A local multiscale probabilistic graphical model for data validation and reconstruction, and its application in industry

Javier Herrera-Vega[a,*], Felipe Orihuela-Espina[a], Pablo H. Ibargüengoytia[b], Uriel A. García[b], Dan-El Vila Rosado[a], Eduardo F. Morales[a], Luis Enrique Sucar[a]

[a]*Instituto Nacional de Astrofísica Óptica y Electrónica, Puebla, México*
[b]*Instituto de Investigaciones Eléctricas, Cuernavaca, México*

Abstract

The detection and subsequent reconstruction of incongruent data in time series by means of observation of statistically related information is a recurrent issue in data validation. Unlike outliers, incongruent observations are not necessarily confined to the extremes of the data distribution. Instead, these rogue observations are unlikely values in the light of statistically related information. This paper proposes a multiresolution Bayesian network model for the detection of rogue values and posterior reconstruction of the erroneous sample for non-stationary time-series. Our method builds local Bayesian Network models that best fit to segments of data in order to achieve a finer discretization and hence improve data reconstruction. Our local multiscale approach is compared against its single-scale global predecessor (assumed as our gold standard) in the predictive power and of this, both error detection capabilities and error reconstruction capabilities are assessed. This parametrization and verification of the model are evaluated over three synthetic data source topologies. The virtues of the algorithm are then further tested in real data from the steel industry where the afore-

[*] Corresponding author

Email addresses: vega@ccc.inaoep.mx (Javier Herrera-Vega), f.orihuela-espina@ccc.inaoep.mx (Felipe Orihuela-Espina), pibar@iie.org.mx (Pablo H. Ibargüengoytia), uriel.garcia@iie.org.mx (Uriel A. García), dnvr301080@ccc.inaoep.mx (Dan-El Vila Rosado), emorales@inaoep.mx (Eduardo F. Morales), esucar@ccc.inaoep.mx (Luis Enrique Sucar)

mentioned problem characteristics are met but for which the ground truth is unknown. The proposed local multiscale approach was found to dealt better with increasing complexities in data topologies.

Keywords: Bayesian networks, data validation, multiscale approach, outlier detection, probabilistic graphical models

1. Introduction

Many areas like industry, medicine and science generate large volumes of data demanding validation. Validating data is a crucial task before information analysis, interpretation and decision making. Data validation encompasses processing techniques rendering quality data guaranteeing optimal matching between real observations and the repository. In other words, data validation is concerned with finding erroneous data in a data set and when appropriate, suggesting a plausible alternative [42]. The data validation process involves a systematic assessment of compliance to a set of acceptance rules defining data validity [21]. In general, the validation process is domain specific [18, 29]. Due to its domain specific nature, data validation is carried out not in few occasions by means of visual inspection, a time consuming approach, exposed to subjectiveness and prone to errors. Yet regardless of the particularities in each domain, a number of problems are recurrent in data validation including *detection* of outliers, incongruent or rogue values and/or gaps or missing data, and *reconstruction* or estimation of these missing or erroneous observations [24].

For some of these common data validation problems automation has been attempted [2, 4, 3, 22, 35, 43, 45]. These problems include the detection of observational outliers (values at the extreme of the data distribution), the detection of signal drift, level shift or abrupt changes altering the trend of the series (innovation outliers [33, 2]), the detection of rogue values (unplausible values in the light of statistically dependent information) and the reconstruction of missing data, among others. Once an error has been detected, the validation process proceeds with the data reconstruction. Reconstruction can capitalize on the signal autoregressive information e.g., classical interpolation [41] or time series analysis [5], statistically dependent information e.g., [29, 23], or a combination of both [24]. The most beneficial reconstruction option depends on the interplay between the variables characteristics in the dataset, including the within-variable information [24].

This paper is concerned with the detection of incongruent values and its reconstruction paying particular attention to datasets with temporal variables (time series). Incongruent or rogue values are suspicious values which may be in range and apparently agree with the signal trend, but that contradicts the associated trend of statistically dependent knowledge [21]. This makes their detection particularly difficult if using only within-variable information. This is inherently a multivariate problem. Previously, the detection of rogue values has been addressed with Bayesian Networks (BNs) in the context of sensor validation [23]. This approach is a *global* solution in which a BN is learned from the complete available dataset and then data validity is checked against probabilistic plausibility. Subsequent reconstruction utilizes probabilistic propagation to estimate expected values for erroneous samples from associated values in statistically related variables [23]. Learning the structure of the BN requires discretization of variables' data ranges into intervals that ultimately determines the detection rate and affects the accuracy of the recovery of alternative values. For stationary signals it is fair that these discrete intervals remain constant for the whole time series. However, for non-stationary signals, as the statistical properties of the series fluctuate, so should the intervals. In this way, dynamic finer discretization can be achieved and consequently a more accurate suggestion of alternative values should follow. Achieving similar discretization with a global solution will imply higher number of intervals, which in turn will require conditional probability tables that will grow exponentially. This quickly becomes computationally intractable.

This paper proposes a new *local* multiscale BN-based approach for the detection and reconstruction of incongruent values in multivariate datasets that we hypothesize to be more suitable for non-stationary signals. The algorithm constructs a two level hierarchy of BN models in which the superior level determines the dataset topology i.e., BN structure, and the inferior level contains a set of submodels providing interval discretizations that locally fits data distribution. In detecting the error and reconstructing the new value, the critical step is deciding the submodel that better fits the sample under scrutiny. The problem of selecting the submodel is solved computing the conditional probability of the observation given the submodel. The solution aims to enhance error detection and suggestion of alternative values that offer a greater congruence with the data series trend by computing conditional probabilities locally. Validation is carried out over synthetic data. Explicative power is evaluated by matching the reconstructed Bayesian structure against the known synthetic ground truth. Predictive power is assessed in its two flavours, error detection capabilities and error reconstruction capabilities and compared against the global predecessor.

The detection process is then applied to a subset of the data coming from the hardening furnace. Manufacturing of seamless steel tubes used for operating at high temperatures and pressures requires the creep resistance resulting from heat treatment. Heat treatment is a set of metalworking processes that alter the mechanical characteristics of the material by means of a sequence of heating and/or cooling to extreme temperatures. For instance, one such heat treatment, annealing, changes material properties such as strength and hardness. This manufacturing process often yields a wealth of data with over 120 different variables with very long series. This data is used to classify the steel tube as compliant or not with resistance requirements. This classification is strongly affected by the quality of the data. However, during data acquisition and storage; defective sensing, noise affecting transmission and transcription mistakes may corrupt the data. Thus, to achieve a more accurate classification, data is put through a data validation process. In this scenario, type I errors i.e., considering faulty an actual correct value, are affordable as long as the suggested alternatives are good approximations, emphasizing the critical importance of the reconstruction. Performance is then compared to the previous existing global solution [23].

Contribution is three-fold; (i) we provide a new solution with overall better capabilities for complex data topologies, (ii) we establish some rules of thumb for model parameterization both for the new approach and its predecessor, and (iii) we verify and validate the approach delimiting its incongruent data validation capabilities.

Organization of the paper is as follows. First, the computational approach is presented, and the datasets both synthetic and real are introduced. Then, to reduce the search space, an initial stage chooses statistically relevant model parameters. After fixing the model parameters, a 5-fold validation exercises explores the face validity of the approach evaluating predictive and explicative properties of the solution. The model parameters are automatically learned from the data structure and distribution to adapt to different problems and scenarios, and in principle it can accommodate any number of variables (other than memory limits) and it is not constraint to a particular data distribution favouring scalability and generalizability. Finally, concurrent validity is established over real data.

2. Preliminaries

2.1. Bayesian Networks

A Bayesian Network (BN) [10, 34] $N = (X, G, P)$ is a directed acyclic graph (DAG) $G = (V, E)$ with nodes $V = v_1, ..., v_n$ and directed links E. The nodes of G represent the set of random variables X of the domain and for each random variable $X_v \in X$ a *Conditional Probability Distribution* P of the form $P(X_v|X_{pa}(v))$ is associated, where $X_{pa}(v)$ are the set of parents of v.

The BN structure and its parameters (CPD) can be defined explicitly. However, these can be learned automatically through a set of data. Several algorithms are available for this purpose [39, 40, 9]. During the network structure learning, a statistical test between each pair of variables must be computed in order to discover relations of conditional independence, a process that is facilitated by the discretization of the variables' ranges.

A defined (learned) BN represents a knowledge base which its mayor purpose is to reason under uncertainty about observed events in its domain. This reasoning is carried out by probabilistic inference whose main task is to compute the posterior marginal probability of an unobserved variable given a set of observed (evidence) variables.

2.2. Discretization

Discretization is the process by which the values of continuous variables are converted to discretized, ordinal or nominal values. The discretization process is non-trivial and many approaches exist [27, 15]. Two classical strategies are equi-distance by which the variables' data range is split in a predetermined number of equally distant intervals, or equi-frequency in which the splitting of the intervals ensure that each interval holds the same number of samples. Recently, we proposed an interval discretization technique based on a Gaussian mixture model (GMM) [21]. This approach optimizes binning based on the data distribution. In GMM-based interval discretization, the data is assumed to be generated by a mixture of Gaussian distributions. Each fundamental Gaussian is characterized by its mean μ and its variance σ^2, and the mixture is given by Eq. 1:

$$p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \sigma_k^2) \tag{1}$$

where K is the number of Gaussians considered, $N(x|\mu_k, \sigma_k^2)$ represents a Gaussian with mean μ_k and variance σ_k^2 and π_k are the mixing coefficients, i.e. weights for the Gaussians. The algorithm has K as a single parameter. The classical Expectation-Maximization algorithm [13] is used to optimize the fitting of the distributions. The critical value discriminating any two contiguous intervals is chosen at the point in which the two involved fundamental Gaussians exhibit equal probability. The main advantage of this approach is that every interval corresponds to a specific distribution of the data. Among the available aforementioned discretization strategies, we opted by the latter based on Gaussian mixture relying on our expertise.

3. Related work

Several approaches that use Bayesian networks are presented by different communities like anomaly detection [6], sensor validation [37], outlier detection [46] and more recently, the work presented in [**?**] propose the use of a Dynamic Bayesian Network to learn a non-stationary process to detect anomalies in a network system. Some approaches are dependent on the application domain. Examples are labeled databases [12], wireless sensor networks [25], high pressure fluid-filled pipe type cables [44] or academic computer networks [38]. The main idea in most of those systems is the assumption of *normal* behavior process data used for training a probabilistic model, and inference is used to compare and measure the correctness of incoming data. Other approaches has addressed the problem of error detection by means of classifiers like the ones presented in [14, 7] who uses decision trees to detect faults or the work of [30] where fault detection is performed by a fuzzy classifier.

