. It has the property that F_{ji} is close to zero if x_i and x_j are strongly linearly related and it has a high breakdown point.

Let $y_t := x_{i,t}$ and $\hat{y}_t := x_{j,t} \cdot \hat{\theta}_2 + \hat{\theta}_1$, $t \in N_i \cap N_j$, for the Theil-Sen estimate $\hat{\theta}$ of Z_{ij}. Let $r_t := \hat{y}_t - y_t$ be the residuals. And let $i_1, \dots, i_k \in N_i \cap N_j$ with $k := |N_i \cap N_j|$ be such that $|r_{i_1}| \leq \dots \leq |r_{i_k}|$. We define

$$f^{\mathsf{TS}}(Z_{ij}) := \frac{1}{\sum_{t=1}^{\lfloor k/\sqrt{2} \rfloor} |y_{i_t}|} \cdot \sum_{t=1}^{\lfloor k/\sqrt{2} \rfloor} |r_{i_t}|. \qquad (1)$$

We define F^{TS} w.r.t. f^{TS}, i.e.

$$F^{\mathsf{TS}}_{ji} := f^{\mathsf{TS}}(Z_{i,j}).$$

Note that the sum goes from 1 up to $\lfloor k/\sqrt{2} \rfloor$. This trimming together with the high breakdown point of Theil-Sen directly implies the following result.

Theorem 1. *It holds that $\varepsilon^*(F^{\mathsf{TS}}) \geq 1 - \frac{1}{\sqrt{2}}$.*

Finally, we compare the sample correlation matrix and the fitness matrix. Let C denote the sample correlation matrix and define $C'_{ji} = 1 - |C_{ji}|$. Both matrices have the property that if some entry is close to 0 then x_i and x_j have a strong linear relation. It is guaranteed that C'_{ji} is at most 1. A value close to 1 means a weak linear relation. For F^{TS}_{ji}, it is not guaranteed that $F^{\mathsf{TS}}_{ji} \leq 1$, but experimental results suggest that it is usually the case. We also note that both matrices obey a weak form of the triangle inequality since if x_i and x_j correlate strongly and x_j and x_k correlate strongly, then also x_i and x_k correlate.

There are two important benefits of fitness matrices over correlation matrices. They are robust and are also defined for incomplete data. On the negative side, the fitness matrix is not positive semi-definite, in particular not symmetric.

2.3 Structure in Fitness Matrices – Algorithm LEARN and IDENTIFY

We want to identify faulty systems. In a first step, we learn a structure appropriate for FDI; see algorithm LEARN. We obtain it via thresholding the fitness matrix. Most of the correlation networks, i.e. structures arising from the sample correlation matrix, are obtained in this way [12; 18; 10]. We denote the threshold by $\theta \geq 0$ and the *threshold fitness matrix* as

$$F_{ji;\theta} := \begin{cases} F_{ji} & \text{if } F_{ji} \leq \theta \\ 0 & \text{if } F_{ji} > \theta \end{cases}.$$

[1] There is a subtle issue here we have to deal with. Regression problems are optimization problems. The solution to the concrete optimization problem does not need to be unique. In our situation, intercept and slope are unique for ℓ_2-regression but not for ℓ_1-regression. The estimator T_{ℓ_1} is however unique for a (deterministic) algorithm solving the optimization problem. We thus think of T_{ℓ_1} as the output of a particular (deterministic) algorithm.

Algorithm 1 Algorithm LEARN with input D (data set) and parameter θ (fitness threshold). Output is a digraph G with edge labels (intercept, slope) representing the threshold fitness matrix.

```
Let $G = (V, E)$ be a digraph with $V = \{1, \ldots, n\}$ and $E = \{\}$.
for all $i \in V$ and $j \in V \setminus \{i\}$ do
    Learn (Theil-Sen) the intercept $a_{j,i}$ and slope $b_{j,i}$ between $x_i$ (dependent variable) and $x_j$ (independent variable).
end for
for all $i \in V$ and $j \in V \setminus \{i\}$ do
    Compute the trimmed fitness $f = f^{\mathsf{TS}}$ (Eq. 1) of $Z_{i,j}$.
end for
if $f \leq \theta$ then
    Add to $G$ the directed edge from $j$ to $i$ with edge labels $(a_{j,i}, b_{j,i})$.
end if
```

The input to algorithm LEARN is a data set D as described in Sec. 2.1. It outputs a digraph $G = (V, E)$, i.e. the (possible sparse) threshold fitness matrix F_θ^{TS}. Additionally, intercept and slope of the simple linear regressions are added as edge labels.

