Spatiotemporal Models for Data-Anomaly Detection in Dynamic Environmental Monitoring Campaigns

Ethan W. Dereszynski and Thomas G. Dietterich
Department of Electrical Engineering and Computer Science
Oregon State University
{dereszet,tgd}@eecs.oregonstate.edu

The ecological sciences have benefited greatly from recent advances in wireless sensor technologies. These technologies allow researchers to deploy networks of automated sensors, which can monitor a landscape at very fine temporal and spatial scales. However, these networks are subject to harsh conditions, which lead to malfunctions in individual sensors and failures in network communications. The resulting data streams often exhibit incorrect data measurements and missing values. Identifying and correcting these is time-consuming and error-prone. We present a method for real-time automated data quality control (QC) that exploits the spatial and temporal correlations in the data to distinguish sensor failures from valid observations. The model adapts to each deployment site by learning a Bayesian network structure that captures spatial relationships between sensors, and it extends the structure to a dynamic Bayesian network to incorporate temporal correlations. This model is able to flag faulty observations and predict the true values of the missing or corrupt readings. The performance of the model is evaluated on data collected by the SensorScope Project. The results show that the spatiotemporal model demonstrates clear advantages over models that include only temporal or only spatial correlations, and that the model is capable of accurately imputing corrupted values.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning—*Parameter learning*; I.5.1 [**Pattern Recognition**]: Models—*Statistical*; *structural*; G.3 [**Mathematics of Computing**]: Probability and Statistics—*Distribution functions*; *Markov processes*; *multivariate statistics*

General Terms: Algorithms, Reliability, Verification
Additional Key Words and Phrases: Anomaly detection, Bayesian modeling, environmental monitoring, quality control, wireless sensor networks

1. INTRODUCTION

The increasing availability (coupled with decreased cost) of lightweight, automated wireless sensor technologies is changing the way ecosystem scientists collect and dis-

Ethan W. Dereszynski and Thomas G. Dietterich, School of Electrical Engineering and Computer Science, 1148 Kelley Engineering Center, Oregon State University, Corvallis, OR 97331-5501.
This work was supported under the National Science Foundation (NSF) Ecosystem Informatics IGERT (grant number DGE-0333257) and funding from the EPFL.
Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.
© 20YY ACM 0000-0000/20YY/0000-0001 $5.00

tribute data. Portable sensor stations allow field experts to transport monitoring equipment to sites of interest and observe ecological phenomena at a spatial granularity of their choosing. These nonpermanent deployments stand in stark contrast to traditional observatory-like environmental monitoring stations whose initial spatial layout remains unchanged over the course of time. However, both approaches are providing researchers with an unprecedented volume of ecological data. The resultant surge in data has potential to transform ecology from an analytical and computational science into a data exploration science [Szalay and Gray 2002].

Temporary sensor deployments, whose durations can range from a single week to several months, represent a new challenge for data quality control. By nature of being in-situ environmental stations, they are prone to the same technical problems as long-term deployments, namely damage due to extreme weather, transmission errors and loss of signal, calibration errors, and drastic changes in environmental conditions. Further, it is important to rapidly detect and diagnose a damaged or failing sensor so that it can be repaired. An insufficiently fast diagnosis could result in the corruption or loss of data from a given sensor for the duration of the deployment; consequently, techniques involving a postmortem analysis of the data are of little or no value. However, the sheer abundance of data provided by large sensor networks operating at fine time resolutions makes manual analysis (visualizing the data) infeasible for both online and offline quality control. This raises the need for efficient automated methods of data "cleaning" that can function in an online setting and that can readily adapt to dynamic spatial distributions.

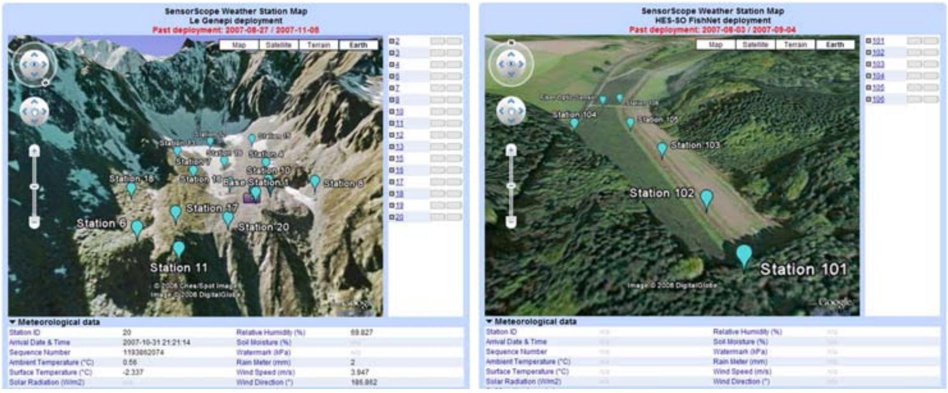
The purpose of this article is to provide an example of spatially distributed environmental monitoring, motivate a need for quality control in this domain through documented examples of sensor failure, and introduce a machine learning approach to automate the data cleaning process. Though we believe our methodology is readily extendable to additional environmental phenomena, our work here deals only with air temperature data. We propose an adaptive quality control (QC) system that exploits both temporal and spatial relationships among multiple environmental sensors at a site. The QC system makes use of a dynamic Bayesian network (DBN, [Dean and Kanazawa 1988]) to correlate sensor readings within a sampling period (time step) to readings taken from past sampling periods. Because the set of potential faults is unbounded, it is not practical to approach this as a diagnosis problem where each fault is modeled separately [Hodge and Austin 2004]. Instead, we employ a general fault model and focus on creating a highly accurate model of normal behavior, known as the process model. The intuition is that if there is a discrepancy between the current estimate of normal behavior (provided by the process model) and the observation taken from the sensor, then the observation is labeled as anomalous. An additional benefit of this approach is that it can impute values for the sensor readings during periods of sensor malfunction.

This article is organized as follows. First, we will discuss the current ecological monitoring campaign, known as SensorScope, that produced the data studied herein. Second, we describe the nature of the air temperature data and the data-anomaly types encountered, followed by a introduction to hybrid Bayesian networks. Third, we describe our quality control model, including learning the process model and incorporating a general fault model. Finally, we present the results of the model applied to temperature data from select SensorScope deployments as well as empirical results on synthetic data. We conclude with a plan for future research.

2. THE SENSORSCOPE SYSTEM

The SensorScope Station, developed at the École Polytechnique Fédérale de Lausanne (EPFL) in Switzerland, represents a significant change in traditional tools for in-situ data collection and distribution. In place of few, expensive long-term or permanent monitoring stations deployed sparsely over a heterogeneous spatial area, SensorScope allows field scientists to deploy many light weight, inexpensive stations at a much higher spatial resolution and monitor at user-specified time granularities. The portability of these stations facilitates dynamic deployments, wherein sensors can be relocated within a deployment to adapt to changing monitoring requirements.

Component sensors for measuring air temperature, skin (surface) temperature, wind speed, wind direction, humidity, etc. are typically acquired from external manufacturers. The SensorScope stations are equipped with a power supply sufficient to host a small set of these sensors (the number dependent on each sensor's energy requirement) operating simultaneously, as well as a radio device for communication with nearby stations. Every deployment contains at least one General Packet Radio Service (GPRS) hub that transmits data received from the SensorScope stations to a central server at the EPFL via cellular signal. Once the data reaches the central server, it is converted from its raw voltage to a value particular to the phenomenon being measured (degrees Celsius in the case of temperatures) via a conversion formula specific to the sensor type. When the data is requested for download or plotting via the SensorScope Web site,[1] it is filtered automatically by a range checker to remove extreme values associated with sensor malfunctions.



▾ Meteorological data

Station ID	20	Relative Humidity (%)	69.827
Arrival Date & Time	2007-10-31 21:21:14	Soil Moisture (%)	n/a
Sequence Number	1193862074	Watermark (kPa)	n/a
Ambient Temperature (°C)	0.56	Rain Meter (mm)	2
Surface Temperature (°C)	-2.337	Wind Speed (m/s)	3.947
Solar Radiation (W/m2)	n/a	Wind Direction (°)	186.862

▾ Meteorological data

Station ID	n/a	Relative Humidity (%)	n/a
Arrival Date & Time	n/a	Soil Moisture (%)	n/a
Sequence Number	n/a	Watermark (kPa)	n/a
Ambient Temperature (°C)	n/a	Rain Meter (mm)	n/a
Surface Temperature (°C)	n/a	Wind Speed (m/s)	n/a
Solar Radiation (W/m2)	n/a	Wind Direction (°)	n/a

Fig. 1. The Genepi Glacier and FishNet SensorScope deployments (from Google Earth)

Figure 1 (left) shows a 3-D visualization of the Le Genepi Glacier deployment, which was in place from August 27 to November 5 of 2007. The glacial valley,

[1] http://sensorscope.epfl.ch/index.php/SensorScope_Deployments

located approximately 60 kilometers south of the western edge of Lake Geneva, slopes downward toward the northeast and is surrounded by mountains on all other sides. The sensors are placed at an elevation range of 2300 meters to 2500 meters. At the time of the deployment, the northeast corner of the glacier was the only area accessible by cellular signal; therefore, the GPRS (labeled “Base Station 1”) was placed in this location. A total of 16 stations were deployed over the area, whose dimensions can be roughly approximated by a 100 meter by 200 meter rectangle, to allow for a comprehensive analysis incorporating the spatial heterogeneity of the relatively small region.

The right portion of Figure 1 shows a much smaller deployment of six stations along a stream, known as the FishNet deployment. The sensors operated from from August 3 to September 4 of 2007. The topographical difference is relatively small compared to Le Genepi (the sensors are all located at approximately 600 meters of elevation), as the deployment was in an agricultural area bordering a forested area to the south. The GPRS station (not shown in the figure) is located approximately 100 meters to the west of Station 104, and the length of stream covered by the deployment is approximately 300 meters.

3. SENSORSCOPE DATA

Each SensorScope station is capable of hosting a changing set of environmental sensors; hence, there is not a consistent set of phenomena recorded by all stations across all deployments. Rather, each station is provided only with those sensors needed to measure the variables of interest for a given field campaign and is then retooled between deployments. As air temperature (the temperature roughly 1.5 meters above the surface) is of interest in nearly all campaigns to date, we shall focus our discussion primarily on this type of data. Air temperature readings are taken from Sensirion SHT75 sensors mounted on the SensorScope stations [Sensirion 2005]. Figures 2 and 3 show two different sets of data streams from the stations at the FishNet and Le Genepi deployments, respectively. The air temperature readings from both sites were sampled at a rate of once every two minutes. The graphs show those readings binned and averaged into 10-minute windows.

Nominal air temperature data contains a regular diurnal (day to day) trend that is dependent on the season and location of the sensor. For example, the FishNet has a more pronounced diurnal signal because the recording period is in the late summer (August) whereas the the Le Genepi deployment has a suppressed diurnal trend due to both the time it was observed (October) and its Alpine location. Storm and cloud coverage events occur at irregular intervals but may also suppress diurnal signal. The FishNet data shows the effect of a storm in days 2 through 4. We have found the following data-anomaly types present in the air temperature data:

—*GPRS Outage.* In the case where the GRPS hub becomes inoperative, data for the entire deployment is lost. The FishNet deployment contains many segmented periods of sensor outages among all stations. These outages indicate a failure of the GPRS, because they occur simultaneously across all sensor streams. The faults are evident between days 15-16 and days 17-20.

—*Sensor Outage.* A sensor outage occurs when an individual sensor stream is lost. Multiple sensor outages can overlap during a give time period; however, unlike in

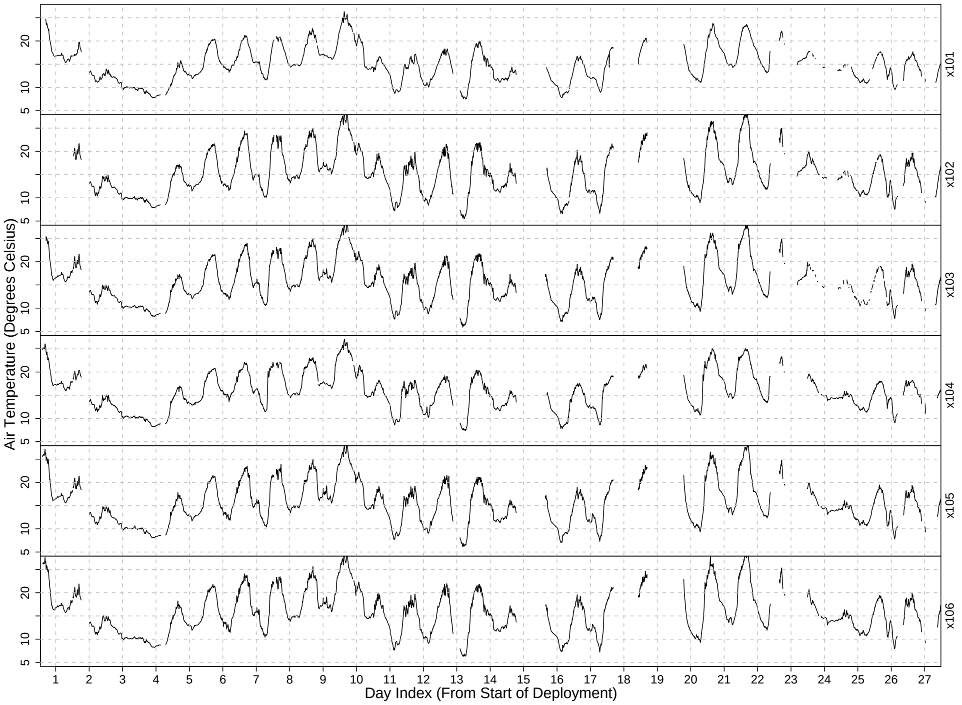


Fig. 2. Air temperature readings from the FishNet SensorScope deployment. Each row represents the sensor labeled in the corresponding upper right corner. The X axis denotes the day (vertical dashed line depicts midnight) since the deployment began, and the Y axis denotes temperature in degrees ℃. Corresponding station names appear on the right side of the graph next to the stream they depict.

a GPRS outage, the start, end, and duration of each outage is not synchronized. There are individual sensor outages in the FishNet deployment at stations 101, 102, and 103, spanning days 24 and 25.

—*Data Anomalies.* Data anomalies are characterized as observations from a given sensor that are corrupted due to sensor malfunction. Such anomalous values are particularly obvious at station 6 in the Le Genepi deployment (Figure 3), where extremely large temperature values are recorded due to incorrect voltages generated at the sensor. Subtler spikes in temperature occur at station 6 on day 41 (Le Genepi) and station 101 on day 17 (FishNet). A flatline in temperature is created by the temperature sensor reporting a 0-voltage value. The conversion algorithm maps this value to -1 ℃ value upon storing it into a data base. An example of this error is provided in Section 6.2.