3.1. Detection of rogue values with related variables: a global solution

In a nutshell, the idea of validating rogue values using related variables is as follows. Statistical dependencies among variables established from the timecourse are exploited to isolate samples that flagrantly i.e. significantly, violate the expected relations. The original algorithm was proposed in [23], and we briefly describe it here. The process, schematically depicted in Figure 1, starts with the discretization of each variable range. In the original algorithm, without losing generality, the discretization was achieved using an equi-distance criterium. Then a BN is trained (structural learning) with domain entities as nodes. This BN will naturally catch conditional dependences between the variables in its arcs. Structural learning of the network can be achieved by any existing algorithm, e.g. PC [39]. Figure 2 illustrates a BN where the domain variables (nodes) and the probabilistic relation between them (edges) can be observed. Once the BN structure is defined, it permits identification of rogue values using a two step process; (i) identification of error candidates and (ii) isolation of the real errors.

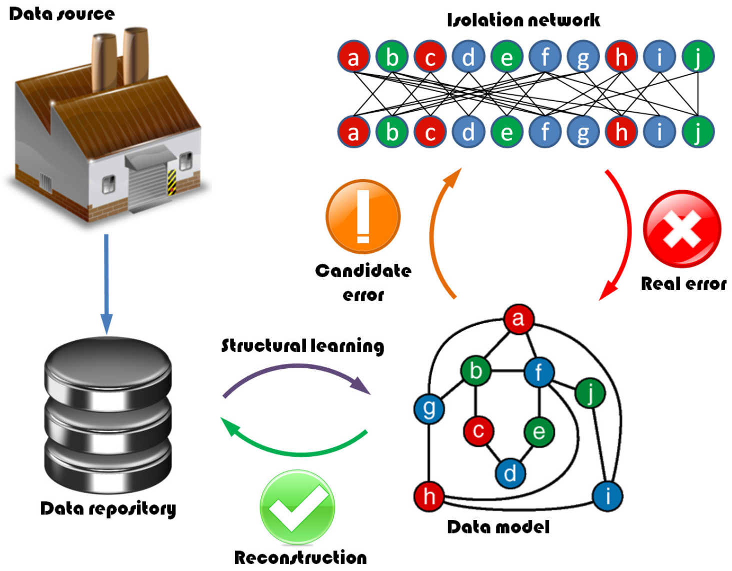


Figure 1: Schematic depiction of the data validation process for rogue value detection and reconstruction. With a given set of data a model is learned which describes the relation between variables of the domain. This model is able to detect suspicious errors (candidate errors). In a second stage, candidate errors are analysed with other model to isolate real errors. The learned model is used later to reconstruct every erroneous data.

Phase I: Identification of error candidates. During the identification of error candidates, suspicious records i.e. those failing to comply with expected conditional probabilistic relations, are labelled as *candidate errors (CE)* if the probability of the sample value in the light of other variables values falls below a given significance *candidacy threshold* p_c. The probability of the sample value is given by the posterior probability distribution of the variable which for any variable can be established by isolating the variable Markov blanket i.e., its parents, children and other parents of its children, and propagating evidence on the net. Equation 2 describes formally the aforementioned process where X_i is the variable to validate, O_i is the observed value for X_i, N is the Bayesian Network Model, $\vec{O}$ is a vector of current observed values and MB_{X_i} is the Markov Blanket of the node X_i for the given model and observation.

$$CE(X_i; N, \vec{O}, p_c) = \begin{cases} \text{TRUE} & \text{If } P(X_i = O_i | MB_{X_i}(N, \vec{O})) < p_c \\ \text{FALSE} & \text{Otherwise} \end{cases} \tag{2}$$

The network is used to detect *candidate errors* estimating the posterior probability of every node given the nodes in its Markov blanket. After a complete cycle a set of apparent errors is obtained. In the first step, some correct values may be flagged as a rogue if they have been estimated with actual rogue values in other variables. Thus, the second stage is necessary to isolate real errors from the set of candidate ones.

Phase II: Isolation of the real errors. After the candidate errors have been tagged, a new isolation BN is built. This subordinate network contains all the nodes of the original network replicated in two levels as illustrated in Figure 3. The upper layer contains the list of all nodes in the phase I network with an indication of R_ -for (R)eal (a true error)- at front of the nodes name. Similarly, the lower layer contains all nodes with names starting with A_ -for (A)pparent (candidate error)-. Let S_A be the set of variables with apparent faults and $EMB(X)$ the Extended Markov Blanket (EMB)[1] of a variable X. S_A is compared with the table of the EMB for each variable. The EMB of each node dictates the relation between nodes across levels with different possible outcomes[23]:

- $S_A = \emptyset$ there are no faults.
- If S_A is equal to the union of several EMBs and the combination is unique[2], there are multiple distinguishable real faults in all the

[1] The set of nodes in the Markov Blanket of a node plus the node itself.

[2] A *unique* combination means that $\forall X_k \neq X_i, i = 1, \ldots, n$ then $EMB(X_k) \not\subseteq EMB(X_1) \bigcup EMB(X_2) \bigcup \ldots \bigcup EMB(X_n)$ and $EMB(X_j) \not\subseteq \bigcup_{i \neq j} EMB(X_i), j = 1, \ldots, n$.

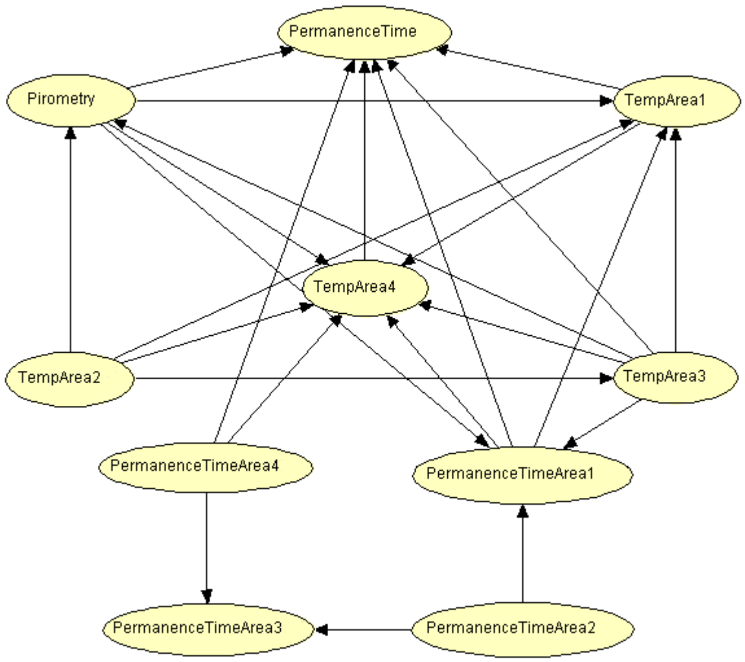


Figure 2: A BN model captures the relations between variables of the steel industry domain based on probabilistic theory. The connecting arrows indicate the direction of the influence i.e., a change in a parent node alter the state of its child nodes. The process of manufacturing steel tubes requires a heating process that involves passing the tubes through the annealing furnace where the tubes are heated at 900°C. Following, the tubes are water cooled reducing their temperatures to 50-60°C. Finally, the tubes go through the hardening furnace where the tubes are heated again. Furnaces are divided in zones; the annealing furnace is divided in 5 zones, whereas the hardening furnace is divided in 4 zones. The shown variables correspond to data from the hardening furnace. More specifically; $PermanenceTime$ is the total time that the tube is in the furnace, $PermanenceTimeArea1$, 2, 3 and 4 is the time that the tube remains in each zone, $TempArea1$, 2, 3 and 4 are the temperatures in each zone of the furnace, and $Pirometry$ corresponds to the temperature reached by the tube itself.

variables whose EMB are in S_A.

- Otherwise, there are multiple faults but they can not be distinguished. Any variable X whose EMBs is a subset of S_A, i.e. $EMB(X) \subseteq S_A$, could have a real fault.

The isolation network follows this procedure for all nodes.

For example, the $PermanenceTimeArea3$ node possess an EMB composed by the set:

$$EMB(PermanenceTimeArea3) = \{PermanenceTimeArea4, PermanenceTimeArea3, PermanenceTimeArea2\} \tag{3}$$

In Figure 3, the node $R\_PermanenceTimeArea3$ connects to the nodes $A\_PermanenceTimeArea4$, $A\_PermanenceTimeArea3$ and $A\_PermanenceTimeArea2$ as shown in the second node from the left in Figure 3.

The parameters of this network are set according to the *noisy-OR* [23] which is able to relate a manifestation with a set of possible causes. The following formula (Eq. 4) is used to calculate the conditional probability tables for the isolation network:

$$P(e|d) = \begin{cases} \Pi_{i \in T_d} & \text{If } \neg e \\ 1 - \Pi_{i \in T_d} & \text{If } e \end{cases} \tag{4}$$

where e is an observed manifestation (i.e., a candidate error) and d a set of possible causes (i.e., real errors). Basically, this equation express the probability that an apparent error be the cause of a real error present in other variable (or a set of variables).

To determine a real error through the isolation network, each candidate error detected in the previous step, is instantiated as true, and the rest of nodes are instantiated as false. Evidence is propagated throughout the network and a posterior probability distribution for every node in the upper level is established. In other words, when all of the nodes belonging to a EMB of a node contain true as candidate error from previous step, the propagation will produce a high probability of real error and low in the others. Theory [23] dictates that whenever there is a real failure, this will generate one or more apparent faults. However, it may be the case that none of the apparent faults are detected e.g. their effects may conceal the effects of the other faults, with the worst case scenario being that the algorithm fails to detect all the apparent faults.

Nevertheless, in these cases, if one variable in the EMB of a faulty node is not detected as apparent faulty in the first stage, including the same variable itself, the second propagation will produce a high probability real fault as proven in our earlier publication [23]. And indeed, experimental results suggests that not detecting the real fault is unlikely. Real errors are identified by thresholding the probability of a node over a significant *isolation threshold* p_i. In phase I, the aim is detecting those data records containing at least one error. In phase II, the task is deciding which are the offending variables.