Algorithm 2 Algorithm IDENTIFY with input G (digraph with edge labels), current data y_i for the i-th system, and parameters k and s (deviation). It outputs the set of all faulty systems H.

```
Set $H = \{\}$.
for all $i \in V = \{1, \ldots, n\}$ do
    Let $N_i^- := \{j \in V : (j, i) \in E\}$.
    if $|N_i^-| = 0$ then
        Continue with the next (system) $i$.
    end if
    if $|N_i^-| \geq k$ then
        Select uniformly at random a $k$-element
        subset $S$ from $N_i^-$.
    else
        Set $S := N_i^-$.
    end if
    Compute $M_i := \{\hat{y}_j = b_{j,i} \cdot y_j + a_{j,i} : j \in S\}$.
    Compute the median $\hat{m}_i$ of $M_i$.
    Add $i$ to $H$ if $|\hat{m}_i - y_i| > s$
end for
Output $H$.
```

In the second step, we identify the faulty systems; see algorithm IDENTIFY. Its input is the result of algorithm LEARN. Algorithm IDENTIFY constructs a random digraph of in-degree at most k for FDI. It works as follows. Independently for every system, we choose uniformly at random at most k of its neighbors in the digraph G and compute the median $\hat{m}_i$ of estimated values derived from the selected neighbors values. We compare the median $\hat{m}_i$ to the current system value and decide whether it has a fault or not via the deviation parameter s.

We discuss the threshold parameter θ and the deviation parameter s in Sec. 3.1. They essentially depend on the variance in the data set D. Parameter k in algorithm IDENTIFY has the purpose of improving running time efficiency. In particular, we have the following result.

Theorem 2. *Let D be a data set with n systems and let $m := \max_i |N_i|$. The running time of LEARN is $O(n^2 \cdot m \cdot \log(m))$. The running time of IDENTIFY is $O(k \cdot n)$.*

Proof. LEARN. There are $O(n^2)$ pairs of systems. The Theil-Sen estimator can be computed in time $O(m \log(m))$ [6]. The computation of f^{TS}, Eq. 1, is done via sorting and thus takes time $O(m \log(m))$.

IDENTIFY. Assume $|N_i^-| \geq k$. We uniformly at random choose a k-element subset out of N_i^- and compute the median. For random selection we can use for example the Fisher-Yates (or Knuth) shuffle [13] which runs in time $O(k)$ and for median selection the algorithm in [1] which also runs in time $O(k)$. The second case, $1 \leq |N_i^-| \leq k-1$, is analogous. This shows that the overall running time of IDENTIFY is $O(kn)$. □

In Sec. 3.2, we provide some sufficient conditions that IDENTIFY works correctly even if $k = O(\log(n))$. This is a strong running time improvement from $O(n^2)$ to $O(n \cdot \log(n))$.

3 Results

3.1 Experimental

In this section we are going to discuss how to apply our method, Sec. 2, to photovoltaic data. In particular, it remains to discuss how the use-case fits to the model. More precisely, why there is strong correlation between PV systems. Finally, we present experimental results to verify the estimation and fault identification quality of our algorithms.

Use-Case Photovoltaics

A simple system model for PV systems is as follows:

$$P_i = c_i \cdot I_i.$$

Here, P_i is the power, I_i the plane-of-array (POA) irradiance of the i-th system, and c_i a constant of the system which can be interpreted as the efficiency of converting solar energy into electrical energy. More complex physical models include system variables such as the module temperature [14; 17]. Our considerations translate to the more complex models as long as they are time-independent. We also note that these models are more accurate, but only slightly, since the POA-irradiance has the most critical influence on the produced energy.