Given the correlation between the sensors within a deployment, it is our goal to be able to identify data anomalies and impute the true temperature values in the case of both individual sensor outages and sensor malfunctions. Sensor failures that manifest themselves as either extraordinarily hot or cold temperatures are simple to diagnose by means of range checking [Mourad and Bertrand-Krajewski 2002];

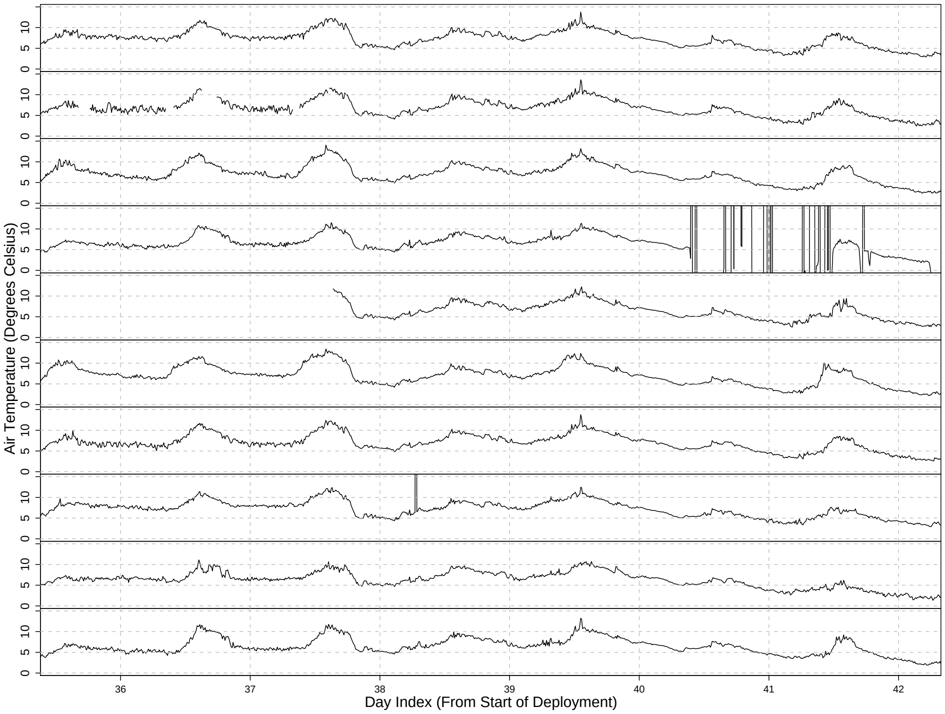


Fig. 3. Air temperature readings from one week at the Le Genepi deployment. Each row represents the sensor labeled in the corresponding upper right corner. The X axis denotes the day (vertical dashed line depicts midnight) since the deployment began, and the Y axis denotes temperature in degrees ℃. Corresponding station names appear on the right side of the graph next to the stream they depict.

however, malfunctions resulting in flatline values and subtler spikes in temperature readings are not detected by extreme value tests. While the values appear anomalous in the context of their immediate temporal neighbors, they are not abnormal in the range of temperatures recorded over the full duration of the deployment.

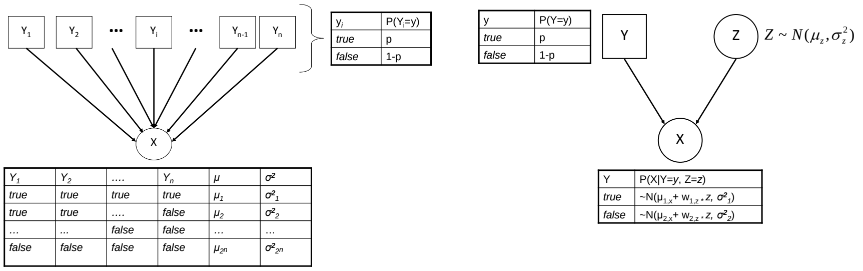
4. HYBRID BAYESIAN NETWORKS

Our probabilistic model of the air temperature domain is a conditional linear-Gaussian network, also known as a hybrid network because it contains both continuous and discrete variables [Lauritzen and Wermuth 1989; Murphy 1998; Pearl 1988]. For the sake of computational convenience, we will restrict our networks so that discrete-valued variables do not have continuous-valued parents and so that all continuous-valued variables are modeled as Gaussians.

In this section, we describe how the probability distributions for continuous-valued variables are parameterized. We consider three cases: (a) continuous variables with discrete parents, (b) continuous variables with continuous parents, and (c) continuous variables with a mix of discrete and continuous parents.

4.1 Continuous Variables with Discrete Parents

Consider a single continuous variable, X. For every possible instantiation of values for the discrete parents of X, X takes on a Gaussian distribution with separate values for μ and σ^2. For example, if X has a single Boolean parent, $Y = y \in \{true, false\}$, then the conditional probability table (CPT) of X would contain two entries: $P\left(X|y = true\right) \sim N\left(\mu_t, \sigma_t^2\right)$ and $P\left(X|y = false\right) \sim N\left(\mu_f, \sigma_f^2\right)$. In general, let $\mathbf{Y} = \{Y_1, Y_2, ..., Y_n\}$ denote the set of discrete parents of the continuous variable X. Further, let $|\mathbf{Y}| = |Y_1| \times |Y_2| \times ... \times |Y_n|$ be the total size (number of possible instantiations) of $\mathbf{Y}$. Then, we specify the CPT of X with the $|\mathbf{Y}|$ dimensional vector, $\vec{\mu} = \left\langle \mu_1, \mu_2, ..., \mu_{|\mathbf{Y}|} \right\rangle$. Similarly, we specify the set of variances of X, depending on the parent configuration, as the vector $\vec{\sigma^2} = \left\langle \sigma_1^2, \sigma_2^2, ..., \sigma_{|\mathbf{Y}|}^2 \right\rangle$. Figure 4 (left) contains an example with binary discrete variables.



y_i	$P(Y_i=y)$
true	p
false	1-p

Y_1	Y_2	….	Y_n	μ	σ^2
true	*true*	*true*	*true*	μ_1	σ^2_1
true	*true*	….	*false*	μ_2	σ^2_2
…	…	*false*	*false*	…	…
false	*false*	*false*	*false*	μ_{2n}	σ^2_{2n}

y	P(Y=y)
true	p
false	1-p

Y	P(X|Y=*y*, Z=*z*)
true	$\sim N(\mu_{1,x} + w_{1,z} * z, \sigma^2_1)$
false	$\sim N(\mu_{2,x} + w_{2,z} * z, \sigma^2_2)$

Fig. 4. Left: Conditional Gaussian Bayesian Network. Right: Conditional Linear-Gaussian Bayesian Network

4.2 Continuous Variables with Continuous Parents

Consider a single continuous variable, X, but now with m continuous-valued parents. For each continuous-valued parent, $Z_i \in \mathbf{Z} = \{Z_1, Z_2, ..., Z_m\}$, X has a weight, w_i, such that mean of X is calculated as:

$$\mu_x = \epsilon + \sum_{i=1}^{m} w_i z_i, \tag{1}$$

where z_i is the value of the parent random variable, Z_i, and ϵ is X's "intercept term" in the linear regression formula. An essential requirement for computational tractability is that the variance of X is specified by a single σ^2 parameter and which is not conditioned on the parents. Note that this conditional distribution has exactly the same form as a linear regression model.

4.3 Continuous Variables with a Mix of Discrete and Continuous Parents

If a variable has a mix of continuous and discrete parents, we employ a distinct linear Gaussian distribution for each combination of values of the discrete parents. This is known as a Conditional Linear Gaussian (CLG) model. Let X be a continuous variable with a set of discrete parents, $\mathbf{Y}$, and continuous parents, $\mathbf{Z}$. Then X has a separate mean, variance, and set of regression weights for each possible instantiation of $\mathbf{Y}$. We specify a CLG variable by a mean vector, $\vec{\mu} = \left\langle \mu_1, \mu_2, ..., \mu_{|\mathbf{Y}|} \right\rangle$, a variance vector, $\vec{\sigma^2} = \left\langle \sigma_1^2, \sigma_2^2, ..., \sigma_{|\mathbf{Y}|}^2 \right\rangle$, and a $|\mathbf{Y}| \times |\mathbf{Z}|$ regression matrix:

$$\begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,|\mathbf{Z}|} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,|\mathbf{Z}|} \\ w_{...,1} & w_{...,2} & \cdots & w_{...,|\mathbf{Z}|} \\ w_{|\mathbf{Y}|,1} & w_{|\mathbf{Y}|,2} & \cdots & w_{|\mathbf{Y}|,|\mathbf{Z}|} \end{bmatrix}. \tag{2}$$

Figure 4 (right) contains a CLG network with a continuous variable having one discrete and one continuous parent.

5. DEVELOPING A SPATIOTEMPORAL PROCESS MODEL

In the following sections, we describe our process model as the combination of two individual pieces: the spatial component that represents the relationships among all sensors within a deployment for a given time slice, and a temporal component that captures the transition dynamics from one time period (10-minute interval) to the next. The complete process model is represented as a Dynamic Bayesian Network [Dean and Kanazawa 1988], in which the task of quality control is achieved by inferring the most likely state of the sensors given the current temperature observations and those of the immediate past. The procedure for building the complete quality control model is summarized in Table I.

5.1 Structure Learning and the Spatial Component

As the geographical layout of stations changes with each deployment, it is desirable to have the QA routines autonomously learn the spatial relationships for each new deployment from the observed data. To this end, we apply Bayesian network structure learning algorithms to learn the set of conditional-independence relationships among sensors at a given deployment. Recall that though each SensorScope station is capable of monitoring several environmental variables according to the type and number of sensors it is hosting, we focus our work to air temperature data for purposes of site-to-site continuity and ignore other data types. Further, we assume that each set of air temperature observations corresponding to a single 10-minute period is generated from a multivariate Gaussian distribution, and thus a sample from a deployment containing n SensorScope stations is generated from an n-dimensional multivariate Gaussian. Our goal then is to learn the elements of the covariance matrix that determine how each dimension (each sensor in a deployment) relates to the others. Our assumptions facilitate the application of a measure known as BGe (Bayesian metric for Gaussian networks having score equivalence, developed by Geiger and Heckerman [1994]) as a scoring metric for candidate networks. We summarize the scoring metric, but ask the interested reader to see the aforementioned reference for further details.

A Bayesian network over n Gaussian distributed variables has a joint distribution equal to a n-dimensional multivariate Gaussian. The network structure is referred to as a sparse representation of the multivariate distribution. “Sparse” in this context means that the representative network may not directly correlate each variable with all other variables; in graphical terms, less than $\frac{n(n-1)}{2}$ edges (the maximum amount of edges for an acyclic graph) are sufficient to represent the covariance structure among n variables. In general, a lesser degree of connectivity in the graph structure will decrease the time required to perform inference in the model and reduce the number of parameters to fit once the structure is determined.

The BGe metric assumes the existence of a prior linear Gaussian network whose full joint distribution represents an initial estimate of the true distribution from which the observations are drawn. This Bayesian network can either be knowledge-engineered by domain experts with information about the deployment or constructed ad hoc in cases where specific domain knowledge is absent. Geiger and Heckerman parameterize the unknown generative, multivariate distribution with a mean vector, $\vec{m}$, and precision matrix, $W = \Sigma^{-1}$. The prior joint distribution on these parameters is assumed to be a normal-Wishart. Under these assumptions, the joint posterior distribution, given a data set D (containing multivariate observations $\vec{x}_1, \vec{x}_2, ..., \vec{x}_l$, each of n dimensions), over $\vec{m}$ and W can be divided into the conditional distribution $P(\vec{m}|W)$ and the marginal $P(W)$. The conditional distribution of $P(\vec{m}|W)$ is given as a multivariate normal

$$P(\vec{m}|W) \sim N\left(\vec{\mu}_l = \frac{v\vec{\mu}_0 + l\bar{X}_l}{v+l}, (v+l)W\right) \tag{3}$$

where v encodes the strength of the prior in terms of an equivalent number of “prior” observations and l is the number of “new” observations in the data set D. The posterior of W is distributed itself as a $Wishart(\alpha + l, T_l)$, where α specifies the degrees of freedom the Wishart distribution (for this reason, $\alpha \geq n$ must be satisfied). $\bar{X}_l$ and S_l are the sample mean and covariance of D, and $\vec{\mu}_0$ and Σ_0 are the mean vector and covariance of the prior network structure. The matrices T_l and T_0 are calculated as

$$T_l = T_0 + S_l + \frac{vl}{v+l}\left(\vec{\mu}_0 - \bar{X}_l\right)\left(\vec{\mu}_0 - \bar{X}_l\right)' \tag{4}$$

$$T_0 = \Sigma_0 \frac{v(\alpha - n - 1)}{v+1}. \tag{5}$$

T_0 is the precision matrix of the of the prior marginal distribution on W (before the observed data D is introduced), given as $P(W) \sim Wishart(\alpha, T_0)$. The BGe metric scores the likelihood of an hypothesized network structure B_s given a data set D and the prior ξ as

$$P(D|B_s,\xi) = \prod_{i=1}^{n} \frac{P\left(D_{x_i,\Pi_i}|B_s^c,\xi\right)}{P\left(D_{\Pi_i}|B_s^c,\xi\right)}, \tag{6}$$

where $P\left(D_{x_i,\Pi_i}|B_s^c,\xi\right)$ is the local score of the data relevant only to variable, x_i, and its parents, Π_i. Specifically, we keep only those rows and columns of the T_0 and T_l that correspond to variables in $x_i \cup \Pi_i$ in the case of the numerator in (6). Similarly for the denominator, we keep only those rows and columns in T_0 and T_l corresponding to the variables in Π_i. The term B_s^c represents a fully connected Gaussian Network with edges among all variables. Both the numerator and denominator in (6) are calculated as follows:

$$P\left(D|B_s,\xi\right) = \left(2\pi\right)^{\frac{-nl}{2}} \left(\frac{v}{v+l}\right)^{\frac{n}{2}} \frac{c\left(n,\alpha\right)}{c\left(n,\alpha+l\right)} \left|T_0\right|^{\frac{\alpha}{2}} \left|T_l\right|^{-\frac{\alpha+l}{2}} \tag{7}$$

$$c\left(n,\alpha\right) = \left[2^{\frac{n\alpha}{2}} \pi^{\frac{n(n-1)}{4}} \prod_{i=1}^{n} \Gamma\left(\frac{\alpha+1-i}{2}\right)\right]^{-1} \tag{8}$$

To score an entire network, the expression in (6) must be evaluated or, equivalently, the expression in (7) must be evaluated for each variable in the domain and then each resultant value multiplied together. In the case of a nonuniform prior over network structures, an additional weighting of $P\left(B_s|\xi\right)$ should be factored into (6).