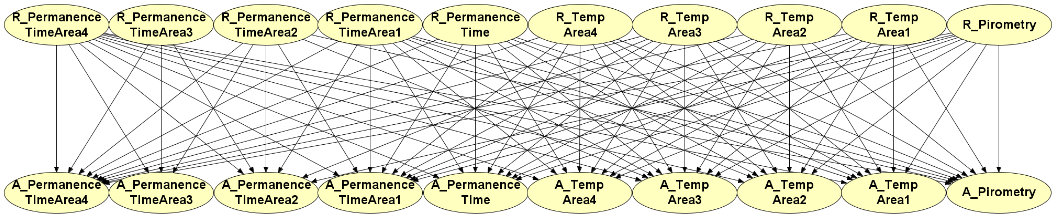


Figure 3: Isolation network for BN data model in Figure 2.

Finally, suggestion of plausible values is performed through inference over the model. The original BN model used in phase I is used for the data reconstruction. Feeding valid values of the related variables into the model, reconstruction is done by propagating evidence through the BN to the affected variable. However, due to the discretization occurring prior to the structural learning process of the BN, the model can only afford an interval estimation. The simplest solution to map this interval to a single value is considering the middle point of the interval. However, as the intervals grow larger, the middle point can be far from a good estimation. The new approach presented in the next section exploits the local distribution around a neighbourhood of the sample under scrutiny to render a finer discretization.

4. Proposed Method

4.1. Detection of rogue values with related variables: a local multiscale solution

We now present the proposed approach for detecting rogue values. This new approach taps the local distribution around a neighbourhood of the sample being validated to obtain a finer discretization and thus more accurate reconstruction of errors without increasing the model complexity i.e., larger conditional probabilities density tables. In the afore described global approach, following interval discretization, the network structure and the conditional probabilities tables are learned from the full set of data available for training. This new approach extends the global model by building local submodels that whilst maintaining the topology of the global network i.e., the structural statistical relations are respected, the conditional probabilities are computed from the distribution of the temporally local neighbourhood of the signals. The structure of the submodels is kept invariant ensuring the probabilistic dependencies are preserved. Figure 4 illustrates the concept.

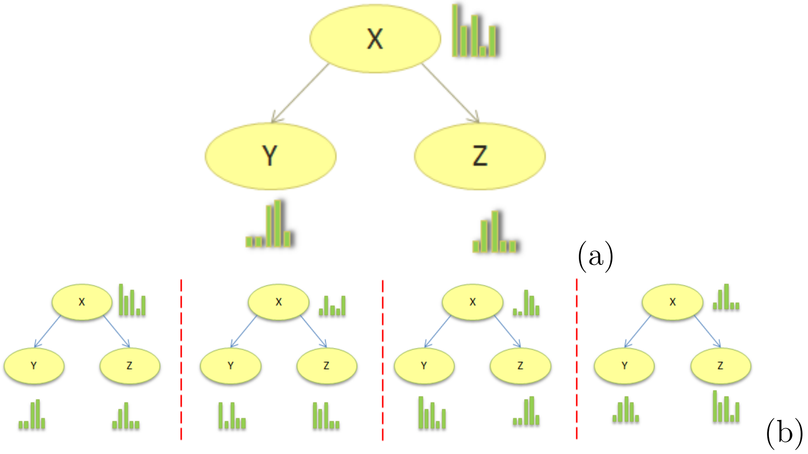


Figure 4: (a) Upper (global) scale model which governs the model topology. The bars next to each node represents the associated data distribution. This model is learned from the full dataset. (b) Local submodels for each of the temporal segments. Although the topology is inherited from the global model, the probabilities are recalculated using only the temporal segment subseries.

In order to construct the bottom level submodel, the main idea consists of splitting the data series in temporal intervals. Figure 5 illustrates the difference in the data distribution for a given dataset between global (full series) and local (subseries) approaches. Ideally, the time intervals are defined such that the subseries are stationary within themselves, or closer to it than the original full series, e.g., weakly stationary. This should result in BN models particularly fitted to the signal different behaviours.

Multiresolution differs from sliding window. Under a sliding window approach only local information is given to the algorithm, whereas under mutiresolution both global and local information reaches the algorithm. The implications are clear; the multiresolution keep into consideration the global information which a simple sliding window would miss.

4.1.1. Time series splitting

In applying a local solution, there is a need for partitioning the time series into local chunks. Besides the obvious manual partitioning, several options are suitable for splitting the series:

Equi-spaced. The user indicates the number of intervals κ, i.e., the number of submodels, and the series is split in equal sized intervals. This is the method chosen in Figure 5.

Overlapped. Similar to the previous one, but the κ intervals are allowed to overlap in their boundaries by a certain amount τ, so that changes are more progressive.

Sliding window. In the extreme case, a neighbourhood is built over each sample using a sliding window of size ν and a model is built for each sample. The sliding window may be constructed using any classical kernel; rectangular, triangular, Gaussian, Welch, Hamming, Hann, etc.

These options represent a compromise between accuracy and computational complexity. The preceding options are generic and do not address the question of whether the original signal is non-stationary and the individual resulting chunks are stationary. Segmenting a time series into its locally stationary parts is a hard computational problem that has already been addressed [16, 31] for which practical exact solutions are yet to be developed considering the order of a would-be exact solution $O(N^N)$ [16]. Although we hypothesize that a segmentation considering stationarity of the time series will improve the results of the local approach proposed, for this work we stick to more naive partitioning; i.e., equi-spaced. For practical purposes, throughout this work the time series were split arbitrarily into 8 intervals. The window size -not the number of them-, e.g. the length in number of samples of the interval, is actually relevant as this relates directly to the observations seen by the BN during structure learning. If splitting breaks a stationary period into two subperiods there is no harm unless the new chunks contain too few samples to permit appropriate parameterization of the BN, but since structure

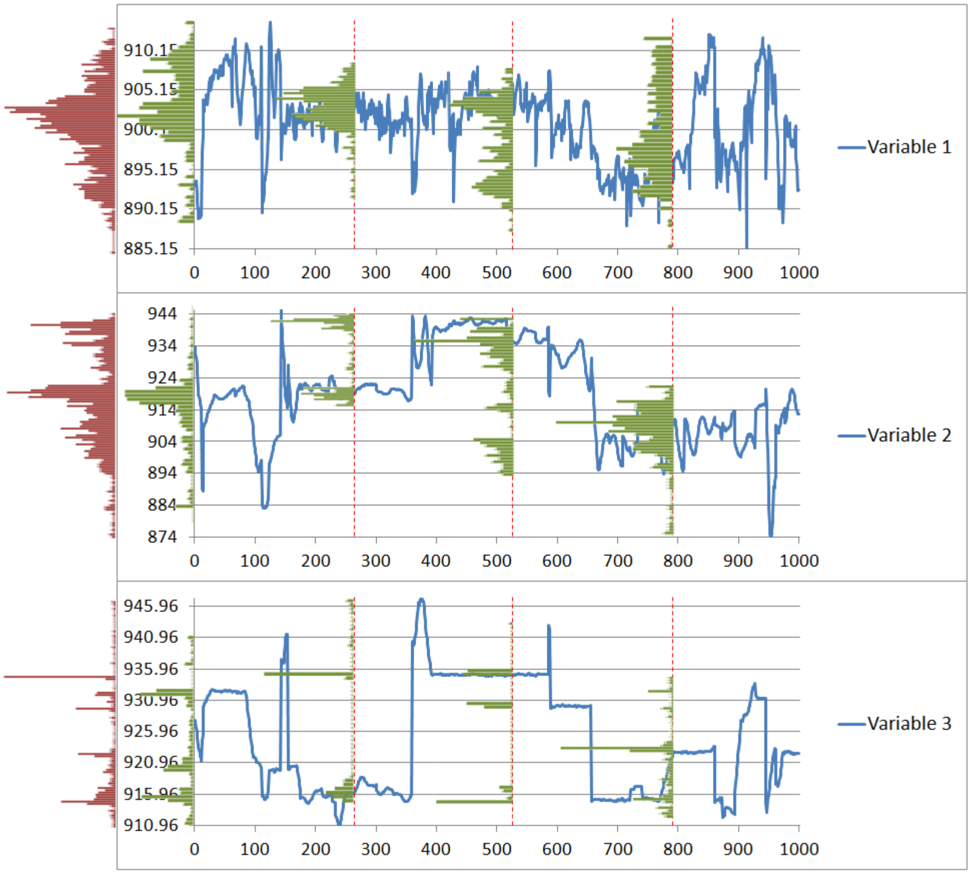


Figure 5: (Best seen in color) Global (red histogram) versus Local multiscale (green histograms) BN model construction. Under the global approach the BN learns from the distribution of the data ranges of the full series. Under the local multiscale approach, BN submodels are learn from the data distributions of the local temporal segment. The example shown uses equi-spaced splitting of the time series. In both cases, the abscissa axis represents the timecourse of the variables in terms of number of samples. The histogram of the series is represented in the ordenada axis. The horizontal dashed lines indicates the interval partition obtained for each variable in this case obtained using equi-distance discretization. It can be appreciated how the local discretization flexibly adjusts the size of the intervals according to the local properties of the subseries.

is learnt globally, this has an attenuated effect. On the other end, as window size grows the local method approaches the global method.

4.1.2. Upper scale model construction

The structure of the upper scale BN model is built considering the full dataset (global scale) for training. The global BN determines the probabilistic relations of the domain regardless of the temporal dynamics. The network topology of the upper level root model is then replicated for each segment and thus inherited by the submodels guaranteeing that the conditional dependencies are kept consistent through the local submodels. However, each local BN submodel probabilities are recalculated to adapt to the local segment, where each one is discretized applying the same discretization method as in the global model.

4.1.3. Submodel selection

The selection criteria chooses the network submodel with higher probability of producing the observation e.g., the t-th record $< x_t^1, x_t^2, \dots x_t^n >$ with n variables.

In this sense, let O be an observation or data record including the samples across all domain variables. O_t is the observation at time t. The probability that observation O_t is generated by a submodel m is given by $P(O_t|m)$. It is possible to approximate this probability from the individual probabilities that value x_t^i observed for variable X^i belongs to one of the intervals for network node X_i. Assuming statistical independence of the samples (not the time series):

$$P(O_t|m) = \prod_{X^i|i=1\dots n} p(a_k \leq x_t^i \leq b_k) \tag{5}$$

where a_k and b_k are the lower and upper boundaries of the $k-th$ interval for variable X^i where the value x_t^i befalls.

Upon deciding on the submodel, data validation check proceeds as per the original algorithm with the identification of error candidates first, and isolation of real errors afterwards. Since all submodels share the same topology, it is possible to generate a single isolation network during the last step. Instantiation of the nodes in the isolation network is carried out according to the chosen submodel for evidence propagation and real error detection.