We get from the above considerations that $P_i = c'_{ij} \cdot P_j$ given that $I_i = I_j$. In our situation we cannot test the condition $I_i = I_j$ since we do not know the POA-irradiance, but $I_i \approx I_j$ holds if the system operate under similar weather conditions and have a similar orientation. The former holds if the systems are close to each other. To reduce the effect of different orientations, see Fig. 2, we consider the following model: $P_i^\Delta = u_{ij} \cdot P_j^\Delta + v_{ij}$. The variable P_i^Δ is the power within a time interval Δ, usually one hour. The variables u_{ij} and v_{ij} are the unknowns.

In more general words, let Y_i be the output of the i-th system and let X_i describe the system input and system internals. Our model assumption is that for a reasonable number of system pairs (i, j), the system outputs Y_i are Y_j are linearly related given that $X_i \approx X_j$. By the above considerations, it is plausible that PV systems fulfill these requirements.

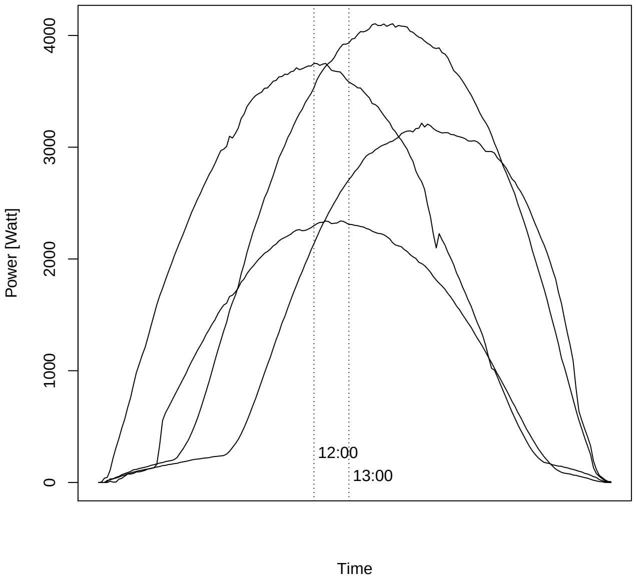


Figure 2: Four power curves of a sunny day in August, data set D_{K}. Two PV systems have their maximum power peak before and the other two after 13:00. They have different orientations, i.e. they produce more energy in the morning or evening.

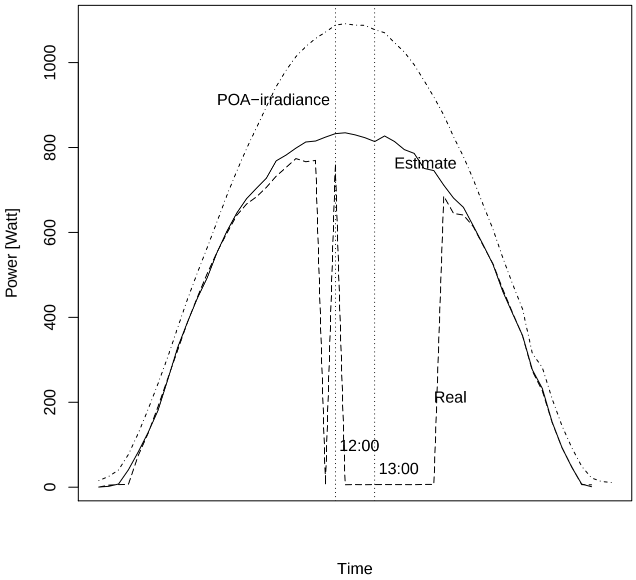


Figure 3: A faulty system. The real power curve of observed values shows a fault from roughly 11:00 to 14:30. The estimated values are considerable higher during this period. The PV system has a plane-of-array irradiance sensor installed. A cross check with its power curve reveals that the fault was detected correctly.

We next describe our experimental setup to verify it by real data.