Provided with a scoring metric for Linear-Gaussian Bayesian Networks, we implement a simple hill-climbing algorithm to find a good structure for the networks. The algorithm is initialized with a prior network structure, then it takes one of the following actions: (1) add an arc between variables x_i and x_j if no arc existed already; (2) remove an existing arc between two variables; or (3) reverse an existing arc between two variables. All three options are undertaken with the constraint that the resulting structure must remain acyclic. Each of the possible networks created by taking one of the these actions is evaluated using the BGe metric, and the action resulting in the highest scoring network is taken. The resultant network then becomes the initialization point for another application of this hill-climbing search. The process is repeated until taking a single action (adding, removing, or reversing an arc) creates no increase in the score, in which case an optimum has been reached. Unfortunately, this hill-climbing methodology is subject to local optima, and so the final network is perturbed. This perturbation is achieved by examining every existing edge in the current structure and, with some probability, removing the edge, reversing the direction of the edge (pending no cycle is created), or making no change. In cases where no edge exists between two variables, we consider adding an edge (pending no introduced cycle) or taking no action. The complete algorithm halts after performing a user-specified number of perturbations/restarts, and the the best-scoring network is returned. We outline the aforementioned hill-climbing algorithm in Algorithm 1.

It is important to note that structure returned from this hill-climbing search may not be unique relative to its score. The BGe metric demonstrates a property known as Score Equivalence, which means that it scores network structures belonging to the same Markov Equivalence Class (MEC) equally. An MEC is the set of graphs that represent the same set of conditional independence relationships between variables. Moreover, the algorithm described above only returns the structure of the network (i.e., the set of parent-child arcs); it does not compute the parameter values for each variable (means, variances, and regression weights). In Section 5.4, we describe how we arrive at these values by computing the Maximum Likelihood Estimates (MLE) for each parameter directly from the data set, D.

The BGe metric is considered a local scoring function because of its decomposition into summing over of node child/parent configurations. Specifically, if we consider taking the log likelihood of the probability in (6), we arrive at the following summation:

Algorithm 1 Hill-climbing with BGe Metric

```
Input: An initial Bayesian Network: $B_{init}$
Input: Number of perturbations to perform: $pturb$

$CurrentScore = BGe\,(B_{init})$
$CurrentNet = B_{init}$
$BestScore = BGe(B_{init})$
$BestNet = B_{init}$
for $i = 1$ to $pturb$ do
    $LastScore = -\infty$
    while $CurrentScore > LastScore$ do
        $LastScore = CurrentScore$
        $AddNet = AddArc(CurrenNet)$
        $RemNet = RemoveArc(CurrentNet)$
        $RevNet = ReverseArc(CurrentNet)$
        $NextNet = \text{argmax}_{net}\,[BGe(AddNet), BGe(RemNet), BGe(RevNet)]$
        if $BGe(NextNet) > CurrentScore$ then
            $CurrentScore = BGe(NextNet)$
            $CurrentNet = NextNet$
        end if
    end while
    if $CurrentScore > BestScore$ then
        $BestScore = CurrentScore$
        $BestNet = CurrentNet$
    end if
    $CurrentNet = Perturb(CurrentNet)$
    $CurrentScore = BGe(CurrentNet)$
end for
return $BestNet$
```

$$\log P\,(D|B_s,\xi) \;=\; \sum_{i=1}^{n} [\log P\,(D_{x_i,\Pi_i}|B_s^c,\xi) - \log P\,(D_{\Pi_i}|B_s^c,\xi)]. \tag{9}$$

We can then compute the log likelihood equivalent of expression Equation 7 to compute the terms within the summation.

$$\begin{aligned}
\log P\,(D|B_s,\xi) \;&=\; \log\left[(2\pi)^{\frac{-nl}{2}}\left(\frac{v}{v+l}\right)^{\frac{n}{2}}\frac{c\,(n,\alpha)}{c\,(n,\alpha+l)}\,|T_0|^{\frac{\alpha}{2}}\,|T_l|^{-\frac{\alpha+l}{2}}\right] &(10)\\
&=\; \left[\frac{-nl}{2}\log\,(2\pi)\right] + \left[\frac{n}{2}\log\left(\frac{v}{v+l}\right)\right] &(11)\\
&+\; [\log c\,(n,\alpha) - \log c\,(n,\alpha+l)] + \left[\frac{\alpha}{2}\log|T_0|\right] &(12)\\
&+\; \left[-\frac{\alpha+l}{2}\log|T_l|\right] &(13)
\end{aligned}$$

Once in summation form, it becomes clear that computing the BGe score for an entire network is only necessary for the initial prior network, B_{init}. Each subsequent change in the network structure by adding, removing, or reversing an arc only requires a modification of some factor in the score in (9). For example, if we add an arc from variable x_i to x_j, then only the parent set of x_i has changed; consequently, we only need to recompute the i^{th} term in (9). The score for the resulting network would be the original network score minus the original i^{th} term (the one *not including* x_j as a parent) plus the new i^{th} term (the one *including* x_j as a parent).

Figure 5 contains a learned structure for the FishNet deployment. The initial prior network assumed complete independence among all six sensor stations at the site and placed a standard Normal distribution over all sensors (mean of 0 and variance of 1.0). The training set is constructed from observations from the SensorScope stations themselves. As we have no ground-truth data for the true temperature variables (X_i's), we consider those observations not excluded by the website range-checker or representative of a flatline sensor failure (i.e., consecutive -1 ℃ values) to be the "true" temperature values. The structure was learned using data from the first half of the deployment; however, the training set was limited to only those observations where all sensors reported a value. If a sample missed at least 1 observation (one sensor failed to report within the 10-minute window), then it was rejected.

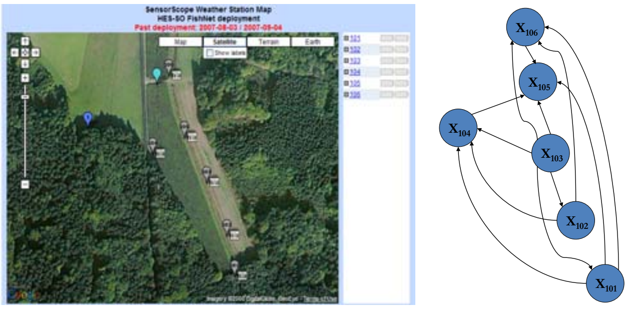


Fig. 5. Left: Top-down view of the FishNet Deployment. Right: Learned dependency relationships among the six sensor stations at the deployment.

5.2 Incorporating a Temporal Model

The spatial model, learned from the methods described in the previous section, captures much of the correlative relationship among multiple sensors within a given deployment; however, the model suffers from some significant drawbacks. Foremost, the spatial model ignores the transition dynamics present in ecological data – a single sample taken from all sensors in a time step is considered independent of all temporally nearby samples (the sample taken 10 minutes ago, 20 minutes ago, etc.). Many types of ecological data are highly autocorrelated. In the case of air temperature, the diurnal cycle and seasonal cycle mean that data observed 24 hours and 365 days in the past, respectively, tend to correlate with data observed now.

Because we generally do not know (or cannot observe due to limited deployment durations) the existence of all cycles in the data a priori, we implement only a first-order Markov relationship in the process model to insure that all stations transition from the current observation period (10 minutes, for example) to the next in a consistent manner. This is achieved through the introduction of a parent lag variable for each "true" temperature variable in the spatial model. The lag variables capture the state of the process in the last time step.

The addition of a Markovian lag changes our model from a static Bayesian network into a dynamic Bayesian network. Conceptually, we can now imagine our learned spatial model being repeatedly "stamped-out" over the course of l samples in our database. Each stamp, or layer, contains the learned Bayesian network representing the spatial relationships in addition to a lag variable for each sensor. Figure 6 depicts the temporal model appended to the learned spatial model for the FishNet deployment.

As mentioned in Section 5.1, the spatial component returned by our hill-climbing search is not a unique representation of the set of conditional independencies between the temperature variables; it belongs to a Markov Equivalence Class. However, once we append our temporal model to each network in the MEC, the resultant models may no longer belong to the same class. To choose the best candidate network for the spatial model, we generate all members of this set by using an approach described by Andersson et al. [1995]. Given a directed acyclic graph (DAG), their algorithm returns an *essential graph* that represents the equivalence class and has directed or undirected edges in place of all edges in the input graph. Undirected edges represent relationships between variables that can be reversed (parent becomes the child and visa versa) without changing the overall set of conditional independence relationships. We input our learned spatial model to create the essential graph representing its MEC. We consider all permutations of orientations for the undirected edges in the essential graph such that no cycles are introduced. For each DAG generated in this fashion, we append a set of lag variables, fit the parameters of the combined model as described in Section 5.4, and score the likelihood of the training data given this new network. We choose the spatial model that yields the highest data likelihood when combined with the temporal component.

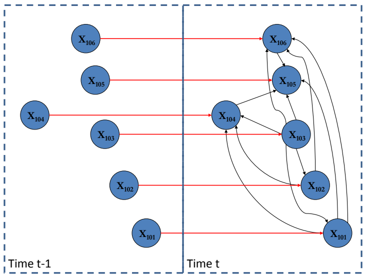


Fig. 6. Time slices are designated by the dashed rectangles. Lag variables are appended for each sensor in the deployment, representing the state of the process in the last time slice.

The first-order Markovian assumption means that we need only consider the state of the process in the previous time step and observations in the current time slice when inferring the posterior distribution over the current state of the process. For example, to compute the posterior of X_{104} at time t from the DBN in Figure 6, we need only the distribution over the previous time step and any observations in the current time step

$$P\left(X_{104}^{t}|X_{101}^{1},X_{101}^{2},...,X_{101}^{t-1},X_{102}^{1},X_{102}^{2},...,X_{102}^{t-1},...,X_{106}^{1},X_{106}^{2},...,X_{106}^{t-1}\right) \quad (14)$$

$$= P\left(X_{104}^{t}|X_{101}^{t-1},X_{102}^{t-1},X_{103}^{t-1},X_{104}^{t-1},X_{105}^{t-1},X_{106}^{t-1}\right) \quad (15)$$

Each variable in our original network now has one additional parent variable and thus one additional parameter (weight associated with the new parent) to estimate. We can still apply our MLE technique for estimating the new parameter values; however, the training set for this model is now a subset of the training set used for our spatial model. This is because we now must place the additional constraint on our training data that it consists of contiguous pairs (two consecutive 10-minute intervals) where all sensor observations are present. We discuss how we respect this constraint in our Experiments and Methodology section.

5.3 Incorporating the Sensor Model

The combined spatial and temporal model represents the transition dynamics of the process over time, as well the correlative structure between the different sensor stations within a deployment. However, we cannot track the progression of the process without external observations; to this end, we incorporate a sensor model that represents the state of the sensor at each time slice and the observation recorded at that station. We represent the sensor state with a discrete variable, S_i, for SensorScope station i that can assume one of two values $S_i \in \{working, broken\}$. The sensor observation is represented as another Normally distributed variable conditioned on the state of the sensor and the current estimate of the air temperature as given by our process model. We will denote the observed variables as O_i where i refers to sensor i within the deployment. Figure 7 represents an abstract visualization of the combined spatial, temporal, and sensor models.

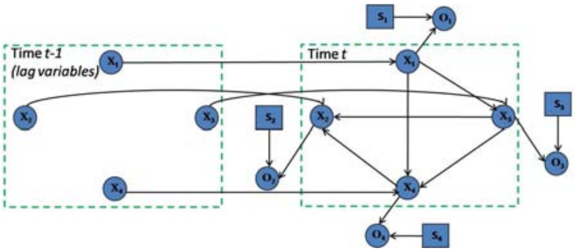


Fig. 7. Time slices are designated by the dashed rectangles. The variables within the dashed area an abstract representation of the learned spatial model among four sensor variables. A sensor state variable and an observation variable are attached to each of the four sensor variables in the current time slice.

We manually set the parameters of the sensor state variables and observation variables. Again, this manual tuning is necessary because there are no known labels for the sensor states in any of the SensorScope datasets. For each S_i, we set the $P(S_i = working) = P(S_i = broken) = .5$ and for O_i

$$P\left(O_i | S_i = working, X_i = x_i\right) \sim N\left(x_i, 0.1\right) \text{ and} \quad (16)$$

$$P\left(O_i | S_i = broken, X_i = x_i\right) \sim N\left(.0001 x_i, 10000.0\right). \quad (17)$$

This parameterization stems from the idea that the sensor state must be able to "explain away" the discrepancy between the observation variable, O_i, and the current estimate of the true air temperature, X_i. That is, if the sensor is believed to be working, then the observation value should be equal to that of the process model's estimate with some small, additional variance (0.1 ℃); contrarily, if the sensor is believed to be broken, then the observation has little do with the actual process and so is much noisier (10000.0 ℃ variance). The 0-mean, large variance distribution of the *broken* state approximates a uniform distribution over the possible range of observed sensor values. If we had ground-truth labels for the sensor state in each observation, explicitly modeling each fault with a separate distribution would not help us identify new anomaly types not seen in the training data. However, we could estimate $P(S_i = working)$ as the ratio of the number of working sensor observations over the total number of observations (and $P(S_i = broken) = 1 - P(S_i = working)$).

5.4 Parameter Estimation

Recall that under the assumption of a linear-Gaussian model, a Normally distributed variable, $X \sim N(\mu_x, \sigma_x^2)$, conditioned on a Normally distributed parent, $Y \sim N(\mu_y, \sigma_y^2)$, has the following density function (assuming both are univariate):

$$P\left(X|Y\right) = \frac{1}{\sigma_x \sqrt{2\pi}} \exp - \frac{\left(x - \left(w_1 y + \mu_x\right)\right)^2}{2\sigma_x^2} \quad (18)$$

That is, $P(X|Y) \sim N(\mu_x + w_1 y, \sigma_x^2)$, where w_1 is a scalar weight multiplied with an input, y, drawn from Y's distribution.

Once our structure learning algorithm has provided each variable in our domain with a set of parents (including the temporal lag variables), the MLE approach to estimating the values of the parameters (μ_i, σ_i^2, and w_i) reduces to solving a multiple linear regression problem [Russell and Norvig 2003]. Specifically, we solve

$$\hat{\theta} = \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \vec{Y}, \quad (19)$$

where $\hat{\theta}$ represents the mean and associated weights of the target variable (the variable whose parameters we are currently estimating), $\mathbf{X}$ is a matrix containing the value of the parents of the target variable in the data set across all samples, and $\vec{Y}$ is a vector containing all the values of the target variable corresponding to the inputs in $\mathbf{X}$.

5.5 Inference

Inference is performed in our models using the Variable Elimination (VE, [Dechter 1996]) algorithm adapted for Conditional Linear Gaussian models [Lauritzen 1992; Lauritzen and Wermuth 1989; Murphy 1998]. There are two inference queries made at each time step, t.