4.1.4. Data reconstruction

Analogously to the global solution, suggestion of plausible values is performed through inference over the submodel that best represents the observed data. The selection of the submodel is done using the same criterium of obtaining the probability of the observation given the model, $P(O_t|m)$. However, upon considering that the variable value to be estimated may be affected by an error, the probability of this is neglected in the computation, and only the value of the related variables are considered. In this sense, the submodel exhibiting higher probability of generating the observation defined by the values in the Markov Blanket of the estimated variable are used to compute the most likely value. As this corresponds to an interval, the numeric value proposed is according to the mid point of the interval.

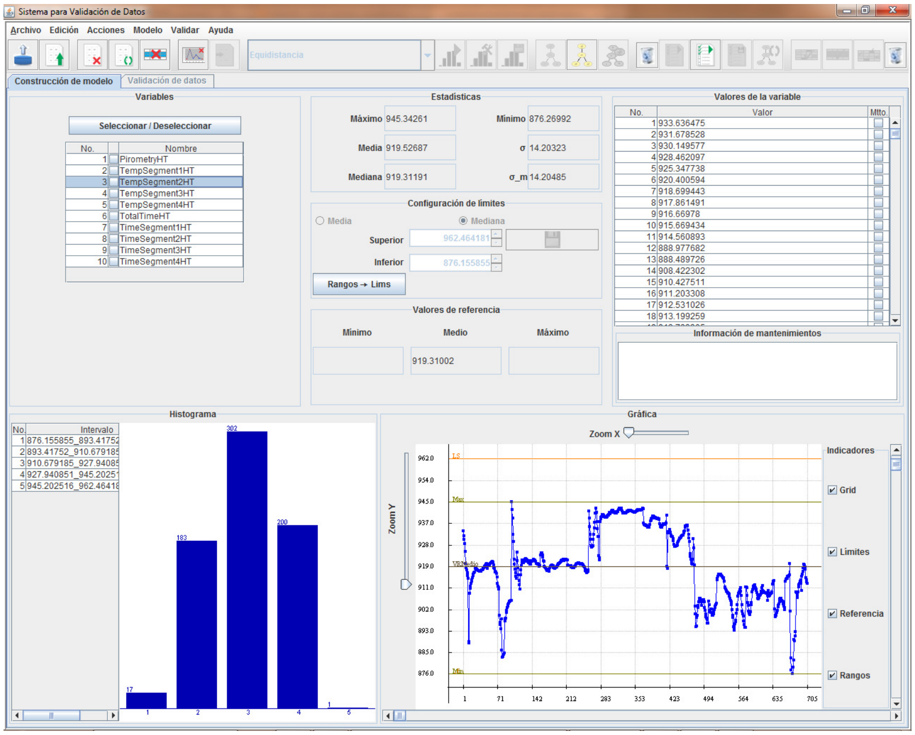
4.2. Simulation environment

Validator is a Java based platform that our group has built for generic data validation [20] (see Figure 6). This environment is not attached to any specific domain. It is capable of outlier detection, as well as detection of sudden changes, and we have incorporate the solutions presented here for the detection of rogue values using related variables (both global and local multiscale). All experiments that follow have been carried out in Validator which is available online at: http://haro.inaoep.mx/∼validator.

5. Validation on synthetic data

In order to verify the behaviour of our approach against a known ground truth, synthetic data topologies were generated using well known time series models. A synthetic topology is a predefined network structure for which the relation among node entities is set in terms of generative time series models. These topologies yield a set of time series with direct and indirect relations among variables, and with single parent and multiple parent cases. In total, three synthetic data topologies were generated (Figure 7) with 2000 samples for each variable. Each topology defines a particular relation of interest between its nodes. The details are explained in appendix A.

Table 1 summarises the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests for the synthetic topologies. Note that not all series in the synthetic topologies are (level) non-stationary, but the topologies go with increasing difficulty.



No.		Nombre
1	□	PirometryHT
2	□	TempSegment1HT
3	□	TempSegment2HT
4	□	TempSegment3HT
5	□	TempSegment4HT
6	□	TotalTimeHT
7	□	TimeSegment1HT
8	□	TimeSegment2HT
9	□	TimeSegment3HT
10	□	TimeSegment4HT

Máximo 945.34261
Mínimo 876.26992
Media 919.52687
σ 14.20323
Mediana 919.31191
σ_m 14.20485

○ Media ● Mediana
Superior 962.464181
Inferior 876.155855

Mínimo	Medio	Máximo
	919.31002	

No.	Valor	Mtto.
1	933.636475	□
2	931.678528	□
3	930.149577	□
4	928.462097	□
5	925.347738	□
6	920.400594	□
7	918.699443	□
8	917.861491	□
9	916.66978	□
10	915.669434	□
11	914.560893	□
12	888.977682	□
13	888.489726	□
14	908.422302	□
15	910.427511	□
16	911.203308	□
17	912.531026	□
18	913.199259	□

No.	Intervalo
1	876.155855_893.41752
2	893.41752_910.679185
3	910.679185_927.94085
4	927.940851_945.20251
5	945.202516_962.46418

☑ Grid
☑ Límites
☑ Referencia
☑ Rangos

Figure 6: Screenshot of the Graphical User Interface (GUI) of the data validation tool Validator. The GUI shows the model construction tab with various components like: variable list, statistics information, histogram and time-series graphs, discretization algorithms and data manipulation tools.

5.1. Statistical analysis

Statistical analysis has been carried out in R [1]. The KPSS test [28] is used for testing a null hypothesis that an observable time series is stationary around a deterministic trend. Effect sizes and corresponding z-scores have been used to establish the effect of thresholds for algorithm stages I (identification of candidate errors) and II (isolation of real errors). A two-way ANOVA model has been built for evaluating the parameterization of models, and in particular for establishing the discretization strategy and the number of intervals for the discretization. Although these parameters are nested i.e., the number of intervals obviously depends on the discretization strategy, for statistical modelling we unfolded this nesting by considering 3 levels of intervals; 5, 10 and automatic (which is manually set to 15 in the case

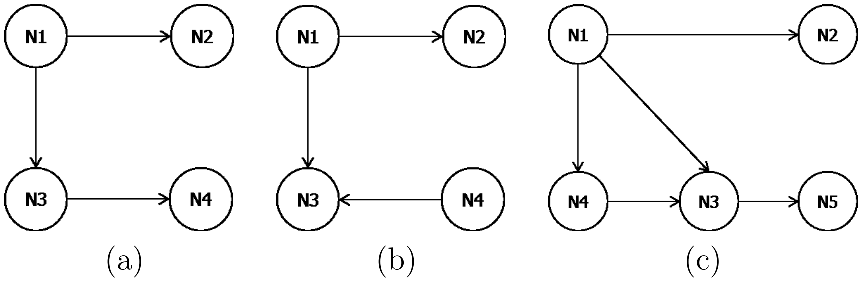


Figure 7: The three synthetic topologies simulated for model verification and face validity. Each node in the synthetic topology is a combination of its own seed plus maybe some influence from the other nodes (see Appendix A). Direct and indirect relations, as well as single parent or multiple parent relations are considered.

of equifrequency) and considered a plain two-way ANOVA. The unfolding of a nested factor is equivalent to considering a plain multifactorial design (number of intervals x discretization strategy). No post-hoc pairwise analysis followed, but instead we rely on boxplots representation for decision.

5.2. Parameterization and verification of the model

As already mentioned, 2000 samples were simulated in the synthetic topologies and local approach always used 8 equally spaced segments for partitioning the series. In this first stage, a single fold test is run over the synthetic data in order to find a combination of parameters that shows a promising behaviour across both the global and local approached. This shall avoid an exhaustive parameter space search during the second stage in which face validity is established. This test also helps to establish the explicative power of the model. Synthetic data was split into 70% training set and 30% test set with the samples chosen at random. Note that a static Bayesian network as the ones underlying the solutions does not take time into account and thus the temporal distribution of the samples does not affect the construction of the models. For this single fold, global and local models were trained and tested using specific combination of parameters including algorithm stages' thresholds p_c and p_i, discretization strategy and number of intervals, in particular, all possible combinations of the parameter values in Table 2. Evaluation was based on sensitivity and specificity analysis (Eq 6 and 7)and the derivated area under the curve (AUC) over the receiver operator curve (ROC) space.

For Synthetic topology 1		
N1:	KPSS Level = 0.126	$p > 0.1$
N2:	KPSS Level = 3.7758	$p < 0.01$ **
N3:	KPSS Level = 0.0684	$p > 0.1$
N4:	KPSS Level = 0.071	$p > 0.1$
For Synthetic topology 2		
N1:	KPSS Level = 15.632	$p < 0.01$ **
N2:	KPSS Level = 17.2546	$p < 0.01$ **
N3:	KPSS Level = 0.126	$p > 0.1$
N4:	KPSS Level = 6.696	$p < 0.01$ **
For Synthetic topology 3		
N1:	KPSS Level = 16.0362	$p < 0.01$ **
N2:	KPSS Level = 16.6984	$p < 0.01$ **
N3:	KPSS Level = 12.7894	$p < 0.01$ **
N4:	KPSS Level = 15.3948	$p < 0.01$ **
N5:	KPSS Level = 15.3923	$p < 0.01$ **

Table 1: Results of the variable-wise KPSS test of signal stationarity in the synthetic topologies. ** Indicates a highly significant value ($p < 0.01$). * Indicates a significant value ($p < 0.05$). Truncation lag parameter = 10 in all cases.

$$Sensitivity = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{6}$$

$$Specificity = \frac{TrueNegatives}{TrueNegatives + FalsePositives} \tag{7}$$

5.2.1. The effect of initialization

Maximum sensitivity for the candidacy threshold p_c was achieved when parameter was valued 0.0001 and 0.001 was below 0.1 and corresponding AUC was below 0.55 which is close to random. Thus these values can be considered as too stringent, and can be discarded from further consideration. With respect to values 0.01 and 0.05, AUC increased slightly (maximum reaching 0.57 and 0.63 respectively) with no significant difference between these two.