Experimental Setup

To demonstrate our method, we use two data sets D_{A} and D_{K}. D_{A} arises from 13 systems from a solar park located in Arizona[2]. The PV systems there are geographically close. We use data for one year. D_{K} arises from 40 systems spread across a typical municipality located in Austria, i.e. the systems can be up to some kilometers apart. Their orientation can differ significantly. Some systems may be orientated to the west, others to the east. We have data for almost a year.

A system is faulty if it produces considerable less energy than estimated; see Fig. 3. This definition is motivated by the fact that most faults imply a drop in energy. The difficulty in setting up an experiment is that we do not know if a PV system is faulty in advance, i.e. we do not have labeled data. We thus design our experiment as follows: We verify the accuracy of the energy estimation, namely the relative deviation $|\hat{m}_i - y_i|/|\hat{m}_i|$ for every system i and over the period of a week, $\hat{m}_i$ and y_i as in algorithm IDENTIFY.

This relative deviation is noted in column *Hour* of Table 1 for the time period 12:00 to 13:00. In column *Day* of Table 1 we note the same but for a whole day, i.e. $\hat{m}_i$ is the estimated energy (power) for the whole day calculated from the hourly estimates and y_i the actual energy for the whole day. For the whole day we consider the time period from 9:00 to 16:00.

The number $|\hat{m}_i - y_i|/|\hat{m}_i|$ can be read as some relative deviation, i.e. the estimation is $100 \cdot x\%$ away from the truth value where x is some entry in the column *Hour* and *Day*. The entry x is an average over all systems and 7 days. The first day is noted in column *Start*.

Algorithm LEARN is executed once for every week and with $\theta = 0.8$ and roughly three months of historical data, e.g. for the months January, February, and March to get estimates for the days April, 1. to April, 7. Algorithm IDENTIFY is executed with $s = 0.25 \cdot |\hat{m}_i|$ and $k = 11$ for both data sets. The choice of parameters θ and s depend on the variance of the input data and were chosen manually, so to get a reasonable number of good estimates. Similar for k. The difficulty in choosing the parameters is that increasing θ will usually reduce the number of neighbors. For a reasonable number of good estimates we need both: A strong linear relation of a system to its neighbors and enough neighbors. The parameters were chosen accordingly. For parameter k, we derive a theoretical result in Sec. 3.2 which says that $k = O(\log(n))$ is a good choice for n the number of systems.

Experimental Results

The false positive rate (FPR), the false negative rate (FNR), and the estimation accuracy are the most interesting numbers for us. As remarked above, we do not have labeled data. The faults as recorded in Table 1 are faults as detected by our algorithm.

We make a worst case assumption, namely that all detected faults are false positives. This yields a FPR of at most 0% to 5% per 7 day period (rows in the table.) To get an understanding of FNR, we simulated faults by subtracting 33% percent of energy. The FNR in this case is at most 10% per 7 day period. In the rows *Sum* and *Sum*−33% in Table 1 we summed up the faults to get the FPR and FNR for the whole

[2] http://uapv.physics.arizona.edu/

(a) Results for D_{A}.

Start	*Hour*	*Faults*	*Day*	*Faults*
April, 1.	0.058	2/77	0.037	1/77
May, 1.	0.040	2/77	0.014	0/77
June, 1.	0.019	0/91	0.021	0/91
July, 1.	0.068	7/88	0.071	7/88
Aug., 1.	0.362	12/91	0.250	7/91
Sept., 1.	0.096	4/65	0.034	1/78
Oct., 1.	0.019	0/84	0.016	0/84
Nov., 1.	0.039	0/72	0.025	0/72
Dec., 1.	0.135	10/84	0.130	7/84
Sum		37/729		23/742
Sum-33%		673/729		682/742

(b) Results for D_{K}.