Table I. Procedure for Constructing Full QC Model

(1) Begin with initial Bayesian Network structure, B_{init}, for the input data, D. If no initial network is provided, B_{init} is a network containing no arcs and each variable $x_i \in B_{init} \sim N(0, 1.0)$

(2) Compute the sample mean and covariance of D, $\bar{X}_l$ and S_l.

(3) Compute the mean and covariance represented by B_{init}: $\vec{\mu}_0$ and Σ_0.

(4) Compute T_0 and T_M using the values from steps (2) and (3) in equations (5) and (4).

(5) Perform hill-climbing (Algorithm 1) initialized from B_{init}. Call the resultant structure B_{post}.

(6) Build the Markov Equivalence Set, $\{B_k | B_k \in \text{MEC}(B_{post})\}$ and append the temporal model to each network B_k.

(7) Compute MLE parameters for each $B_k \in \text{MEC}(B_{post})$ from the data, D.

(8) Compute $B_{best} = \arg\max_{B_k} P(D|B_k)$.

(9) Append Sensor State (S_i) and Observation variables (O_i) for each sensor variable (X_i) in B_{best}. The parameters of these variables are manually set.

First, we wish to compute the maximum a posteriori (MAP) assignment of the discrete sensor variables, $\vec{S}^t$, given the set of sensor observations, $\vec{O}^t$,

$$P\left(S_1^t, S_2^t, ..., S_n^t | O_1^t = o_1^t, O_2^t = o_2^t, ..., O_n^t = o_n^t\right). \tag{20}$$

This requires marginalization of the the hidden "true" temperatures (continuous variables) at time t and $t-1$. The remaining sensor-state variables (discrete) are contained in a single potential whose distribution is represented by a table having an exponential number of entries. Each entry corresponds to one of the 2^n possible configurations of n sensor-state variables; consequently, construction of this table occurs in time exponential with the sensor count. The sensor counts in the deployments discussed herein were not prohibitively large; however, for deployments containing more sensors, we could consider approximate inference algorithms, such as Gibbs Sampling [Geman and Geman 1984] or other particle filter methods. These algorithms approach the exact solution as the number of samples or particles used increases. While each sample can be generated in linear time, the number of samples required to reasonably approximate the true joint posterior may be exponential in the number of sensors. Alternatively, we could impose a prior on our spatial structures that would encourage learning disjoint spatial models (i.e. spatial models where one or more of the X_i variables is disconnected from the remainder). In this case, exact inference would be exponential in the number of sensors in the largest subgraph.

Second, we treat the MAP assignment as new evidence for the sensor states at time t and compute the updated estimate of the hidden "true" temperatures, $\vec{X}^t$,

$$P\left(X_1^t, ..., X_n^t | S_1^t = s_1^t, ..., S_n^t = s_n^t, O_1^t = o_1^t, ..., O_n^t = o_n^t\right). \tag{21}$$

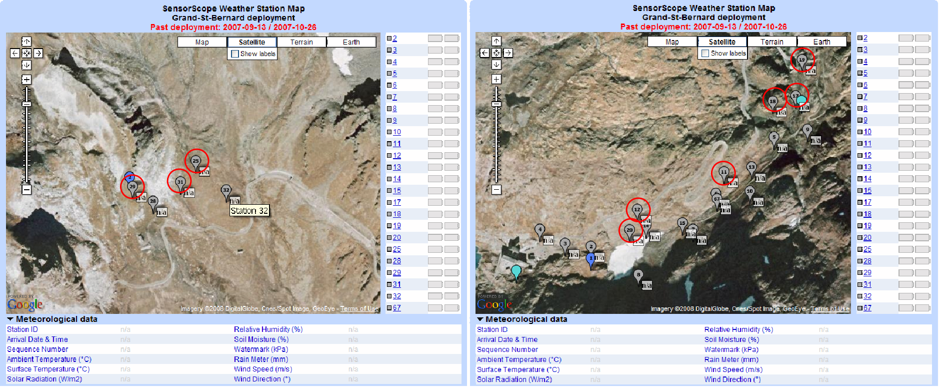
Because we now observe the sensor states, computing the posterior over the true temperatures becomes a query over a linear-Gaussian model. Variable Elimination takes cubic time in the number of sensors for this query and so is tractable to perform exactly. The posterior distribution on the true temperatures is passed forward as a message to be used in inference at time $t+1$. The joint posterior distribution over the true temperature variables can be thought of as an α message in the forward pass of a filtering algorithm [Rabiner 1990]. If the MAP estimate of the sensors at time t indicates that sensor i is working ($S_i = working$), then we input its corresponding observation (O_i) for the true temperature's lag variable at time $t+1$; otherwise, we use the corresponding α message to specify a distribution over the lag's value. We then repeat this two-step query procedure for time $t+1$.

Our motivation for handling inference in this two-step process is that, in an online setting, we must make a decision that each sensor at time t is working or broken rather than postponing this decision and maintaining a "belief state," that is, 79% working and 21% broken. Not only is this approximation useful for an online QC system, it also exempts us from having to maintain an exact belief state that increases in size after each time step. To clarify, the exact belief state at time t would be a 2^{nt} component mixture of n-dimensional multivariate Gaussians. Once we have determined the state of each sensor, we need to propagate forward an α message regarding the true temperatures $\vec{X}$ to time $t+1$ (computed in (21)). Thus, our approximation is made by considering only the mixture component corresponding to the MAP of the S_i variables at each time step.

6. EXPERIMENTS AND METHODOLOGY

Our experiments focus on the validation of our learned spatial models across varying deployments and the efficacy of our complete DBN model as a tool for quality control. We address each issue in turn. We perform the former validation through a series of hold-one-out prediction tests to determine the relative strength of multiple stations as a predictor for an individual missing station. Second, we provide a comparative analysis of the performance of three different QC models as applied to real data from the SensorScope project. The three models are the spatial, temporal, and spatiotemporal models already discussed, each augmented with a sensor model as described in Section 5.3. The experiments reflect the weaknesses and strengths of each model, and show preliminary justification for pursuing a spatiotemporal approach. Lastly, we evaluate the performance of our model in terms of type I and type II error rates. This experiment is performed via the addition of artificial noise to the original datasets in order to create labels that can be matched against our predictions of the sensor variable.

All experiments in this section were performed with a data set spanning from the beginning of the respective deployment to its end. Because the SensorScope stations are not necessarily synchronized to sample at the same time, the data was binned and averaged into 10-minute windows consistent across all stations. A training set and testing set were created for each deployment by roughly splitting the data into halves, in which the first half (representing the first chronological half) became the training data and the second half became the test set. In all experiments, only training samples (10 minute windows) where readings for all of the stations were present were used, and so often the training sets are significantly smaller than the testing sets. For experiments containing a temporal model, only those training samples that had a fully observed preceding sample (the last 10-minute period) were used. Data for the experiments comes from the FishNet and the Grand St. Bernard deployments. Grand St. Bernard was a third deployment located in the Grand St. Bernard Pass between Switzerland and Italy (at an elevation of 2300 meters) and was in place from September 13, 2007 to October 26, 2007. All six sensors were used in the FishNet deployment; however, only a subset comprising 9 of the 23 stations were used from the Grand St. Bernard (see Figure 8). Because we are only including training samples where all sensor measurements are present, including all stations from the Grand St. Bernard would exclude too many potential samples.



SensorScope Weather Station Map
Grand-St-Bernard deployment
Past deployment: 2007-09-13 / 2007-10-26

▾ Meteorological data

Station ID	n/a	Relative Humidity (%)	n/a
Arrival Date & Time	n/a	Soil Moisture (%)	n/a
Sequence Number	n/a	Watermark (kPa)	n/a
Ambient Temperature (°C)	n/a	Rain Meter (mm)	n/a
Surface Temperature (°C)	n/a	Wind Speed (m/s)	n/a
Solar Radiation (W/m2)	n/a	Wind Direction (°)	n/a

SensorScope Weather Station Map
Grand-St-Bernard deployment
Past deployment: 2007-09-13 / 2007-10-26

▾ Meteorological data

Station ID	n/a	Relative Humidity (%)	n/a
Arrival Date & Time	n/a	Soil Moisture (%)	n/a
Sequence Number	n/a	Watermark (kPa)	n/a
Ambient Temperature (°C)	n/a	Rain Meter (mm)	n/a
Surface Temperature (°C)	n/a	Wind Speed (m/s)	n/a
Solar Radiation (W/m2)	n/a	Wind Direction (°)	n/a

Fig. 8. Left: The portion of the Grand St. Bernard deployment on the Italian side of the mountain pass. Right: The Swiss side of the Grand St. Bernard deployment, located approximately 2 kilometers east of the Italian deployment. The stations circled in red denote those stations chosen for purposes of modeling.

6.1 Leave-One-Out Prediction

The leave-one-out experiments are performed by withholding a sensor's observation and computing the posterior distribution over the hidden sensor value given all other sensor observations and the learned spatial (Section 5.1) and spatiotemporal models (Section 5.2). We report results for the FishNet and Grand St. Bernard SensorScope deployments. In both cases, the spatial model is learned and parameterized using only the first half of the data (approximately 1400 and 440 training samples, respectively).

Once the spatial and spatiotemporal models are trained, we process the testing data (second half of the collected samples) in an iterative manner. In each iteration, a single observation, representing the measurement taken at one station at one time point, is removed. We compute a posterior prediction for the removed value using the learned spatial model and the observations from all other stations in the case of the spatial model, and all other stations in addition to all measurements from the previous time step in the case of the spatiotemporal model. We compute the mean squared error (MSE) between the predicted value for the withheld observation and its actual value in the test set, as well as the variance in our prediction. Let

$t = 1, ..., T$ denote the time (sample) index, $i = 1, ..., n$ index the "true" temperature variable X_i, and x_i^t be the value of the true temperature at station X_i at time t. The MSE and Variance for station X_i is then computed as

$$MSE_i \; = \; \frac{1}{T} \sum_{t=1}^{T} \left(E\left[P(X_i | \mathbf{X} \backslash X_i) \right] - x_i^t \right)^2. \tag{22}$$

The variance of the posterior estimate of X_i depends only on the set of variables that are observed (included in the set $\mathbf{X} \backslash X_i$); not on the exact value of those observations. Thus, we need only examine $Var\left[P(X_i | \mathbf{X} \backslash X_i) \right]$ for any one of the t samples above to determine the variance. The leave-one-out error is also measured using cumulative log likelihood (CLL),

$$CLL_i \; = \; \sum_{t=1}^{T} \log P(X_i = x_i^t | \mathbf{X} \backslash X_i), \tag{23}$$

and is shown as the dashed horizontal line in Figures 9, 10, and 11. We then perform a further computation, removing an additional variable's observation from the testing data. We compute the cumulative error over the training data (sum log likelihood) in predicting our original target variable with one additional sensor's observation missing. Using the same notation as above, we compute this as

$$CLL_{i,j} \; = \; \sum_{t=1}^{T} \log P(X_i = x_i^t | \mathbf{X} \backslash \left\{ X_i, X_j \right\}). \tag{24}$$

Each bar in Figures 9, 10, and 11 corresponds to the new CLL value after the variable X_j has been hidden. The purpose of removing a second variable X_j is to measure the contribution of second variable in predicting the value of the first removed variable X_i.

Figure 9 (upper left plot) indicates that station 101 was not only the most difficult to predict (MSE of .56 ℃), but also gained the least from the presence of other sensors. Additionally, removing the observations of station 104 resulted in the largest increase in error for station 101; however, even this effect was not particularly significant in comparison to removing any of the other remaining stations. The likely reason for this lack of correlation is due to station 101's position on the south edge of the deployment (Figure 5), near the wooded border. Station 104, its most similar station, is also located in close proximity to a wooded, shady region, which may explain its role as the strongest predictor for station 101. This example highlights the fact that our "spatial" learning is discovering more correlation than those just based on spatial proximity as we might see in a Kriging model [Matheron 1963]. Rather, our model is capturing all sources of linear correlation between sensors at a given time step, without the use of a feature set describing each sensor.

Stations 105 and 106 (bottom center and bottom right plots, respectively) appear to be very highly correlated, as indicated by the dramatic increase in prediction error when either station is held out while predicting the other. Moreover, we see that when holding out each station (105 and 106), there is little error in reproducing the withheld observation given the presence of the other 5 sensors (MSEs of .058 ℃ and .074 ℃, respectively). The Sensirion SHT75 documentation reports a measur-

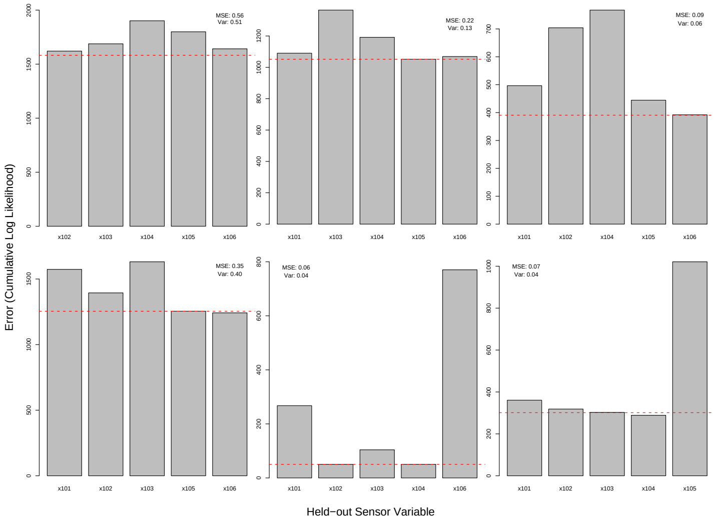


Fig. 9. Redundancy Test for FishNet Spatial Model. Dashed line indicates the error in predicting the individual missing sensor. Each bar along the X axis represents the change in error from removing the additional sensor variable corresponding to that bar. The Y axis is the error measured as the cumulative log likelihood over all test cases of the true value given the predicted distribution.

ing accuracy of ±.35 ℃ in operating conditions of 15.21 ℃ (average temperature of the FishNet site for the testing period).

Figure 10 conveys a similar analysis performed with nine stations selected from the Grand St. Bernard deployment (Figure 8). The top row of bar plots depicts the analysis of stations 11, 12, and 17, while the bottom row corresponds to stations 25, 29, and 31. It is apparent from the plot that stations 17 and 29 are the most difficult to predict from the remaining 8 sensors. This stands to reason for station 29, for though it is located on the Italian side of the deployment with 25 and 31, it is still separated by a steep hillside dividing the region. We could not ultimately discern the reason for station 17's discordant behavior from the remaining sites. The Sensirion SHT75 documentation reports a measuring accuracy of ±1.0 ℃ in operating conditions of 1.83 ℃ (average temperature of the Grand St. Bernard site for the testing period).