Effect sizes $(\mu_{local} - \mu_{global})/\sigma_{global}$ over the AUC were computed between every pair of values given to the isolation threshold p_i across topologies and

Parameter	Tested values
Candidacy threshold p_c	0.0001, 0.001, 0.01 and 0.05
Isolation threshold $p_i \in [0.5, 1]$	0.51, 0.6, 0.7 and 0.8
Discretization strategy	Equi-distance, equi-frequency and GMM
Number of intervals (nested to discretization strategy)	Equi-distance: 5, 10 and Automatic Equi-frequency: 5, 10 and 15 GMM: 5, 10 and automatic

Table 2: Discretization of the parameter search space for model verification. Tested parameterizations are all the possible combinations from these values. In each case, both a global and a local multiscale model were built.

models. The maximum size effect corresponded to the comparison between values 0.51 and 0.8 as expected, and was found to be 0.18 which corresponds to a non-significant p-value $p = 0.42$. This suggests that the choice of the isolation threshold p_i is virtually irrelevant for the final detection of errors, which makes the approach robust to this parameter. Tables 3 and 4 summarises the results of the initialization stage

Parameter	Values	Sensitivity	Specificity	AUC
p_c	0.0001	0.006 ± 0.008	0.99 ± 0.008	0.50 ± 0.004
	0.001	0.015 ± 0.018	0.99 ± 0.001	0.50 ± 0.008
	0.01	0.034 ± 0.041	0.99 ± 0.007	0.51 ± 0.017
	0.05	0.059 ± 0.077	0.98 ± 0.022	0.52 ± 0.029

Table 3: Summarised results for initialization of parameter p_c

5.2.2. The effect of discretization

Discretization involves two important decisions. First, the binning approach as discussed in Sect. 2.2. Then, nested to this decision, is the number of intervals used for this binning. Figure 8 shows the boxplots for the AUC over the two factors; binning approach and number of intervals. The two way ANOVA model results summarized in Table 5 suggests that both the effect of the binning strategy and the number of intervals were found to be highly significant ($p < 0.001$). Based on the previous boxplots, the equi-distance

Parameter	Values	Effect Sizes
p_i	0.51 - 0.6	0.02
	0.51 - 0.7	0.15
	0.51 - 0.8	0.18
	0.6 - 0.7	0.13
	0.6 - 0.8	0.16
	0.7 - 0.8	0.03

Table 4: Summarised results for the initialization of parameter p_i

binning with automatic partitioning was regarded as the most suitable parameterization.

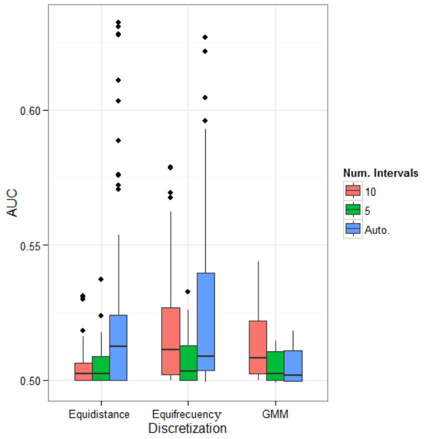


Figure 8: Parameterization of the discretization strategy and subsequent choice on the number of intervals. Three discretization algorithms with different number of intervals were evaluated. The boxplots points to Equidistance discretization with automatic number of intervals to be the binning strategy with the highest AUC.

5.2.3. Scalability with increasing data complexity

We also questioned how do the models deal with the increasing complexity of the topologies. Effect sizes comparing the difference in performance between the two models over the AUC were computed for each topology. In Figure 9 , it can be appreciated the monotonic trend suggesting how the local multiscale model better deals with increasing complexity of the data

Factor	**Df**	**Sum. Sq.**	**Mean Sq.**	**F value**	$Pr(> F)$
Binning strategy	2	0.02371	0.011853	28.48	2.49e-12
Num. intervals	2	0.03468	0.017339	41.66	<2e-16
Binning strategy: Num. intervals	4	0.03065	0.007664	18.41	5.83e-14
Residuals	423	0.17605	0.000416		

Table 5: Two way Type I ANOVA results for assessing the effect of discretization. Both the effect of the binning strategy and the number of intervals were found to be highly significant ($p < 0.001$). Df: degrees of freedom; Sum. Sq.: Sum of squares; Mean Sq.: Mean square.

topology, as the effect size i.e., difference between the two models, increases with increasing complexity of data.

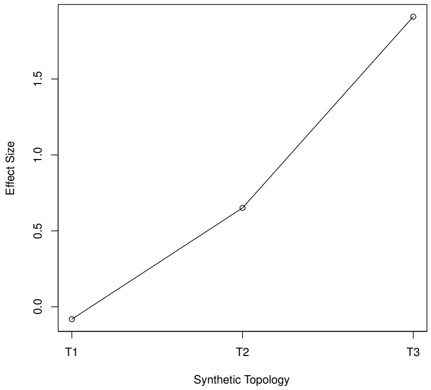


Figure 9: Effect size measuring the difference in response by the two models with regard to the increasing complexity of the topology. Note that the dependent variable in the plot is the effect size already implicitly expressing the relation among the two models; the global and the local.

Finally, the defined configuration of parameters was used to measure the performance of the global and local methods over the three synthetic topologies. Table 6 shows the sensitivity, specificity and area under the curve (AUC) of the ROC curve. A t-test was conducted to compare the results of table 6 between the AUC of the global and local models. The t-test suggested that there was not significant difference for topology 1 ($t(269.6) = 0.775, p = 0.78046$) but there were a significant difference for topology 2 ($t(173.9) = -2.469, p = 0.0073$) and 3 ($t(169.9) = -6.806, p = 0$). So, as can be seen, the AUC for the local model for topologies 2 and 3, surpass the error detection capabilities of the global model. These results are in agreement with the findings stated in figure 9 that local model deals better with complex data topologies.

Topology	Method	Sensitivity	Specificity	AUC	p-value
T1	Global	0.048 ±0.058	0.989 ±0.012	0.519 ± 0.025	0.780
T1	Local	0.055 ±0.064	0.978 ±0.027	0.517 ± 0.019	
T2	Global	0.019 ±0.030	0.996 ±0.009	0.508 ± 0.010	0.007 *
T2	Local	0.038 ±0.082	0.991 ±0.019	0.515 ± 0.032	
T3	Global	0.037 ±0.021	0.995 ±0.004	0.516 ± 0.009	0.000 *
T3	Local	0.084 ±0.076	0.986 ±0.014	0.535 ± 0.031	

Table 6: Error detection performance of the global and local methods evaluated on each synthetic data topology. * Indicates a significant value ($p < 0.05$)

5.3. Computational complexity

The computational complexity of the proposed method needs to be split in two parts: the models learning stage and the validation stage. In the first one, the complexity is composed by the time used in the structure and parameters learning of the BN. As the structure is learned with the PC algorithm this takes (in the worst case) a time in (p^q) [26] where p is the number of variables, q the maximum number of vertex adjacent to any vertex in the graph. The parameters are learn, with the EM algorithm, with a time in $(l + kl^2)$ [8], where l is the number of samples and k the number of intervals of each variable. Then, to learn the global model the complexity is $O(p^q) + O(l + kl^2)$. As the submodels share the same structure of the global model, the time to define the structure of the submodels is constant i.e. $O(1)$, but the parameters learning should be repeated N times (one for each segment of the time series) over a set of size l/N samples, so the time to learn the submodels is $O(1) + N \cdot O((l/N) + k(l/N)^2)$ In the case of the validation stage the computational time is basically the time employed by the probability propagation algorithm, which is known to be NP-Hard [36]. The local approach adds to this the time related to model selection, which is linear over the number of submodels $O(N)$. Then, the complexity for the validation process with the local submodels is increased linearly by the model selection.

Variable	**p-value**
Pyrometry	$p < 0.01$ **
TemperatureArea1	$p < 0.01$ **
TemperatureArea2	$p < 0.01$ **
TemperatureArea3	$p < 0.01$ **
TemperatureArea4	$p < 0.01$ **
TotalTime	$p < 0.01$ **
PermanenceTimeArea1	$p = 0.04061$ *
PermanenceTimeArea2	$p = 0.05843$
PermanenceTimeArea3	$p = 0.04314$ *
PermanenceTimeArea4	$p = 0.02301$ *

Table 7: Results of the variable-wise KPSS test of signal stationarity. ** Indicates a highly significant value ($p < 0.01$). * Indicates a significant value ($p < 0.05$).

6. Application domain: Steel industry

We apply the new data validation technique to a dataset coming from the steel industry in the manufacturing of seamless steel tubes. The dataset corresponds to the subset of 10 variables involved of the hardening furnace stage of the tube manufacturing process. In the hardening furnace, under differential hardening, different areas of the furnace provides separate heat treatments. The hardening process at hand separates four areas of furnace and thus it contains 4 temperatures from each area, 4 exposure times i.e., one per area, 1 total time of the tube in the furnace and 1 pyrometric measurement.

As we hypothesized that the suitable scenario for application of the local multiscale solution is in non-stationary signals, we tested all variables for stationarity using the KPSS test. The results are summarized in Table 7. All variables except PermanenceTimeArea2 exhibited non stationary behaviour. Upon splitting the signals into intervals some shifts in statistical properties responsible for this become apparent (see Figure 10).

6.1. Performance in the Error Detection

The ground truth of the errors present in the dataset is unknown. Although the data will naturally be affected by noise, we do not know neither the distribution of this noise nor whether any of the sensors responsible for the data corresponding to each variable did failed or not. Thus, in order to

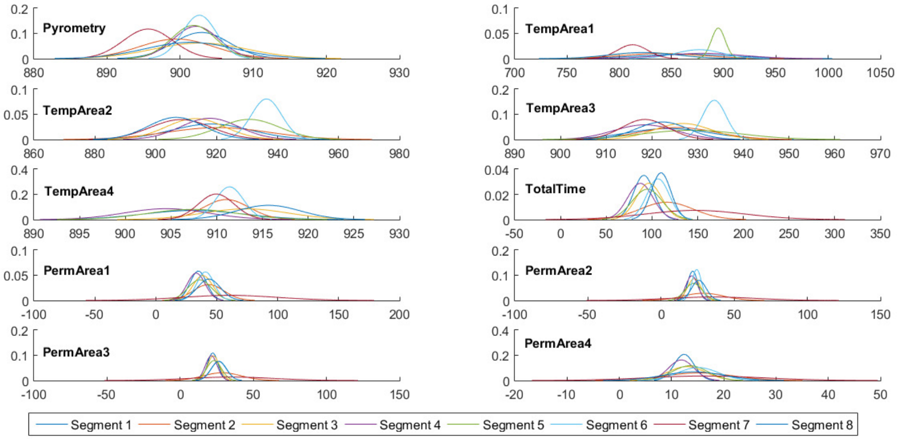


Figure 10: Distribution of the different segments across the dataset variables. Each color corresponds one of the 8 segments. The varying statistical properties of the segments in most variables can be appreciated.

explore the error detection capabilities of the multiscale approach, for this exercise we assume that our dataset is error-free and manually introduce additional errors in the dataset that will act as our ground truth. Note that this mean that we will be truly dealing with a higher noise ratio than actually reported, but we have already analyzed the approach over synthetic noise-free data. In all cases, 70% of the original dataset was used for training from which the structure of the BN is learned and 30% for testing. We test the solution in three different scenarios:

- **White Noise Contamination**. Addition of white noise to the test set is achieved by adding/subtracting a random value minor than 5% of the signal standard deviation to the sample value. Added noise was sampled from a Gaussian distribution. A total of 5% of the test data was altered in this way. After noisy signals were constructed in this way, both the global and the local multiscale solutions were applied to this noisy test set for detection of errors.