Start	*Hour*	*Faults*	*Day*	*Faults*
June, 1.	0.056	7/269	0.068	11/273
June, 15.	0.055	7/238	0.097	17/258
July, 1.	0.077	7/267	0.068	9/280
July, 15.	0.025	0/267	0.044	6/280
Aug., 1.	0.037	2/279	0.030	3/280
Aug., 15.	0.031	1/280	0.032	0/280
Sept., 1.	0.040	0/280	0.033	0/280
Sept., 15.	0.092	20/280	0.056	0/280
Sum		42/2160		46/2211
Sum-33%		1960/2154		2033/2207

Table 1: The values in column *Hour* and *Day* contain the relative deviation $|\hat{m}_i - y_i|/|\hat{m}_i|$, $\hat{m}_i$ and y_i as in algorithm IDENTIFY. They are averages over all systems and the period of a week. Column *Start* contains the start date of the 7 day period. The two columns labeled *Faults* contain the number of (possible false detected) faults relative to the total number of analyzed hours and days, respectively. The rows *Sum* contain the summed up number of faults, once for the actual data sets and then with a simulated fault of -33% less energy.

data sets.

The interpretation of these results is as follows. Setting the parameter s to $0.25 \cdot |\hat{m}_i|$ means that we define a fault as a 25% relative deviation of the observed produced energy from its true value. Setting s to this value, yields the above mentioned FPR. Simulating a 33% drop in energy, which corresponds naturally to the faults we want to detect, yields the above FNR.

For the data set D_{A} we have knowledge about the POA-irradiance. We can thus cross-check with the irradiance to check if faulty systems were identified correctly; see Fig. 3. This manual inspection suggests that the FPR is much smaller than 5%, close to 1%. Furthermore, increasing the drop implies a decreasing FNR, i.e. stronger energy drops are easier to identify.

Depending on the application, these rates may be considered appropriate or not. In some applications, we may want to detect faults which yield a drop in energy of less -25%. This worsens the FPR and FNR. On the other side, if we want to improve the FPR and FNR, we may have to specify a fault as a drop in energy of -50%. In other words, our parameter setting is one out of many reasonable parameter settings.

3.2 Theoretical

We argued in Sec. 3.1 that algorithm LEARN yields good estimates for the systems current value. For an estimate to be good, the neighboring system j in G of system i needs to work correctly. Moreover, the regression estimates, the intercept and slope, need to be accurate enough. In this section, we provide a supporting theoretical result which says that, if enough estimates are good, algorithm IDENTIFY correctly identifies all faulty systems.

The input to IDENTIFY is a digraph $G = (V, E)$ with edge labels. Let y_i be the current value of system i. Let $y_i = \tilde{y}_i + \Delta_i$. We think of $\tilde{y}_i$ as the true value. We say that system i is *correct* if $\Delta_i = 0$ and *faulty* otherwise.

The input to IDENTIFY has to satisfy two conditions, Eq. 2 and 3, to work correctly. These conditions state that there are more good than bad estimates. We formulate them below.

Theorem 3. *Let $0 < p < 1$ and $s > 0$. Let $H := \{i \in \{1, \dots, n\} : |\Delta_i| > 2s\}$. Assume that the input digraph G satisfies Eq. 2 and 3. Then, algorithm IDENTIFY outputs H with probability at least $1 - p$.*

Let $\hat{y}_j$ be the estimates as computed in IDENTIFY. Fix a system i and let $j \in N_i^-$. We say that $\hat{y}_j$ is *s-good* for system i if $|\tilde{y}_i - \hat{y}_j| \le s$. Let $A_i := \{j \in N_i^- : |\tilde{y}_i - \hat{y}_j| \le s\}$ be the s-good estimates for system i. Condition 2 is as follows: For every system i with $1 \le |N_i^-| \le k - 1$ it holds that

$$|A_i| > \frac{|N_i^-|}{2}, \tag{2}$$

i.e. there are more good than bad estimates. For the case that $|N_i^-| \ge k$ we assume

$$|A_i| > \left(1 - \frac{1}{c_{n,p,k}}\right) \cdot |N_i^-|, \tag{3}$$

with $c_{n,p,k} := (\frac{n}{p} \cdot 18^k)^{2/(k-1)}$. Setting $k = \Omega(\log(\frac{n}{p}))$ makes $c_{n,p,k}$ larger than some constant *independent* of n and p. This is the most reasonable setting as it implies that a constant fraction of estimates can be bad and IDENTIFY still identifies the faulty systems correctly. We remark that the asymptotic analysis which yields $c_{n,k,p}$ is not optimal. In particular, it seems that the factor 18^k is not optimal and may be improved to a factor as small as $2^{k/2}$. For practical applications, the following heuristic seems reasonable: For n systems and a failure probability p of IDENTIFY, set k to $10 \cdot \log(\frac{n}{p})$.