It is interesting to note in Figure 10 that, in all cases, there exists at least one sensor whose removal actually seems to decrease the amount of error in predicting the hold-out sensor's value. This trend suggests that our learned model may have overfit the original training data, and thus poorly generalized to the test set. In the case of air temperature data measured over 1-2 months (especially during seasonal transitions), data monitored at the beginning of the observation period can differ significantly from data measured toward the end of the observation period. This compounds the difficulty of our work, as now our underlying assumption of a single generative multivariate distribution creating our training and testing data is no longer valid. Future work will need to focus on time-series analysis techniques that can map the test set to our training set without full knowledge of the trend effects

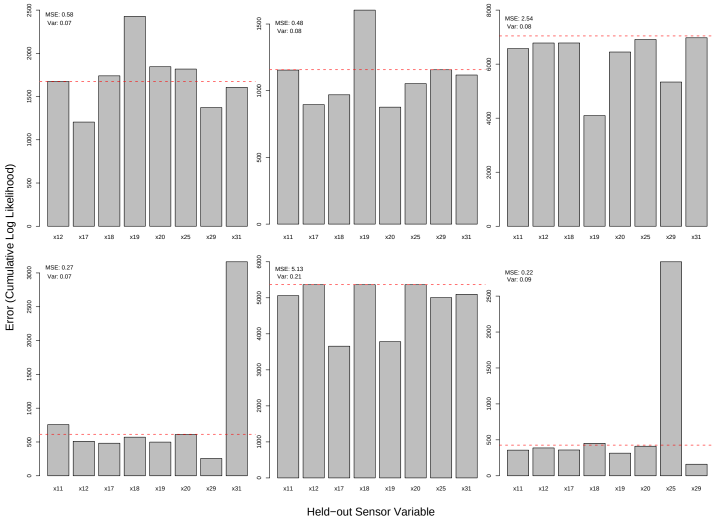


Fig. 10. Redundancy Test for Grand St. Bernard Spatial Model. Dashed line indicates the error in predicting the individual missing sensor. Each bar along the X-axis represents the change in error from removing the additional sensor variable corresponding to that bar. The Y-axis is the error measured as the cumulative log likelihood over all test cases of the true value given the predicted distribution.

that shape the generative distribution over time.

Finally, Figure 11 shows the hold-one-out analysis applied to a spatiotemporal model learned from the Grand St. Bernard data. Recall that this model is simply the original spatial model with the relevant lag variables appended to its structure, and the parameters reestimated to account for the additional set of parents. We notice that, in all cases, the hold-one-out error decreases with the incorporation of a lag effect. Also significant is that, with the exception of stations 17 and 29, the lag effect has the greatest predictive power for every station. This makes intuitive sense, as air temperature is unlikely to change significantly over the course of 10 minutes (the duration of the lag). Stations 17 and 29 suffer from the same overfitting problem in this revised model, as hiding the Markovian variable reduces error in both cases. In fact, the large magnitude of the gain incurred from hiding the lag variable from station 17 seems to support the theory that the nature of the correlative effect changed drastically between the training and testing periods. If it had not, then it is unlikely that the spatiotemporal model would have lent the lag variable such significant weight based on the training data.

In addition to providing some intuition about the values of parameters and network structures learned in the spatial component of our QC system, this type of hold-one-out analysis can be used to identify redundant sensors. For purposes of quality control, two sensors measuring the same phenomenon (or one able to near-perfectly predict the other's missing value) is necessary to truly validate recorded observations; however, for purposes of capturing all the heterogeneity encompassed within a site, it may be preferable to relocate any sensor considered redundant. This analysis can be easily generalized to hold-two-out in order to detect clusters

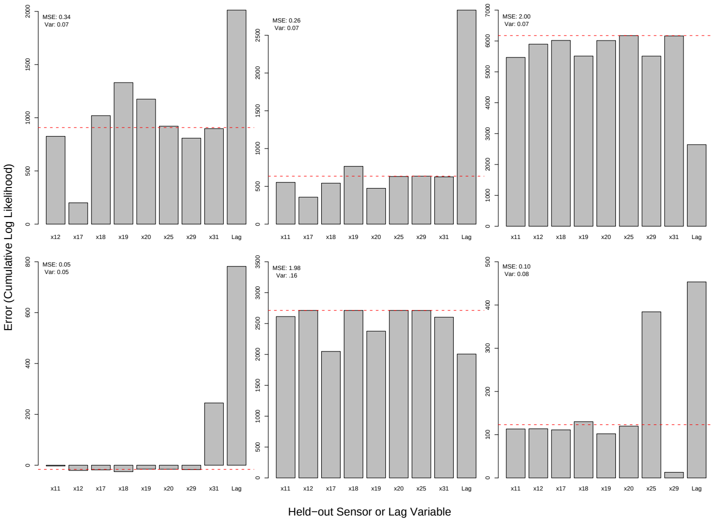


Fig. 11. Redundancy Test for Grand St. Bernard Spatiotemporal Model. Dashed line indicates the error in predicting the individual missing sensor. Each bar along the X-axis represents the change in error from removing the additional sensor or lag variable corresponding to that bar. The Y-axis is the error measured as the cumulative log likelihood over all test cases of the true value given the predicted distribution.

of 3 sensors where one sensor can accurately predict the value of the other missing two (redundancy at this level would even be unnecessary for QC purposes).

6.2 Quality Control Experiments

We begin this section by providing a comparative analysis of the spatial-only and temporal-only models. Every model type discussed here contains the sensor state model, as described in Section 5.3, appended to the structure. That is, the spatial-only model is a learned network structure over the first half of the data collected from the deployment with a discrete sensor-state variable and a continuous, Normally distributed observation variable added for each sensor. The temporal-only model assumes independence between all stations, but contains an additional lag variable for each sensor and is auto correlated with that lag. Figure 12 demonstrates the performance of the spatial model as applied to the second half (testing set) of the Grand St. Bernard deployment data.

The spatial-only model is able to recognize the spiky, anomalous behavior observed in both stations 17 and 29 between days 21 and 25. Moreover, this QC system detects the flatline anomaly when station 17's air temperature sensor reported 0-voltage, which is by default converted to a reading of −1 °C. The dashed line represents the system's prediction of the actual temperature value and appears to be consistent with the neighboring stations of sensor 17. Unfortunately, the lack of a temporal connection means that this model's behavior is static over time. The overall mean of each station remains constant, because there is no transition function to allow the mean of the process model to track the true temperature, nor is there any explicit conditioning on the seasonality or index in the diurnal cycle. The

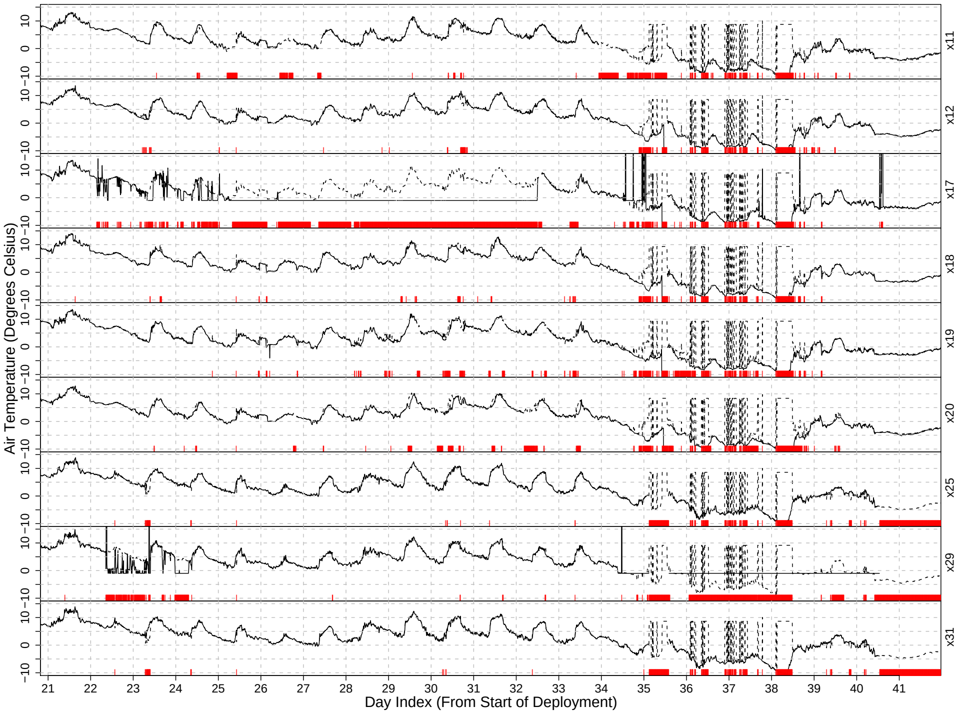


Fig. 12. Quality Control performance on Grand St. Bernard using the spatial-only model. Solid line indicates the actual temperature recorded at each station. Dashed line indicates the posterior prediction made for that station. Red hashes at the base of each graph indicate a value labeled as anomalous (i.e., $S_i = broken$) by our system for that time period. The X-axis denotes the day (vertical dashed line depicts midnight) since the deployment began, and the Y-axis denotes temperature in degrees. Corresponding station names appear on the right side of the graph next to the stream they depict.

end result is that, as the mean of the true temperature begins to decrease to the point where it significantly differs from the learned mean in the training data, the model labels these new values as anomalous. This begins to manifest itself at day 35. Each time the average reading from all 9 sensors drops significantly below the training data mean, an anomaly is raised and the model imputes the training data mean as the correct value. The large disparity between the model's prediction and the actual observations between days 35 and 39 results in most of the observations therein being misclassified as anomalous, save for daytime high values.

Figure 13 shows the performance of the temporal-only model on the same Grand St. Bernard data. This model is equivalent to n disjoint Kalman Filter models [Kalman 1960], with an additional discrete sensor-state variable that explains away any discrepancy between the observation and predicted value of the air temperature. Of immediate note is that the -1 ℃ flatline in station 17 is no longer properly flagged as anomalous. A few nominal observations near -1 ℃ beginning on day 25 confuse the temporal model into tracking this flatline behavior. If the transition between temperature observations over time is gradual enough, the temporal-only

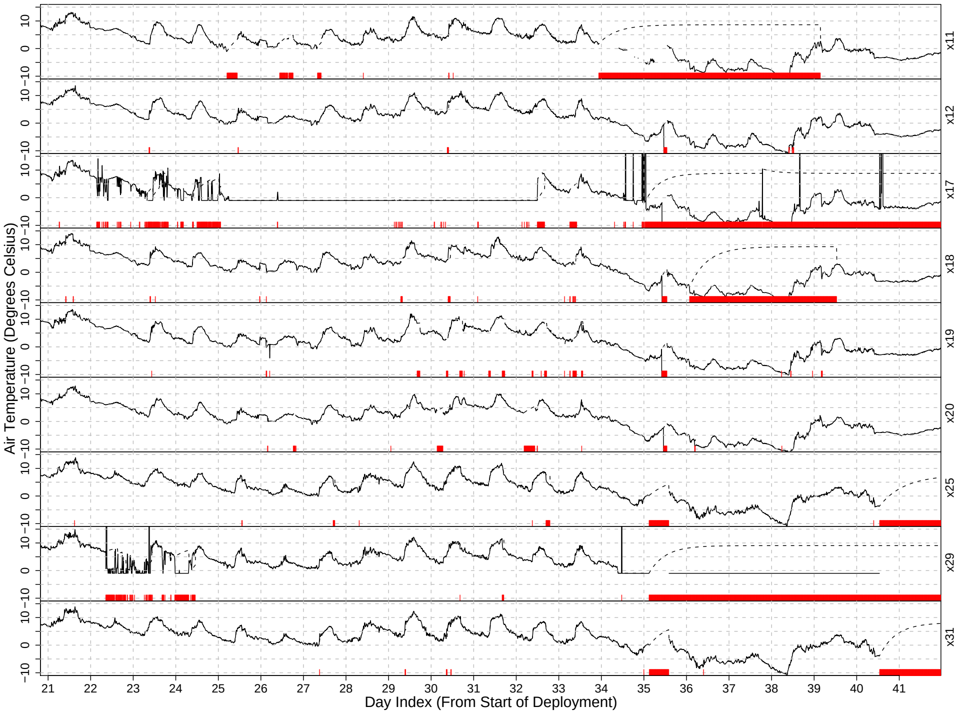


Fig. 13. Quality Control performance on Grand St. Bernard using the temporal-only model. Solid line indicates the actual temperature recorded at each station. Dashed line indicates the posterior prediction made for that station. Red hashes at the base of each graph indicate a value labeled as anomalous (i.e., $S_i = broken$) by our system for that time period. The X-axis denotes the day (vertical dashed line depicts midnight) since the deployment began, and the Y-axis denotes temperature in degrees. Corresponding station names appear on the right side of the graph next to the stream they depict.

model will track the temperature signal through periods of anomalous readings caused by sensor malfunction. Without external observations from correlated stations, the independent sensor cannot differentiate between slow changes in the observations due to a change in the process signal (warming or cooling trends) or the breakdown of the sensor. In cases where the observed value disagrees with the model's predicted value (the model loses tracking), future predictions drift toward the training data mean. This can be seen at station 11 on day 34 when the signal is completely lost, or station 17 on day 35 when an erratic spike followed by a drop in temperature throws off the model.

The temporal model's ability to track the process even as it drifts away (albeit slowly) from the trained mean gives it an advantage over the spatial-only model. We can see this in stations 12, 18, 20, 25, and 31, where a slow cooling effect does not disrupt the model's ability to track the process during the second half of the training period. Unfortunately, the assumption of complete independence between stations means that the model cannot accurately reconstruct the true value of the temperature at stations diagnosed as *broken*, as seen in stations 11, 25, 29, and 31.

To this end, we turn to the spatiotemporal model.