- **Shifting to extreme values** (5% of test data affected). To simulate errors, the values of those samples affected by errors was substituted by the most extreme signal value opposite to the sample value. Of the discrete values of the variable, in order to alter the value of the sample, the value is shifted to the furthest extreme of its interval; whether the upper or lower depending on the distance to the interval boundary. Note that a substitution for a value beyond this point will be considered an outlier, not a rogue value. In this scenario, a total of 5% of the test data was altered in this way, and the data affected was picked randomly.

- **Shifting to extreme values** (100% of the test records affected, i.e., one variable of every record was modified, that is 10% of test data affected). Test data was altered in the same manner than before, but ensuring that each record contain one and only one error. That is 100% of the records and 10% of the test set are affected.

In all cases the results are compared to the error detection rates achieved by the global solution and the structure of the BN is learned using the PC algorithm [39].

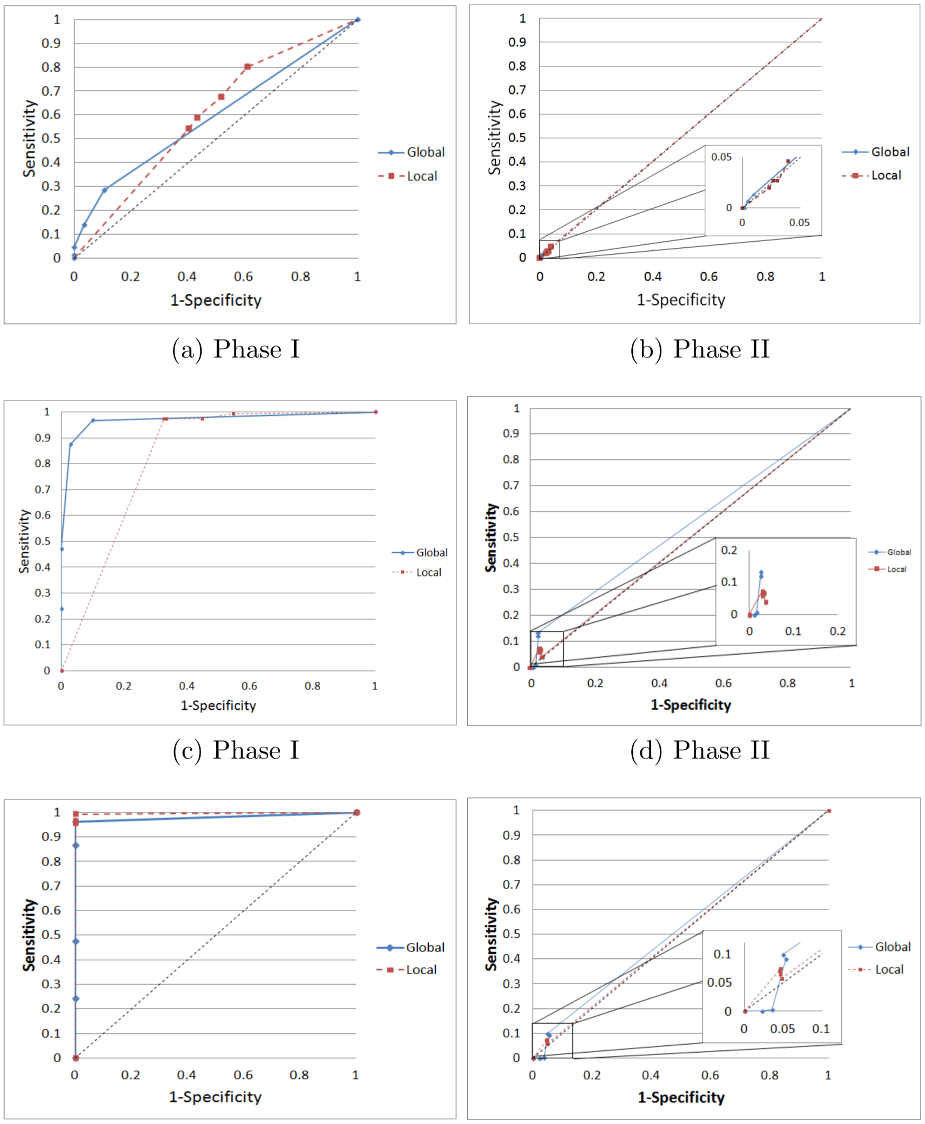
Results are presented using the receiver operating characteristic (ROC) analysis. The area under the curve (AUC) summarises the error detection rate. The AUC is a very standard way of summarizing the ROC curve; basically, the available points of the curve given by the pairs of <sensitivity, specificity>are joint and the integral (by trapezoidal approximation) is calculated. In each case, the results in both stages of the error detection process are reported.

6.1.1. White noise contamination (5% error rate)

Figure 11 summarises the error detection rate for both approaches in the presence of white noise in both error detection phases. The local multiscale approach conduct a more aggressive error detection. It captures more true positives at the expense of increasing Type I errors. Which as discussed earlier are affordable if suggested alternatives are good approximations, which we will show it is the case.

6.1.2. Shifting to extreme values (5% error rate)

The error detection rate for both approaches in the presence of this kind of simulated errors in both phases are summarised in Figure 11. In this case the global approach reached a detection of the 95% of true positives for phase I with a high p-value threshold (0.05). On the other hand, the local approach reached the same true positives rate with the lowest p-value threshold but increasing false positives.



(a) Phase I

(b) Phase II

(c) Phase I

(d) Phase II

(e) Phase I

(f) Phase II

Figure 11: Detection of candidate errors for the steel industry domain. (a) ROC curve for different configurations of the p-value (0.0001; 0.001;0.01;0.05) for Phase I (b) ROC curve for different configurations of the p-value (0.0001; 0.001;0.01;0.05) for Phase II (the zoomed window shows the details of the curve). The top row (a) and (b)- correspond to error detection under white noise contamination. Middle row (c) and (d)- correspond to error introduction of extreme values (5%). Bottom row (e) and (f)- correspond to error introduction of extreme values (10%).

	Phase I		Phase II	
Scenario	**Global**	**Local**	**Global**	**Local**
White Noise Contamination	0.5924	0.5991	0.5017	0.5031
Shifting to extreme values (5%)	0.9702	0.8262	0.5527	0.5026
Shifting to extreme values (10%)	0.9812	0.9966	0.523	0.5057

Table 8: Area under the curve (AUC) of the ROC analysis for the Global and Local Multiscale approached for the three studied scenarios.

6.1.3. Shifting to extreme values (10% error rate)

The error detection rate for both approaches in the presence of this kind of simulated errors in both phases are summarised in Figure 11. In this test, the global approach increases the true positives detection when p-value increases. The maximum rate reached by the global model is 87%. The local approach reports a detection greater than 95% for the lowest p-value threshold.

Finally, Table 8 summarise the AUC for both approaches (Global and Local multiscale) across both phases of error detection and for all scenarios. As shown in table, both approaches report a subtle difference for the test set affected with white noise. For the second scenario the global approach reported a higher AUC. Finally, the last scenario shows a AUC higher for the local approach than the obtained for the global. Values in Figure 11 and Table 8 for stage II have to be interpreted carefully because Phase I has already certainly made a good job in telling candidate errors, which are the only that reach the second stage. This means it is not an overall random output, but only that none of the approaches can easily decide which of the variables is responsible for the error in the case of white noise, and that the only thing that they can tell with some certainty is that there is an error. In other words, white noise is particularly challenging for this approach to decide the offending signal but no so much to decide whether there is an error. Moreover, in general dealing with determining the real faulty value, is a more difficult task that the one for stage I since among the set of suspicious values there could be more than one real error, obfuscating the final decision and hence affecting the ability of both approaches to detect incongruent values. The bottom line is that both method exhibit similar tolerance to noise in the data. Table 9 further shows the results over the real data.

p_c value	Phase	Method	TP	TN	FP	FN	Sensitivity	Specificity
0.01	I	Global	21	137	5	129	0.140	0.964
		Local	107	66	76	43	0.713	0.464
	II	Global	1	2757	13	149	0.006	0.995
		Local	4	2683	87	146	0.026	0.968
0.05	I	Global	43	127	15	107	0.286	0.894
		Local	121	55	87	29	0.806	0.387
	II	Global	2	2743	27	148	0.013	0.990
		Local	7	2660	110	143	0.046	0.960

Table 9: Error detection performance of the global and local methods on the application domain dataset for phases I and II of the algorithm.

6.2. Data reconstruction

After error detection, the data validation process continues to yield an alternative value which can substitute the error value. In order to test the capabilities of the local multiscale approach for data reconstruction we carried out a very simple test. We consider the whole series of one variable at a time to be erroneous, and we reconstruct that variable series only from the information available from the other variables. We repeat the process for each of the variables. Again, we compare the results against the outcome from the global approach.

Figure 12 illustrates an exemplary reconstruction for one of the variables, that is *TemperatureArea4*. The local multiscale solution clearly achieves a more faithful reconstruction than the global approach. Although the global approach reconstruction exhibits the right trend, it lacks detail. In contrast, the local multiscale solution not only exhibits the right trend but also achieves finer detail. Table 10 summarizes the mean absolute errors during reconstruction across all variables. The reconstruction stage does not dependent on the parameters of detection stages (p_c or p_i). For reconstruction, the Global Bayesian network or the selected Local Bayesian network are used to propagate the evidence through the BN to the affected variable to estimate the most probable value.