3.3 Proof of Theorem 3

We apply the following lemma with $A = G_i$ and $M = N_i^-$. It directly gives us the probability that IDENTIFY correctly identifies the faulty systems since the median works correctly if $|S \cap A_i| > |S \cap (N_i^- \setminus A_i)|$, where S is the (random) set chosen in IDENTIFY.

Lemma 1. *Let M be a finite set and $A \subseteq M$. Let $k \ge 2$ be an integer. Let $S \subseteq M$ be a k-element subset selected uniformly at random. Then*

$$\Pr_S(|S \cap A| > |S \cap (M \setminus A)|) \ge 1 - 18^k \left(\frac{|M \setminus A|}{|M|}\right)^{\lfloor k/2 \rfloor}.$$

Proof. Let $M := \{1, \dots, m\}$, $F := M \setminus A$, and $r := |F|$. First, we are going to bound the number of k-element subsets $S \subseteq M$ for which $|S \cap G| \leq k'$ with $k' = \lfloor k/2 \rfloor$. The exact number of these sets is

$$\sum_{i=0}^{k'} \binom{m-r}{i} \binom{r}{k-i} \tag{4}$$

since there are $\binom{|A|}{i}$ ways to choose an i-element subset from A and $\binom{|F|}{k-i}$ ways to choose from F for the remaining $k-i$ elements.

Note that $|S \cap A| > |S \cap F|$ iff $|S \cap A| > \lfloor |S|/2 \rfloor = k'$. Moreover, we can assume that $r = |F| \geq 1$ since the claim holds for $r = 0$. To provide a lower bound for the probability of this event we show an upper bound on the complementary event, i.e. $|S \cap A| \leq k'$. First, we derive an upper bound for Eq. 4 using

$$\left(\frac{m}{k}\right)^k \leq \binom{m}{k} \leq \left(\frac{me}{k}\right)^k \tag{5}$$

for $e = 2.714\dots$ and $1 \leq k \leq m$. (See e.g. pg. 12 in [11].) Since this inequality holds just for $k \geq 1$ we rewrite Eq. 4 as

$$\binom{r}{k} + \sum_{i=1}^{k'} \binom{m-r}{i} \binom{r}{k-i}. \tag{6}$$

It holds that $\binom{r}{k} \leq (\frac{re}{k})^k$ and for the second term in Eq. 6

$$\sum_{i=1}^{k'} \binom{m-r}{i} \binom{r}{k-i} \leq \sum_{i=1}^{k'} \left(\frac{(m-r)e}{i}\right)^i \left(\frac{re}{k-i}\right)^{k-i}$$

$$= (re)^k \sum_{i=1}^{k'} \left(\frac{m-r}{r}\right)^i \left(\frac{k-i}{i}\right)^i \left(\frac{1}{k-i}\right)^k$$

Next, we prove the upper bound on the probability p that $|S \cap A| \leq k'$. We select uniformly at random a k-element subset of M. Its probability is $\binom{m}{k}^{-1}$. We multiply Eq. 6 with $\binom{m}{k}^{-1}$ and get two parts $p_1 + p_2 \geq p$. For the first part $p_1 \leq (\frac{re}{m})^k$ since $\binom{m}{k}^{-1} \leq (k/m)^k$. For the second part p_2, we use $\frac{m-r}{r} \leq \frac{m}{r}$, $((k-i)/i)^i \leq 2^k$, and $(k/(k-i))^k \leq 2^k$. The latter since $i \leq k'$. We get an upper for the second part:

$$p_2 \leq \left(\frac{re}{m}\right)^k \sum_{i=1}^{k'} \left(\frac{m-r}{r}\right)^i \left(\frac{k-i}{i}\right)^i \left(\frac{k}{k-i}\right)^k \leq$$

$$\left(\frac{12r}{m}\right)^k \sum_{i=1}^{k'} \left(\frac{m}{r}\right)^i.$$

An upper bound for the geometric sum is $k'(m/r)^{k'}$. In total

$$p \leq p_1 + p_2 \leq \left(\frac{re}{m}\right)^k + k'\left(\frac{12r}{m}\right)^k \left(\frac{m}{r}\right)^k.$$