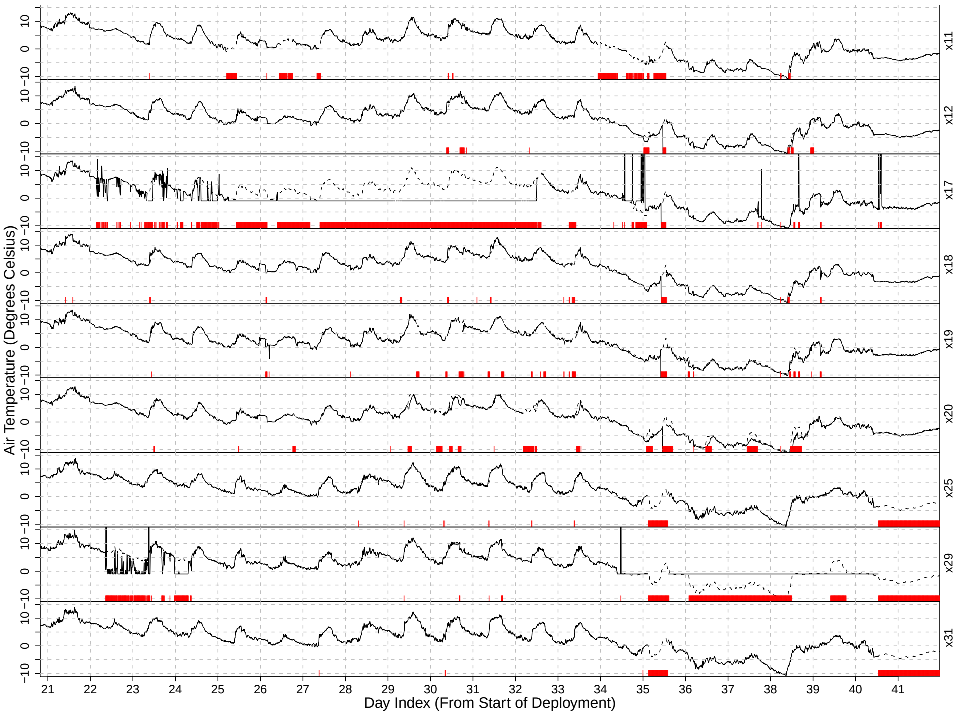


Fig. 14. Quality Control performance on Grand St. Bernard using the spatiotemporal model. Solid line indicates the actual temperature recorded at each station. Dashed line indicates the posterior prediction made for that station. Red hashes at the base of each graph indicate a value labeled as anomalous (i.e., $S_i = broken$) by our system for that time period. The X-axis denotes the day (vertical dashed line depicts midnight) since the deployment began, and the Y-axis denotes temperature in degrees. Corresponding station names appear on the right side of the graph next to the stream they depict.

The performance of the spatiotemporal model (Figure 14) appears robust to the weaknesses in the spatial-only and temporal-only models. In particular, it is able to both detect and reconstruct the anomalous values from flatlined senors (station 17, days 25 to 33) and missing values (station 29, days 35 to 42). Further, the model permits some drift in the original learned distribution of the process model, as indicated by its accurate tracking of the air temperature from days 35 through 39, with few apparent false positives. Like the spatial and temporal models, the combined model is able to diagnose the obvious spikes in air temperature that are also indicative of sensor malfunctions (station 17, days 35 to 41). The overall false positive rate seems minimal (save for the midday periods on days 36-38 at station 20, where the predicted estimates are slightly higher); however, without ground-truth data, we cannot determine the true type I and II error rates.

Unfortunately, in cases where GPRS is lost (either due to a required reboot, or a failure at the station) and all station signals are lost, our spatiotemporal model

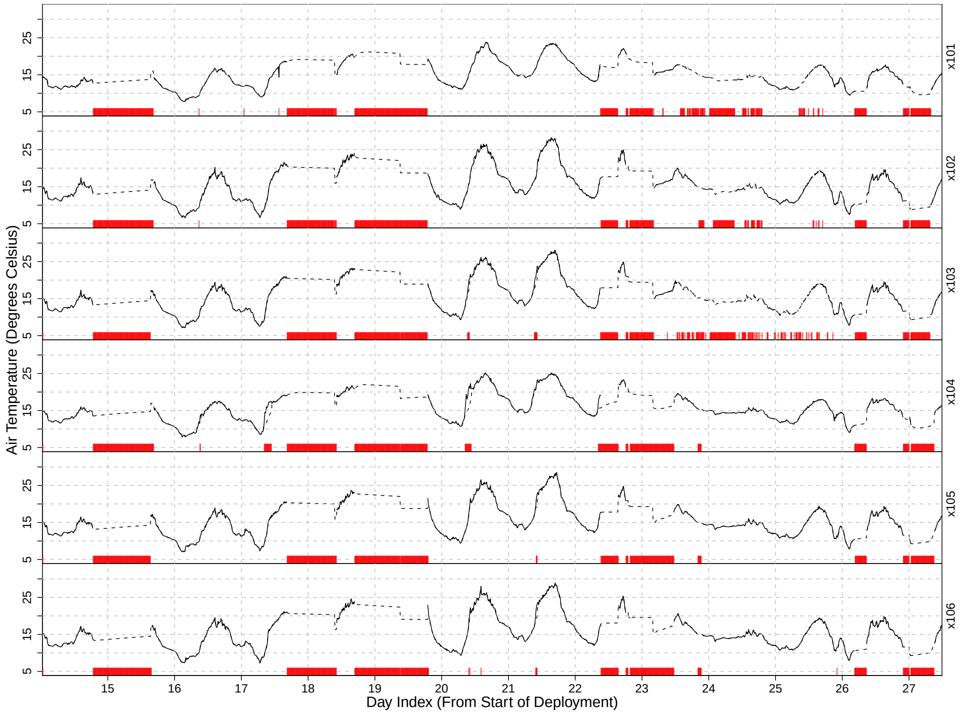


Fig. 15. Quality Control performance on FishNet using the spatiotemporal model. Solid line indicates the actual temperature recorded at each station. Dashed line indicates the posterior prediction made for that station. Red hashes at the base of each graph indicate a value labeled as anomalous (i.e., $S_i = broken$ by our system for that time period. The X-axis denotes the day (vertical dashed line depicts midnight) since the deployment began, and the Y-axis denotes temperature in degrees. Corresponding station names appear on the right side of the graph next to the stream they depict.

cannot recreate the missing values. This is apparent in the FishNet dataset, where GPRS outages were relatively frequent (Figure 15) compared to Grand St. Bernard. In cases where no sensors are providing observations, the variance over the joint posterior of the current process state grows. This is because a lack of evidence about the state of the process means that we are becoming increasingly uncertain about its current value. Eventually, the variance will grow so large that almost any observed value will seem likely, and so our process will begin to retrack the observations. When the sensor readings resume, the very first observation is recognized as nonanomalous and causes the spatial and temporal components to shift the process model to that observed value for all correlated sensors. For example, during the outage beginning on day 18.5 and lasting until day 19.75, there is a single observation around day 19.5 that causes the collective temperature prediction for most stations to shift down 4 ℃. Otherwise, during these periods of prolonged GPRS outage, the prediction will drift toward the mean for each station while variance on the prediction grows larger.

Without knowledge of neighboring observations within a site or a sufficiently high-order Markovian model, there is little the current model can do to correctly track the air temperature during periods in which all sensors fail to report. One potential solution would be to introduce a baseline calculation that represents prior knowledge about the air temperature at a given site for each time of day and time of year. In the absence of evidence to correct this baseline estimate, the model would default to the baseline values. While the baseline may be inaccurate given the temporal context (warmer/cooler than usual, storm effect, etc.), it would likely guide the process model such that it was closer to the actual signal when observations recommenced. Formulation of this baseline and its performance in a long-term stationary QC domain is reported in Dereszynski and Dietterich [2007]. The problem remains that, in the short-term ecological monitoring setting, it may be difficult to estimate this baseline in the absence of a full cycle of the observed phenomenon (one year in the case of air temperature).

6.3 Noise Injection Experiments

To obtain a quantitative assessment of the accuracy of our quality control methods, we performed a series of experiments in which we injected noise into the SensorScope data. Initially, all readings are assigned to be nonanomalous. Then any missing values or 10-minute average of exactly -1.0 degrees ℃ are labeled as anomalous. Finally, synthetic faults are introduced by taking each data point and, with probability η, adding a noise value drawn from a normal distribution with zero mean and variance σ_n^2. Each synthetic fault is labeled as anomalous.

We report the results of the noise injection experiments for both the FishNet and Grand St. Bernard deployments. The experiments were performed using values of σ_n^2 ranging from 3 ℃ to 30 ℃ in increments of 3 ℃ and values of η ranging from 5 to 50 percent in increments of 5 percent. Thus, we evaluate 100 different variations on the testing data for each dataset, and record the results in terms of the number of true positives (TP, number of anomalous data values classified by our system as such), true negatives (TN, the amount of clean values that were not flagged by our system as anomalous), false positives (FP, misclassified anomalies that were actually clean), and false negatives (FN, values that were actually anomalous, but not detected by our system). The results are summarized in terms of Cohen's κ statistic (the rate of agreement between our classifier and the true labels correcting for chance agreement [Cohen 1960]), precision (the total number of true positives divided by the number of true positives plus false positives), and recall (the total number of true positives divided by the number of true positives plus false negatives). Cohen's κ reflects the degree to which our algorithm reproduces the true labels as created by our noise injection process corrected for chance predictions [Cohen 1960; Viera and Garrett 2005]. It is calculated as

$$\kappa = \frac{P(O) - P(E)}{1.0 - P(E)}, \tag{25}$$

$$P(O) = \frac{TP + TN}{FN}, \tag{26}$$

$$P(E) = \frac{TN + FP}{N} \times \frac{TN + FN}{N} + \frac{FP + TP}{N} \times \frac{FN + TP}{N}, \tag{27}$$

where $P(O)$ is the observed probability of the classifier agreeing with the true label,

$P(E)$ is the expected probability of chance or coincidental agreement, and N is the total number of samples in our testing data. Regarding the latter two statistics, precision provides a sense of how many of the values we label as anomalous are truly indicative of sensor faults, while recall summarizes what percent of the total genuine sensor faults we detect in the data. In this application, we are interested in achieving as much precision as possible at high levels of recall. In other words, we want to make sure we detect most of the sensor faults, even if this leads to some false alarms (false positives).

Let us consider what results we should expect from injecting noise. First, we would expect that as the magnitude of the noise (σ_n^2) increases, the noise will become easier to detect, because it will be clearly distinct from nearby values in time and space. Increasing the amount (η) of noise should not decrease our ability to detect it; however, fewer non-noisy values will make tracking short-term changes in the air temperature (due to storm effects, cold/warm fronts, etc.) more difficult. For this reason, we expect that larger values η will result in more false positives in cases where the model loses tracking and its predictions drift away from the true air temperature. Ultimately, the best data-anomaly detection performance will be obtained when there is a small amount of very obvious noise in the data (small η and large σ_n^2).

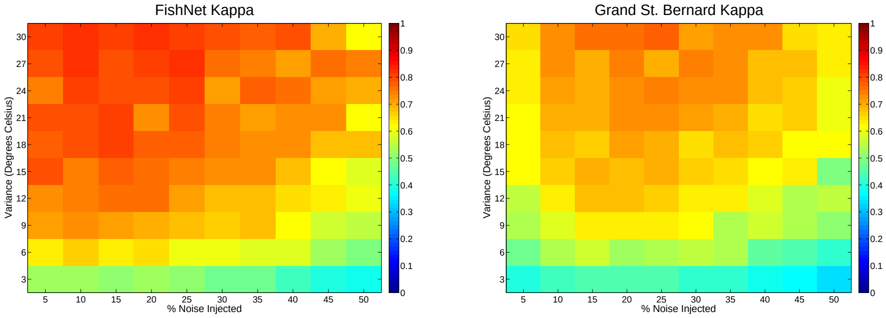


Fig. 16. κ as a function of both noise variance (Y-axis) and percentage of data points modified by noise (X-axis). The shade of color associated with each grid cell reflects the degree of κ (on a scale of 0 to 1.0) for that configuration of noise level and saturation.

Figure 16 shows the κ rates of our model applied to the FishNet (left) and Grand St. Bernard (right) deployments. The value of κ is displayed on a color scale, with higher values shown in darker shades of red and lower values shown in darker shades of blue. As expected, larger values of σ_n^2 resulted in better κ scores for both the FishNet and Grand St. Bernard data sets. Data anomalies drawn from a higher variance distribution are more evident to our classifier; consequently, there is more genuine agreement. In the of case FishNet, κ increases from .527 to .826 as we increase σ_n^2 from 3 to 30, and from .440 to .755 in the case of Grand St. Bernard (at $\eta = 20\%$). Interesting to note is that more noise in the data does not have an adverse effect on our κ scores until more than 25% to 30% noise is introduced. In fact, the κ scores for both data sets increase up to this point. As more anomalies are introduced into the data and correctly identified by our algorithm, the likelihood of coincidental agreement ($(P(E))$) decreases; however, there appears to be a threshold at approximately 25% noise where κ begins to decrease. This suggests a tradeoff where further abundance of anomalies in the data makes them appear haphazard rather than systematic.

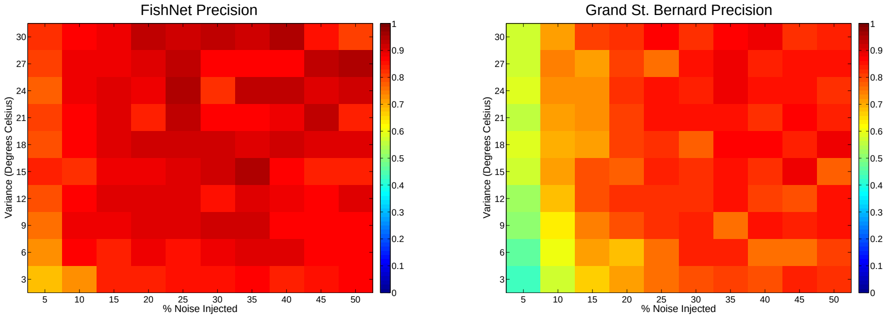


Fig. 17. Precision as a function of both noise variance (Y-axis) and percentage of data points modified by noise (X-axis). The shade of color associated with each grid cell reflects the degree of precision (on a scale of 0% to 100%) for that configuration of noise level and saturation.

Precision results in Figure 17 further support our hypothesis regarding the effect of larger values for σ_n^2. The true anomalies are more disparate from the normal data, and so the ratio of true positives to false positives increases directly with magnitude of the noise (fewer false positives result in higher precision scores). Further, as we increase the amount of noise in the data, any value we classify as anomalous has a higher chance of actually being so due to a greater proportion of the data containing noise. When there are few anomalies in the data (5% injected noise), our scores suffer due to the presence of relatively many false positives. Consider our false positive rates (percent of all “good” data missclassified as anomalous) shown in Figure 18. Though we achieve very low false positive rates at this level of noise (average of 1.01% at FishNet and 3.08% at Grand St. Bernard for $\eta = 5\%$), there are too few synthetic anomalies in the data, causing the false positive counts to dominate the precision scores. In addition, while some of these false positives are “good” values misdiagnosed as anomalies by our system, it is likely that many of these errors come from suspicious values that we did not pre-flag as existing anomalies due to a lack of domain expertise (for example, sensors 17 and 29 of Grand St. Bernard on days 22 through 25 in Figure 12). Thus, even though our model is catching these likely faults, each is being labeled as a false positive.