7. Discussion

Both approaches present good detection rates for phase I, identification of error candidates, i.e., detection of suspicious records. However, this detection

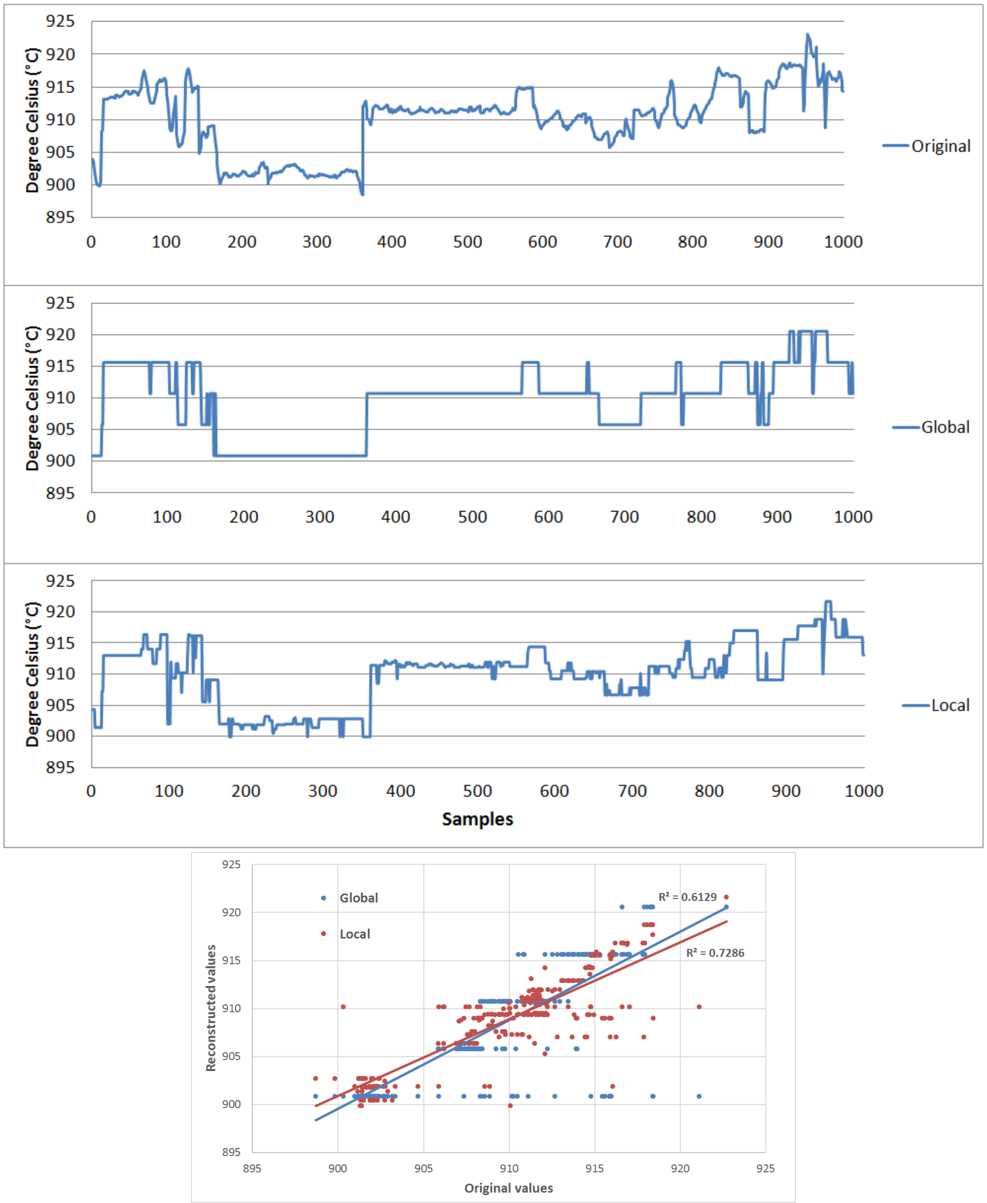


Figure 12: Data reconstruction capabilities. The original variable *TemperatureArea4HT* (top) was completely reconstructred from statistically related information in the other variables only. The reconstructions achieved by the global approach (2nd row) and the local multiscale approach (3rd row) are shown. Abscissa represent samples of the time series and ordinate represent signal value. Bottom: The greater detail achieved by the local approach (red) over the global approach is demonstrated by the higher r^2 of the regressive linear model.

Variable	**Global**	**Local**	**p-value**
Pyrometry	2.239	1.577	$p < 4.216e - 13$
TemperatureArea1	9.584	4.479	$p < 2.2e - 16$
TemperatureArea2	4.352	2.077	$p < 2.2e - 16$
TemperatureArea3	1.765	1.081	$p < 2.2e - 16$
TemperatureArea4	3.079	2.529	$p < 2.2e - 16$
PermanenceTime	9.190	5.437	$p < 2.2e - 16$
PermanenceTimeArea1	7.977	2.962	$p < 2.2e - 16$
PermanenceTimeArea2	4.072	2.646	$p < 2.2e - 16$
PermanenceTimeArea3	3.968	2.732	$p < 2.2e - 16$
PermanenceTimeArea4	4.261	1.449	$p < 2.2e - 16$

Table 10: Mean absolute error in sample value reconstruction per variable. Statistical comparison was made using a t-test.

rates drops sharply for the phase II, isolation of real errors i.e., pinpointing the erroneous variable from the record. This situation can arise from the complexity of the detection network, which because it is densely connected. Since the best scenario to isolate a real error is when the set of variables with apparent errors corresponds to the Markov Blanket of the variable with the real error [17], with a densely connected BN the error propagates to all the network and results in all variables being part of the Markov blanket.

The local multiscale approach offers a finer discretization without the exponential growth of the conditional probability tables. There is however a price to pay for the finer discretization and better reconstruction achieved with the local submodels. The lower scale models are imposed a network structure that may not correspond to the local properties of the signals. The consequence is a higher number of false positives (global 28.7 ± 58.8, local 65.12 ± 135.34). Another reason for this is the finer discretization in the local model so that a small change in a data can move it to another interval. This effect may limit the applicability of the new model to scenarios in which Type I errors are acceptable as long as the suggested reconstruction is close enough to the real value, which is the case in the domain at hand.

According to our motivating aim, the reconstruction achieved with the local multiscale approach surpasses the reconstruction achieved by the global approach. Whilst certainly a higher number of intervals in the discretization of the global scale will result in a reconstruction with finer detail, as it has already been mentioned, higher number of intervals will require larger conditional probability tables which can be prohibitive for the global approach.

A different avenue to improve data estimation may be using a dynamic Bayesian network to consider past and future data as evidence in the model, as for instance using dynamic Bayesian networks [11], temporal nodes Bayesian networks [19], or autoregressive Bayesian networks [24]. The choice of model may be dictated by the particularities of the domain. In the problem at hand, it is plausible that the underlying process is non-stationary, but importantly the structure encoding the statistical dependencies among variables can be expected to remain unchanged. For instance, two temperature sensors will maintain their dependent records even if the process varies with time. In other words, the temperature of the process may change in time altering the statistical properties of the sensor records, but they since both sensors will be analogously affected they should remain dependent throughout the process. However, this is a particular scenario of this domain and cannot be assumed a good proxy of other problems in different domains, where changes in the statistical properties of the records may be accompanied by changes in the network structure.

8. Conclusions and future work

A new local multiscale BN based approach has been presented for the detection of rogue values in time series when statistically dependent information is known. Contrary to the existing global approach, the local multiscale solution aims to adapt the interval discretization to the neighbourhood of the sample for finer reconstruction. The new approach matches the detection capabilities of the global approach but succeeds in obtaining a significantly more accurate reconstruction. The price to pay is computational time, as one submodel must be learned for each time interval in addition to the global model which determines the submodels topology, as well as limiting the applicability, due to the higher number of false positives. Furthermore, the selection of the submodel for validation and suggestion of alternative values add additional computational burden.

The number of scales for the local model in this paper has been fixed to two (the global and one local level of the submodels). However, it is trivial to extend to a more generic solution with higher number of levels to account for periods of stationarity.

In the future we plan to split the signal in an automatic way to learn the local models. This, based on a balance between: 1) the minimum data required to learn an appropiate model, 2) perceptible changes in the time-series behavior and importantly 3) considering partitions aware of the stationarity of the resulting parts. We think that this can improve the error detection rate and signal reconstruction in the local model approach.

References

[1] (2014). Statistical package r.

[2] Abraham, B. and Box, G. (1979). Bayesian analysis of some outlier problems in time series. *Biometrika*, 66:229–236.

[3] Bao, X. and Dai, L. (2009). Partial least squares with outlier detection in spectral analysis: A tool to predict gasoline properties. *Fuel*, 88:1216–1222.

[4] Blake, N. (1993). Detecting level shifts in time series. *Journal of Business & Economic Statistics*, 11:81–92.

[5] Box, G. E., Jenkins, G. M., and Reinsel, G. C. (2013). *Time series analysis: forecasting and control*. John Wiley & Sons, 4th ed. edition.

[6] Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41.

[7] Chen, M., Zheng, A. X., Lloyd, J., Jordan, M. I., and Brewer, E. (2004). Failure diagnosis using decision trees. In *Proceedings. International Conference on Autonomic Computing, 2004.*, pages 36–43.

[8] Chen, Z., Haykin, S., Eggermont, J. J., and Becker, S. (2008). *Correlative learning: a basis for brain and adaptive systems*, volume 49. John Wiley & Sons.

[9] Chow, C. K. and Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, IT-14:462–467.

[10] D., K. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

[11] Dagum, P., Galper, A., and Horvitz, E. (1992). Dynamic network models for forecasting. In *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence*, pages 41–48. AUAI Press.

[12] Das, K. and Schneider, J. (2007). Detecting anomalous records in categorical datasets. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, U.S.A. ACM Press.

[13] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

[14] Doreswamy and Narasegouda, S. (2014). Data repository for sensor network: A data mining approach. *International Journal of Database Management Systems*, 6(4):1–9.

[15] Friedman, N. and Goldszmit, M. (1996). Discretizing continuous attributes while learning bayesian networks. In *Machine Learning, Proceedings of the Thirteenth International Conference (ICML 96)*, pages 157–165, Bari, Italy.

[16] Fukuda, K., Stanley, E. H., and Nunes Amaral, L. A. (2004). Heuristic segmentation of non-stationary time series. *Physical Review E*, 69.

[17] González, P. H. I. (1997). *Any Time Probabilistic Sensor Validation.* PhD thesis, Department of Computer & Mathematical Sciences TIME Research Institute.

[18] Gonzalez, R., Huang, B., Xu, F., and Espejo, A. (2012). Dynamic bayesian approach to gross error detection and compensation with application toward an oil sands process. *Chemical Engineering Science*, 67:44–56.