Substituting $\frac{k-1}{2}$ for k' and further simplification yields

$$p \leq \frac{(k+2)(12)^k}{2} \left(\frac{r}{m}\right)^{(k-1)/2} \leq 18^k \left(\frac{r}{m}\right)^{(k-1)/2}.$$

The latter since $((k+2)/2)^{1/k} \leq 1.5$ for $k \geq 3$. We have thus a lower bound for the probability $1-p$ and the claim follows. □

Proof of Theorem 3. We show that the success probability of IDENTIFY is at least $1-p$. Let $p' := \frac{p}{n}$. We show that for every $i \in V$, $G = (V, E)$, the success probability of a single iteration in the loop of IDENTIFY is at least $1 - p'$. This implies the above claim since $(1-p')^n \geq 1 - p'n$ by e.g. the Binomial Theorem.

Fix some $i \in V$, i.e. we consider one iteration in the loop of IDENTIFY. We apply Lemma 1. Let us assume that $|S \cap A_i| > |S \cap (N_i^- \setminus A_i)|$, S the random k-element subset as in IDENTIFY and A_i the good estimates as defined above. Since $|S \cap A_i| > |S \cap (N_i^- \setminus A_i)|$, it follows that $|\hat{m}_i - \tilde{y}_i| \leq s$ for the median $\hat{m}_i$ as computed in IDENTIFY and $y_i = \tilde{y}_i + \Delta_i$.

Assume $\Delta_i = 0$, i.e. system i works correctly. Then, $|\hat{m}_i - y_i| = |\hat{m}_i - \tilde{y}_i| \leq s$. Thus, i is not output.

Assume $\Delta_i \neq 0$, i.e. system i is faulty. Here, $|\hat{m}_i - y_i| = |\hat{m}_i - \tilde{y}_i - \Delta_i|$. It follows from $|\hat{m}_i - \tilde{y}_i| \leq s$ and $|\Delta_i| > 2s$ that $|\hat{m}_i - y_i| > s$. Thus, i is output.

Finally, we want that the probability of failure for a single step is at most $\frac{p}{n}$. By Lemma 1, $18^k \alpha^{\lfloor k/2 \rfloor} \leq 18^k \alpha^{(k-1)/2} \leq \frac{p}{n}$ with $\alpha := \frac{|N_i^- \setminus A_i|}{|N_i^-|}$. With $c = c_{n,p,k} := (\frac{n}{p} \cdot 18^k)^{2/(k-1)}$, $c \cdot |N_i^- \setminus A_i| \leq |N_i^-|$ and thus $(1 - 1/c) \cdot |N_i^-| \leq |A_i|$. □

4 Conclusions and Open Problems

We presented a method for learning structure to identify faulty systems. The basic method of correlation networks has found many applications in biology and finance. In our application, the presence of many faults required the design and analysis of robust algorithms. We provided an experimental analysis of our algorithms to verify their estimation and fault identification quality. We also provided a supporting theoretical result which allowed us to considerable improve the running time of algorithm IDENTIFY.

Improving the running time of LEARN remains as an open problem. It is not directly clear that it is necessary to compare every two systems. The reason is that if systems (i, j) and (j, k) correlate strongly, then also (i, k) correlate, but not necessarily strongly. Thus, it may not be necessary to solve a simple linear regression problem for every system pair.

In other applications it may be useful to solve a general linear regression problem instead of a simple linear regression, e.g. if our model depends on more than one variable per system. The corresponding correlation networks are based on the partial correlation coefficient [12]. Since robust estimators for general linear regression are based on regression problems which are NP-hard, it remains as an open problem to find a robust alternative to partial correlation networks that can be computed efficiently.