There are cases where poor performance is caused by multiple sensor streams being affected by noisy values simultaneously. Consider the QC model applied to the noise-injected data in Figure 19 ($\eta = 50$, $\sigma_n^2 = 15$). Beginning on day 36, a cooling trend affects all stations in the deployment and reduces the true air temperature below 0 degrees ℃. Incidentally, station 29 flatlines at exactly -1.0 degrees ℃ during this period. As the true signal dips below -1.0 degrees ℃, we notice that all stations begin predicting values hovering near this boundary until

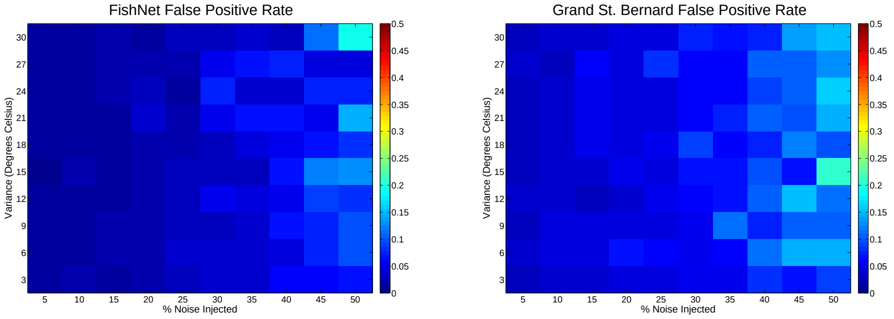


Fig. 18. False Positive rates as a function of both noise variance (Y-axis) and percentage of data points modified by noise (X-axis). The shade of color associated with each grid cell reflects the degree of precision (on a scale of 0% to 50%) for that configuration of noise level and saturation.

the cooling trend ends around day 39. Further, instead of recognizing the values reported from station 29 as anomalous for this period (as in Figure 14), these values are the only ones labeled as nominal. This behavior occurs because the amount and degree of injected noised in the data makes it unlikely that a set of observations at time t will be consistent with the spatial component of the QC model, and makes it even less likely that two sets of contiguous observations (times t and $t + 1$) will be consistent with the temporal transition component. During this period, station 29 behaves very consistently from a temporal perspective (its observations are constant from day 34.5 to 40.5), and the variance of the injected noise is enough to bring the observed signal from the other stations within close proximity of the flatline value. The end result is that station 29 becomes the standard for nominal behavior until the true temperature deviates from the flatline by a margin larger than the magnitude of added noise.

In the above-mentioned scenario, increased values of both η and σ_n^2 negatively affect the precision score. As η increases, the likelihood of all sensors encountering noisy values simultaneously also rises. Noisy observations drawn from a wide-variance distribution are more likely to strongly disagree with the QC model's predictions and, as a result, this frequently causes the model to ignore such observations. This kind of scenario becomes more probable in cases where there are relatively few, highly-correlated sensors that are prone to simultaneous malfunctioning.

Recall values, displayed in Figure 20, are largely invariant to changing values of η. Increasing the amount of noise in the original data has no significant effect on our ability to find all data anomalies. This is unsurprising. As per our discussion of precision, an abundance of anomalous values may introduce tracking problems that leads to misclassifying normal values as anomalous; however, false positive values do not factor into recall scores. An increase in the magnitude of the noise distribution directly benefits our recall scores, which is again consistent with our hypothesis. In both data sets, we are able to achieve greater than .70 recall once the variance of the noisy data reaches 15 °C. For the FishNet deployment, we can simultaneously reach a precision score of .87 while keeping our false positive rate

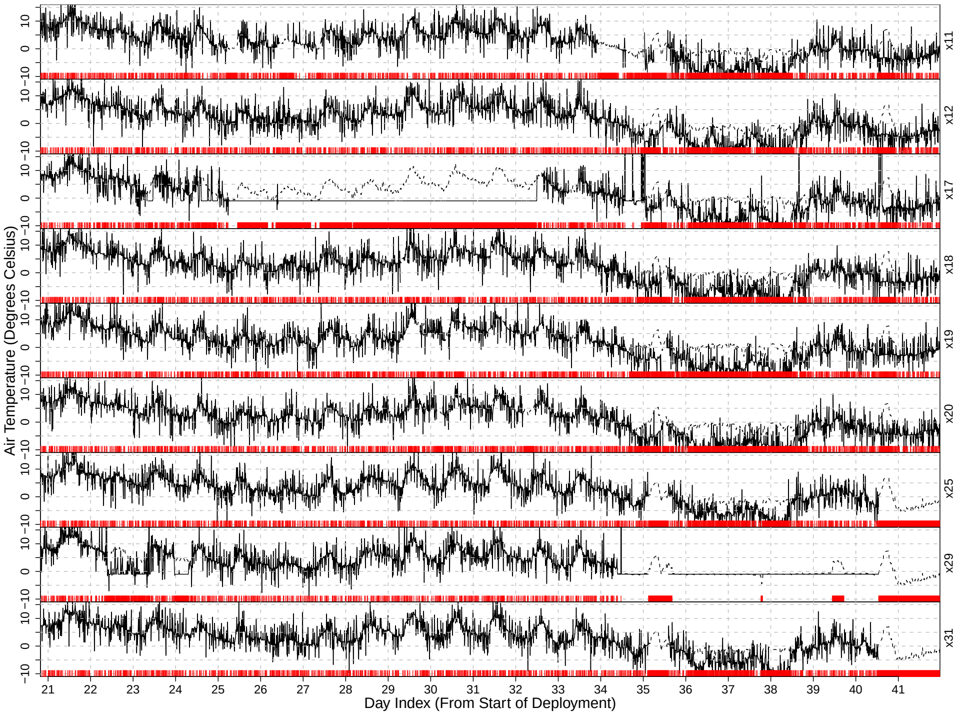


Fig. 19. QC results for Grand St. Bernard data with 50% added Gaussian noise with variance of 15 °C. Red hash marks depict a sensor diagnosis of *broken* for that particular value. Corresponding station names appear on the right side of the graph next to the stream they depict.

at 4.6%. At Grand St. Bernard, we operate at .78 precision with a false positive rate of 7.0%. Again, Grand St. Bernard's worse performance is partially due to the presence of suspicious values that we did not preflag as data anomalies before injecting noise.

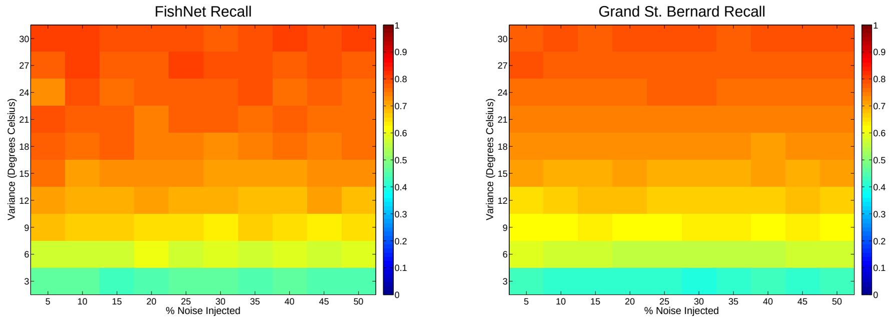


Fig. 20. Recall as a function of both noise variance (Y-axis) and percentage of data points modified by noise (X-axis). The shade of color associated with each grid cell reflects the degree of recall (on a scale of 0% to 100%) for that configuration of noise level and saturation.

Lastly, we provide an individual analysis of the effects of increasing the frequency and magnitude of noise in each of these datasets. Figure 21 shows the average κ, recall, and precision for the FishNet (left) and Grand St. Bernard (right) deployments, as a function of only η (top) and only σ_n^2 (bottom). In the case of the top graphs, each value on the vertical axis represents an averaging over all values for σ_n^2; likewise, each value on the vertical axis for the bottom graphs is an average across all values for η.

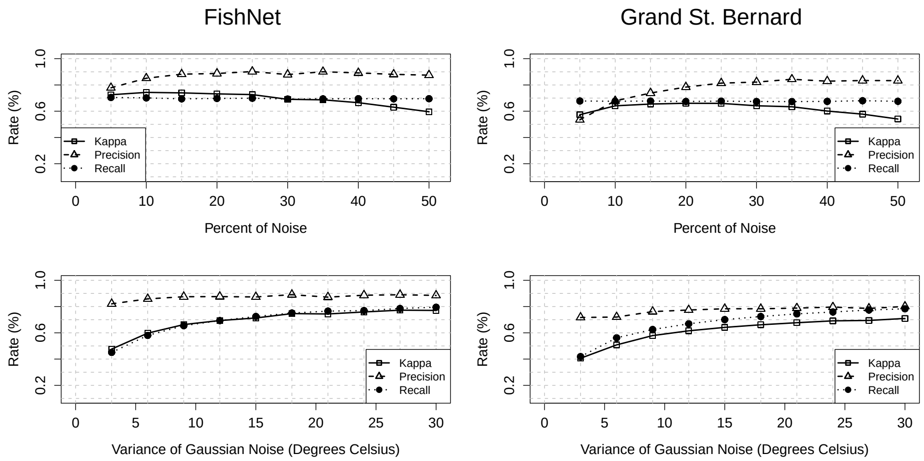


Fig. 21. Left: Precision, Recall, and κ for the FishNet noise injection experiments as function of percentage of noise (top) and degree of variance (bottom). Right: Precision, Recall, and κ for the Grand St. Bernard noise injection experiments as function of percentage of noise (top) and degree of variance (bottom).

Increasing the amount (η) of anomalous values in the data resulted in a increase of precision for both data sets. The effect seems less pronounced for the FishNet data set (.778 to .875) compared to Grand St. Bernard (.535 to .833) as η varies from 5% to 50%. Again, this is likely attributable to the existence of many suspicious values in the Grand St. Bernard dataset that were not preflagged as data anomalies. Prior to any Gaussian noise being added, these values would appear anomalous to our model and would be classified as data anomalies. When η is small, it is unlikely our noise injection process will target these observations and turn them into true cases of data anomalies. The end result is these observations become false positives. As η increases, more of these suspicious values are rightly cast as data anomalies during noise injection. κ increases initially with more noisy values injected into the data; however, as discussed with the κ results, we see diminishing returns and an eventual loss in κ as η grows over 25% for both the FishNet and Grand St. Bernard datasets.

The lower portion of Figure 21 shows a definite gain in overall precision, recall, and κ as the degree of noise in the data (σ_n^2) rises from 3 °C to 30 °C. Specifically, κ increases by approximately .3 in both data sets, recall increases by .34 in FishNet and .36 at Grand St. Bernard, and precision increases by .06 at FishNet and .08 at

Grand St. Bernard. In regards to recall and κ, this is because as the added noise in the data becomes more obvious (higher variance), it becomes easier for our model to detect it. The increase in precision is less pronounced than the increase in recall and κ, which could be caused by our QC model having too a great a sensitivity to anomalous values (too small of a variance for the sensor observation variable when the sensor state is believed to be *working*).

6.4 Noise Injection & Model Comparison

In this section we use our noise injection methodology to validate that learning a spatial model provides superior performance than arbitrarily choosing a spatial structure. We examine four spatiotemporal QC models.

—*Best*: QC model having the highest-scoring (best) spatial structure as returned from the algorithm described in Algorithm 1.

—*Worst*: QC model having the lowest-scoring (worst) spatial structure with equal connectivity (smallest vertex cut) to the Best model.

—*Full*: QC model having a spatial structure that is fully-connected ($\frac{n(n-1)}{2}$ edges among n sensors).

—*Empty*: QC model having a completely disconnected spatial model. This is identical to a temporal-only QC model.

These QC systems are compared in terms of their κ, precision, and recall scores as a function of σ_n^2 as in the bottom portion of Figure 21. The results can be seen in Figure 22.

In both the FishNet and Grand St. Bernard datasets, our learned QC Model (*Best* model) clearly outperforms the other 3 QC models in κ and precision. The difference in performance is most pronounced in the FishNet deployment. Recall scores are comparable for both FishNet and Grand St. Bernard across all levels of σ_n^2 and all four model types. The *Full* model performs slightly better than than the *Worst* model in κ and precision. This suggests that though there are disadvantages to assuming full spatial connectivity, assuming a densely connected spatial model incurs less error than assuming a spatial model with few or no connections (as in the *Empty* model). The performance gain of the *Best* QC model over the *Full* model is less significant in the Grand St. Bernard dataset. We suspect this stems from the models being trained on data observed in mid September and tested on data from mid to late October (a seasonal transition period). Both our learned spatial model and the fully-connected model will fit the test data poorly, because the training data has little resemblance to the test data. Thus, neither model has a clear advantage over the other."

6.5 Discussion

The statistic of principal interest to our quality control problem is recall. If our method can correctly identify and filter out all nonanomalous data points, then the expert can save time by considering only those points that our model has marked as anomalous. We want to filter out as many existing anomalies from the data prior to review by a domain expert (in order to save time) and prior to publication of the data (in order to prevent distribution of invalid measurements). Our noise

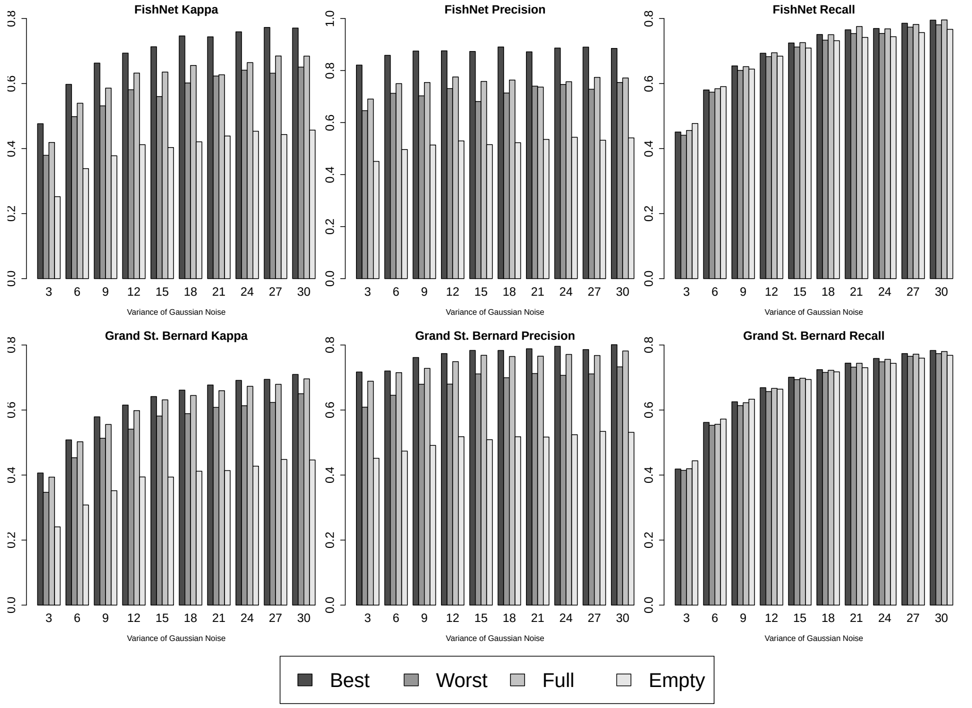


Fig. 22. κ, Precision, and Recall for the FishNet (top) and Grand St. Bernard (bottom) noise injection experiments. X-axis refers to the magnitude of the noise injected (variance in degrees Celsius). The Y-axis corresponds to the κ, Precision, or Recall score.

injection experiments confirm that the worst case for recall is when there exist low-variance anomalies in the data. This stands to reason, for if the anomalies we need to detect fall within the range of noise for the non-anomalous data, they will be nearly-indistinguishable from the real data. Furthermore, if they are frequent, then when the model is fitted to the data, it will use them to define the normal level of variation.