[19] Hernández-Leal, P., González, J. A., Morales, E. F., and Sucar, L. E. (2013). Learning temporal nodes bayesian networks. *International Journal of Approximate Reasoning,*, 54(8):956–977.

[20] Herrera-Vega, J. (2011). Validation data system based on a bayes network approach. In DyNaMo Research Meeting on Dynamic Probabilistic Graphical Models and Applications,Puebla, Mexico.

[21] Herrera-Vega, J., Orihuela-Espina, F., Morales, E. F., and Sucar, L. E. (2012). A framework for oil well production data validation. In Villa-Vargas, Luis; Sheremetov, L. H.-D. H., editor, *In Workshop on Operations Research and Data Mining*, page 10.

[22] Hoo, K., Tvarlapati, K., Piovoso, M., and Hajare, R. (2002). A method of robust multivariate outlier replacement. *Computers and Chemical Engineering*, 26:17–39.

[23] Ibargüengoytia, P., Vadera, S., and Sucar, L. E. (2006). A probabilistic model for information and sensor validation. *British Computer Journal*, 49(1):113–126.

[24] Ibargüengoytia, P. H., García, U. A., Herrera-Vega, J., Hernández-Leal, P., Morales, E. F., Sucar, L. E., and Orihuela-Espina, F. (2013). On the estimation of missing data in incomplete datasets: autoregressive bayesian networks. In Ege, R. and Koszalka, L., editors, *The Eighth International Conference on Systems (ICONS 2013)*, pages 111–116.

[25] Janakiram, D., Reddy, A. M., and Phani Kumar, A. (2006). Outlier detection in wireless sensor networks using bayesian belief networks. In *Proceedings of the 1st International Conference on Communication System Software and Middleware*.

[26] Kalisch, M. and Bühlmann, P. (2007). Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8(Mar):613–636.

[27] Kotsiantis, S. and Kanellopoulos, D. (2006). Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering*, 32(1):47–58.

[28] Kwiatkowski, D., Phillips, P. C. B., Schmidt, P., and Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics*, 54:159–178.

[29] Lamrini, B., Lakhal, E.-K., Lann, M.-V., and Wehenkel, L. (2011). Data validation and missing data reconstruction using self-organizing map for water treatment. *Neural Computing and Applications*, 20:575–588.

[30] Lemos, A., Caminhas, W., and Gomide, F. (211). Adaptive fault detection and diagnosis using an evolving fuzzy classifier. *Information Sciences*, 220:64–85.

[31] Ligges, U., Weihs, C., and Hasse-Becker, P. (2002). Detection of locally stationary segments in time series. In *Proceedings of the 15th Symposium in Computational Statistics (Compstat'02)*, pages 285–290.

[32] Makridakis, S. and Hibon, M. (1997). Arma models and the box–jenkins methodology. *Journal of Forecasting*, 16(3):147–163.

[33] Muirhead, C. (1986). Distinguishing outlier types in time series. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48:39–47.

[34] Pearl, J. (1982). *Probabilistic reasoning with intelligent systems.* Morgan and Kaufmann,, 1st ed. edition.

[35] Peng, J., Peng, S., and Hu, Y. (2012). Partial least squares and random sample consensus in outlier detection. *Analytica Chimica Acta*, 719:24–29.

[36] Roth, D. (1996). On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2):273–302.

[37] Sharifi, R. and Langari, R. (2013). Sensor fault diagnosis with a probabilistic decision process. *Mechanical Systems and Signal Processing*, 34:146–155.

[38] Siaterlis, C. and Maglaris, B. (2004). Towards multi-sensor data fusion for DoS detection. In *Proceedings of the ACM Symposium on Applied Computing*, pages 439–446. ACM Press.

[39] Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, Prediction, and Search. Adaptive Computation and Machine Learning.* MIT Press, 2nd. ed. edition.

[40] Steck, H. (2001). *Constrained-Based Structural Learning in Bayesian Networks Using Finite Data Sets.* PhD thesis, Institut for der Informatik der Technischen Universität München.

[41] Stoer, J. and Bulirsch, R. (2002). *Introduction to numerical analysis.* Springer.

[42] Tamrapani, D. and Johnson, T. (2003). *Exploratory Data Mining and Data Cleaning*. John Wiley, 1st edition edition.

[43] Tsay, R. S. (1988). Outliers, level shifts, and variance changes in time series. *Journal of Forecasting*, 7:1–20.

[44] Tylman, W. and Anders, G. J. (2006). Application of probabilistic networks for decision support in power system analysis. *Energy*, 31:2874–2889.

[45] Walczak, B. (1995). Outlier detection in multivariate calibration. *Chemometrics and Intelligent Laboratory Systems*, 28:259–272.

[46] Zhang, Y., Meratnia, N., and Havinga, P. (2010). Outlier detection techniques for wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials*, 12(2).

Appendix A. Synthetic data topologies

Three sets of synthetic data were generated. Each set has been generated according to the topologies showed in Figure 7 and the details of this are described as follow.

First, a number of signal seeds equal to the number of nodes on the topology are defined. These seeds are expressed in terms of existing univariate time series models such as AutoRegressive Moving Average (ARMA), AutoRegressive Integrated Moving Average (ARIMA) or seasoned ARIMA (sARIMA). The notation $ARMA(p,q)$ refers to the ARMA model with p autoregressive terms and q moving-average terms in Eq. 8:

$$S(t) = \epsilon_t + \sum_{i=1}^{p} \alpha_i S(t-i) + \sum_{i=1}^{q} \beta_i \epsilon_{t-i} \tag{8}$$

Where $\alpha_1, ..., \alpha_p$ are parameters of an Autoregressive model(AR), $\beta_1, ..., \beta_q$ are the respective parameters of a Moving Average model (MA) and the random variable ϵ_t is white noise [5]. The notation $ARIMA(p,d,q)$ denotes an ARIMA model, which is a generalization of an ARMA model. In the ARIMA model, the parameters p,d and q are non-negative integers where p and q are the same like in ARMA models and d is the integrated part [32]. Finally, a seasonal effect can be added using an sARIMA model. An sARIMA model has two pairs of parameter triplets (p, d, q), the first triplet for the ARIMA model, and the last one for the seasonal component.

Once the seeds are defined, these are combined so that each node in the synthetic topology is a combination of its own seed plus maybe some influence from the other nodes as depicted in Figure 7. Following, the generated topologies are described.

Topology 1. For the first topology we have the following seeds:

$S_1(t) \quad = ARMA(2,1)$ with $\alpha_1 = 0.6$, $\alpha_2 = -0.3$ and $\beta_1 = 0.5$
$S_2(t) \quad = ARIMA(2,1,1)$ with $\alpha_1 = 0.9$, $\alpha_2 = -0.6$ and $\beta_1 = 0.8$
$S_3(t) \quad = sin(0.2ARIMA(0,0,0) + \pi)$
$S_4(t) \quad = 5.3cos(2\pi t/300) + 2$

And the nodes of the topology are defined by:

$$N_1 = S_1 \tag{9}$$
$$N_2 = 3.4N_1 + 0.7S_2 \tag{10}$$
$$N_3 = 7.1N_1S_3 \tag{11}$$
$$N_4 = 3.2N_3S_4 \tag{12}$$

Topology 2. In the second topology, the seeds are:

$S_1(t) \quad = ARIMA(1,1,1)$ with $\alpha_1 = -0.1$ and $\beta_1 = 0.8$
$S_2(t) \quad = ARIMA(2,1,1)$ with $\alpha_1 = 0.78$, $\alpha_2 = -0.378$ and $\beta_1 = 0.01$
$S_3(t) \quad = sin(0.2ARIMA(1,1,1) + \pi)$ with $\alpha_1 = 0.58$ and $\beta_1 = 0.03$
$S_4(t) \quad = sARIMA((2,1,1),(2,0,1))$ with $\alpha_1 = 0.1$, $\alpha_2 = -0.6$ and $\beta_1 = 0.8$ for the ARIMA model, and $\alpha_1 = 0.6$, $\alpha_2 = -0.8$ and $\beta_1 = 0.3$ for the seasonal component.

and the nodes are specified by:

$$N_1 = S_1 \tag{13}$$
$$N_2 = 3.4N_1S_2 \tag{14}$$
$$N_3 = 7.1N_1S_3 + N_4 \tag{15}$$
$$N_4 = S_4 \tag{16}$$

Topology 3. For the third topology, we have the following seeds:

$S_1(t) = ARIMA(1,1,1)$ with $\alpha_1 = -0.4$ and $\beta_1 = 0.3$

$S_2(t) = ARIMA(2,1,2)$ with $\alpha_1 = 0.78$, $\alpha_2 = -0.378$ and $\beta_1 = 0.01$, $\beta_2 = -0.1$

$S_3(t) = \theta sin(0.2\theta + \pi)$ where $\theta = ARIMA(0,0,0)$

$S_4(t) = sARIMA((2,1,1),(2,0,1))$ with $\alpha_1 = 0.1$, $\alpha_2 = -0.6$ and $\beta_1 = 0.8$ for the ARIMA model, and
$\alpha_1 = 0.6$, $\alpha_2 = -0.8$ and $\beta_1 = 0.3$ for the seasonal component.

$S_5(t) = sARIMA((1,1,1),(2,0,1))$ with $\alpha_1 = -0.8$ and $\beta_1 = 0.8$ for the ARIMA model, and
$\alpha_1 = 0.6$, $\alpha_2 = -0.8$ and $\beta_1 = 0.11$ for the seasonal component.

and the nodes are specified by:

$$N_1 = S_1 \tag{17}$$

$$N_2 = 3.4N_1S_2 \tag{18}$$

$$N_3 = 7.1S_3N_4 + 3.2N_1 \tag{19}$$

$$N_4 = 1.01S_4N_1 \tag{20}$$

$$N_5 = -0.18S_5 + N_4 \tag{21}$$

Topology complexity is related to (i) the number of existing elements in the set, which in the case of the graph, is given by the nodes and number of connections between them, (ii) and the distribution of these connections related to the number of ways that it is possible to travel from a certain element to a connected element. Topology 1 has only 4 nodes and 3 edges with all elements having at most 1 parent i.e. there is only one path to reach a certain element. Topology 2, maintains 4 nodes and 3 edges (same number of elements as topology 1) but increases the complexity by allowing node 3 to be reached from two different nodes. Finally, topology 3 increases the complexity of the two previous topologies with more elements (both nodes and edges), and further incorporating alternative routes to go from one node to another e.g. N1 to N3.