Finally, to put our method and results into a broader context, we approached the problem of FDI via learning graphical models. It seems to be a challenge to learn classical component-models of technical systems to conduct diagnosis. In this work we were able to close the gap between (structure) learning on the one side and FDI on the other side for a concrete problem setting.

References

[1] Manuel Blum, Robert W. Floyd, Vaughan Pratt, Ronald L. Rivest, and Robert E. Tarjan. Linear time

bounds for median computations. In *Proc. of the 4th Annual ACM Symposium on Theory of Computing*, pages 119–124, 1972.

[2] H. Braun, S. T. Buddha, V. Krishnan, A. Spanias, C. Tepedelenlioglu, T. Yeider, and T. Takehara. Signal processing for fault detection in photovoltaic arrays. In *37th IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1681–1684, 2012.

[3] K. H. Chao, S. H. Ho, and M. H. Wang. Modeling and fault diagnosis of a photovoltaic system. *Electric Power Systems Research*, 78(1):97–105, 2008.

[4] Jinran Chen, Shubha Kher, and Arun Somani. Distributed fault detection of wireless sensor networks. In *Proc. of the 2006 Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks*, pages 65–72, 2006.

[5] A. Chouder and S. Silvestre. Fault detection and automatic supervision methodology for PV systems. *Energy Conversion and Management*, 51:1929–1937, 2010.

[6] R. Cole, J.S. Salowe, W.L. Steiger, and E. Szemeredi. An optimal-time algorithm for slope selection. *SIAM Journal on Computing*, 18(4):792–810, 1989.

[7] M. Ding, Dechang Chen, Kai Xing, and Xiuzhen Cheng. Localized fault-tolerant event boundary detection in sensor networks. In *Proc. of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 902–913, 2005.

[8] S.K. Firth, K.J. Lomas, and S.J. Rees. A simple model of PV system performance and its use in fault detection. *Solar Energy*, 84:624–635, 2010.

[9] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*. Springer, 2008.

[10] Steve Horvath. *Weighted network analysis: applications in genomics and systems biology*. Springer Science & Business Media, 2011.

[11] S. Jukna. *Extremal combinatorics: with applications in computer science*. Springer, 2nd edition, 2011.

[12] Dror Y. Kenett, Michele Tumminello, Asaf Madi, Gitit Gur-Gershgoren, Rosario N. Mantegna, and Eshel Ben-Jacob. Dominating clasp of the financial sector revealed by partial correlation analysis of the stock market. *PLoS ONE*, 5(12):e15032, 12 2010.

[13] Donald E. Knuth. *The art of computer programming: seminumerical algorithms*, volume 2. Addison-Wesley Longman Publishing Co., Inc., 3rd edition, 1997.

[14] B. Marion. Comparison of predictive models for PV module performance. In *33rd IEEE Photovoltaic Specialist Conference*, pages 1–6, 2008.

[15] J. Matousek, D. M. Mount, and N. S. Netanyahu. Efficient randomized algorithms for the repeated median line estimator. *Algorithmica*, 20(2):136–150, 1998.

[16] Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*, volume 589. John Wiley & Sons, 2005.

[17] P. Traxler. Fault detection of large amounts of photovoltaic systems. In *Online Proc. of the ECML PKDD 2013 Workshop on Data Analytics for Renewable Energy Integration (DARE'13)*, 2013.

[18] Bin Zhang and Steve Horvath. A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, 4(17), 2005.

[19] Chongming Zhang, Jiuchun Ren, Chuanshan Gao, Zhonglin Yan, and Li Li. Sensor fault detection in wireless sensor networks. In *Proc. of the IET International Communication Conference on Wireless Mobile and Computing*, pages 66–69, 2009.

[20] Yang Zhang, N. Meratnia, and P. Havinga. Outlier detection techniques for wireless sensor networks: a survey. *Communications Surveys and Tutorials, IEEE*, 12(2):159–170, 2010.