Nevertheless, we have demonstrated that we can obtain recall rates of .70 when the average variance of the noise is 15 ℃ and close to .80 for values of $\sigma_n^2 \geq 20$ ℃. While this may seem unreasonable in terms of noise levels we should expect to encounter in real-world scenarios, consider that approximately 70% of the additive noise is less than or equal to one unit of standard deviation (≈ 3.87 ℃ for $\sigma_n^2 = 15$ and 4.47 ℃ for $\sigma_n^2 = 20$). In addition, we are maintaining a low false positive rate (4 to 7%) for most values of η. In cases where there are fewer complete sensor outages and the temperature data in the training distribution more closely matches that in the testing distribution, we would expect these values to further improve.

With respect to creating a sparse representation of the joint distribution of SensorScope stations, the applied structure learning algorithm resulted in a savings of parameters in both FishNet and Grand St. Bernard. Each variable (or station) in a linear-Gaussian model is specified by a scalar mean and variance (2 parameters) in addition to a weight for every parent (1 parameter for every arc in the graph). Thus, the total number of parameters for a graphical representation of the joint distribution is $2n + k$ where n is the number of variables in the model and k is the number of arcs or edges. The full joint distribution would consist of an n-dimensional mean vector and an $n \times n$ covariance matrix, of which $n + \frac{n(n-1)}{2}$ entries would need to be specified (a total of $2n + \frac{n(n-1)}{2}$ parameters). The created network structure for FishNet contained 12 arcs for a total of $12 + 2 * 6 = 24$ parameters. The regular full joint distribution would required $2 * 6 + \frac{6(6-1)}{2} = 27$ parameters. The savings in this case was minimal; however, the FishNet station covers a relatively small and homogeneous area compared to Grand St. Bernard and thus we expect some measurable correlation among all the sensors at the deployment. The Grand St. Bernard network consisted of 24 edges, so the Bayesian network representing this model requires $2 \times 9 + 24 = 42$ parameters. The full joint distribution would require $2*9+\frac{9(9-1)}{2} = 54$ parameters to be completely specified. We see that the savings, in terms of parameters to be estimated, grows very quickly as the number of sensors increases, and that we can exploit spatial 1heterogeneity to provide a more compact representation. The number of spatial structures we considered in determining the best spatial models (size of the MEC for B_{post} in Table I) for FishNet and Grand St. Bernard were 3 and 8, respectively. This result is consistent with empirical data obtained by experiments involving the evaluation of MEC class sizes [Gillispie and Perlman 2002].

Finally, our empirical results also show that sparseness in the spatial model represents a form of regularization. Figure 22 shows that fully connected spatial models behave worse or on par with our learned spatial models, which contain fewer arcs. As we cannot evaluate the entire set of all spatial models for each dataset, we cannot be certain there do not exist even sparser models that perform better. However, we have some evidence from the *Worst* and *Empty* models' performance in Section 6.4 that we cannot capture all the necessary correlative relationships with fewer edges. The *Worst* spatial model for FishNet contained 10 edges compared to 12 in the learned model, and the *Worst* spatial structure for Grand St. Bernard contained 19 arcs compared to 24 in the learned model; neither performed as well as the *Best* model in each dataset. The placement of the edges in the *Worst* model is an additional contributing factor to its poor performance.

7. RELATED WORK

A simple (though common) approach to data-anomaly detection is to provide a visual representation of the data and allow a domain expert to manually inspect, label, and remove anomalies. In Mourand and Bertrand-Krajewski [2002], this method is improved upon through the application of a series of logical tests to prescreen the data. These tests include range-checks to insure that the observations fall within reasonable domain limits, similar checks for the signal's gradient, and direct comparisons to redundant sensors. The ultimate goal is to reduce the amount of work the domain expert has to do to clean the data, which is consistent with our approach.

Temporal methods evaluate a single observation in the context of a time segment (sliding window) of the data stream or previous observations corresponding to similar periods in cyclical time-series data. Reis et al. [2003] use a predictor for daily hospital visits based on multiday filters (linear, uniform, exponential) that lend varying weight to days in the current sliding window. The motivation for such an approach is to reduce the effect of isolated noisy events creating false positives or false negatives in the system, as might occur with a single-observation-based classifier. In a similar vein, Wang et al. [2005] construct a periodic autoregressive model (PAR, [Chatfield 2000]), which varies the weights of a standard autoregressive model according to a set of user-defined periods within the time series. A daily visitation count is predicted by the PAR model, and if it matches the observed value, then the PAR model is updated with the observation; otherwise, the value is flagged as anomalous, an alarm is raised, and the observation is replaced with a uniformly smoothed value over a window containing the last several observations. A machine-learning based approach was adopted by Wong et al. [2002] wherein the logical tests, or rules, are learned in an online setting. Past observations (taken from set lag periods representative of current temporal context) are mined for rules stated as reasonable values for individual, pairs, or tuples of attributes. The significance of the rules are determined by Fisher's Exact Test.

Spatial methods are useful in cases where there exist additional sensors distributed over a geographic area. The intuition is that if an explicit spatial model exists that can account for the discrepancies between observed values at different sites, then these sensors can, in effect, be considered redundant. An example of this approach can be found in Daly et al. [2005], where each distributed sensor is held out from the remaining set of sensors, and its recorded observation validated against an interpolated value from the remaining set. Each station's value in the network is given a weight associated with confidence in its estimate. This confidence value is calculated using a set of summary statistics based on that station's latest observation in the context of its historical record. Unlike our approach, there is no specific attempt to model the joint distribution between all stations or the overall correlation between sensors in the network. Moreover, this approach relies on a significant historical record for each station in the network in order to compute the necessary summary statistics for that station.

Belief Networks [Pearl 1988] have been employed for sensor validation and fault detection in domains such as robotic movement, chemical reactors, and power plant monitoring [Nicholson and Brady 1992; Mehranbod et al. 2003; Ibarguengoytia et al. 1996]. Typically, the uncertainty in these domains is limited to the sensor's functionality under normal and inoperative conditions. That is, the processes in these domains function within some specified boundaries with a behavior that can be modeled by a system of known equations [Isermann 2005; Aradhye 1997]. Ecological domains are challenging because accurate process models encompassing all relevant factors are typically unavailable [Hill and Minsker 2006]; consequently, uncertainty must be incorporated into both the process and sensor models. Eskin [2000] handles this uncertainty with a mixture model over the true and anomalous data, which is similar to our observation variable once we have marginalized away the sensor state. The distribution parameters are learned iteratively over each sample in the dataset. For each value, the change in likelihood of moving that value's membership from the clean to the anomalous distribution is computed; if the the likelihood increases, the value changes membership, it becomes anomalous permanently (it cannot rejoin the clean distribution), and the nonanomalous distribution parameters are updated. Das et al. [2007] use the probabilistic approach in the multivariate setting in which rare co-occurrences of attribute values are not, in and of themselves, indicative of anomalous values. Here, pairs or tuples of attributes are probabilistically scored based on their values; however, they are normalized by likelihood of the individual values taking on those assignments independently, as determined by the training data (to add support to low-frequency events). An entire record (consisting of multiple attribute tuples) is then scored according to the rarest tuple of attribute values within that record.

Perhaps most related to our own work, Hill et al. [2007] apply a DBN model to analyze and diagnose anomalous wind velocity data. The authors explore individual sensor models as well as a coupled-DBN model that attempts to model the joint distribution of two sensors. The nature of the data-anomaly types in the data appear to be either short-term or long-term malfunctions in which the wind speed drastically increases or decreases; consequently, a first-order Markov process is sufficient to determine sharp rates of increase or decrease in wind speed. The joint distribution is modeled as a multivariate Gaussian conditioned on the joint state of the respective sensors (represented as a discrete set of state pairs). Our current approach primarily differs in the scale (number of sensors we are trying to simultaneously monitor) and that we attempt to discover the correlative structure between the sensors. Instead of assuming a full covariance matrix over the joint distribution of sensors and computing the MLE parameters for that matrix, we apply structure learning to obtain a sparse representation of the joint distribution.

8. CONCLUDING REMARKS AND NOTES ON FUTURE RESEARCH

This article has described a new type of dynamic environmental monitoring based on short-term wireless sensor deployments, as well as demonstrated an accompanying need for adaptive, automated quality control. We have provided background information regarding the SensorScope Project and have given examples of the data collected and the data anomalies contained therein. However, our primary contribution has been to offer an initial means of automating QC in this domain. Our experimental results thus far demonstrate that a Dynamic Bayesian Network approach, based on a generative model of the deployment site, can diagnose many of the data-anomaly types present in ecological data. Further, in all but the severest case of a complete site outage, the model is able to reconstruct reasonable estimates of missing or corrupted data from individual sensors or subsets of sensors. We have shown that structure learning techniques can be successfully applied in this domain to learn a compact representation of the covariance matrix over the generative distribution, and that this sparse matrix performs better or comparable to a fully specified covariance structure.

Thus far we have only applied our method to detect data anomalies present in air temperature sensor streams. We suspect that other environmental data types may provide more challenges to our approach. For example, wind velocity sensors may demonstrate significantly less temporal and spatial correlation over the relatively small geographical areas they are deployed, or surface-temperature data may not be very spatially correlated if the observation area displays surface heterogeneity.

However, a model that examines the correlation across these phenomenon may overcome these challenges. Other domains that may be difficult to perform QC on exclusively (precipitation, soil moisture, solar radiation) may be leveraged with other correlated phenomena to produce a truly inclusive system for quality control.

With regards to structure learning, the BGe metric and hill-climbing search represent only one prior-based technique for determining the underlying spatial model. In learning a compact form of the covariance matrix, there appear to be two primary methodologies. Standard machine learning approaches focus on the discovery of some metric to score child/parent configurations and a search algorithm over the space of DAGs that may penalize nonsparse representations. For example, Tsamardino et al. [2006] attempt to determine a candidate set of neighbors (the Markov Blanket) using a Max-Min Parents heuristic for each variable in the network and then employ hill-climbing over the subset. Schmidt et al. [2007] similarly attempt to identify a subset of variables to consider as a potential set of neighbors using L1-Regularization. However, both of these methods were developed to address very large databases containing thousands of variables (gene expression data, etc.) with relatively few samples – situations in which overfitting is a significant concern. While ecological sensor networks may include dozens of sensors at a deployment, it seems unlikely that the aforementioned techniques would be necessary due to the quantity and frequency at which data is collected. The second approach focuses on learning the covariance matrix directly rather than iteratively. While perhaps not feasible for domains of high dimension, this method does have the advantage of performing a more global evaluation of the structure, making it more robust to local maxima (unlike hill-climbing). The work of Yuan and Li [2007] is an example of this approach, where Lasso regression is used as part of an optimization to force off-diagonal elements of the covariance matrix toward 0. The result is an undirected graph representing the covariance structure and, as we are not primarily interested in developing causal models of the sensor correlations, this would be suitable in the current domain. Also of interest would be a way to integrate our fixed temporal model into the hill-climbing search for an optimal spatial structure. That is, we would like the scoring function to take into account that lag variables will be appended to each of variables in the graph in a specific manner when evaluating candidate structures.

An additional direction for future work is to extend this model to an online learning scheme, in which the spatial structure and parameterization is refined over time. Given that the BGe metric requires a prior network structure as an initialization point for search, one could conceive of an algorithm in which the network learned on incremental batches of observations served as the prior for the next network. We could begin with a very weak assumption on the generative distribution (total independence among sensors with each sensor having a univariate Normal distribution over the range of plausible domain values), and use this as our initial QC system. Those points not labeled as anomalous by this primitive model would then be employed to train a more sophisticated spatial model, and then the process could be repeated. On a related note, there may be other metrics for conditional independence that merit exploration, especially if we are to loosen our assumption on normally distributed generative model or on a linear correlative relationship between the variables.

ACKNOWLEDGMENTS

We would like to thank Marc Parlange and the EFLUM Group for hosting and supporting this research at the EPFL. We would also like to thank Mounir Krichane and Guillermo Barrenetxea for providing us with the SensorScope datasets and for their technical advice regarding the sensor stations. Finally, we wish to acknowledge Marc-Olivier Boldi for his statistical expertise and his contributions to the development and debugging of our QC model.

REFERENCES

Andersson, S., Madigan, D., and Perlman, M. D. 1995. A characterization of markov equivalence classes for acyclic digraphs. *Annals of Statistics 25*, 505–541.

Aradhye, H. 1997. Sensor fault detection, isolation, and accommodation using neural networks, fuzzy logick, and bayesian belief networks. M.S. thesis, University of New Mexico, Albuquerque, NM.

Chatfield, C. 2000. *Time-Series Forecasting.* Chapman & Hall/CRC, New York, NY.

Cohen, J. A. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement 20*, 37–46.

Daly, C., Redmond, K., Gibson, W., Doggett, M., Smith, J., Taylor, G., Pasteris, P., and Johnson, G. 2005. Opportunities for improvements in the quality control of climate observations. In *15th AMS Conference on Applied Climatology.* American Meteorological Society, Savannah, GA.

Das, K. and Schneider, J. 2007. Detecting anomalous records in categorical datasets. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, New York, NY, USA, 220–229.

Dean, T. and Kanazawa, K. 1988. Probabilistic temporal reasoning. In *Proceedings of the Seventh National Conference on Artificial Intelligence.* MIT Press, Cambridge, Massachusetts, 524–529.

Dechter, R. 1996. Bucket elimination: A unifying framework for probabilistic inference. In *Twelthth Conf. on Uncertainty in Artificial Intelligence*, E. Horvitz and F. Jensen, Eds. Morgan Kaufmann, Portland, Oregon, 211–219.

Dereszynski, E. and Dietterich, T. 2007. A probabilistic model for anomaly detection in remote sensor data streams. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI 2007)*, R. Parr and L. van der Gaag, Eds. AUAI Press, Vancouver, B.C., 75–82.

Eskin, E. 2000. Anomaly detection over noisy data using learned probability distributions. In *In Proceedings of the Seventeenth International Conference on Machine Learning*, P. Langley, Ed. Morgan Kaufmann, Stanford, CA, 255–262.

Geiger, D. and Heckerman, D. 1994. Learning Gaussian networks. Tech. Rep. MSR-TR-94-10, Microsoft Research, Redmond, WA.

Geman, S. and Geman, D. 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 6,* 20, 721–741.

Gillispie, S. B. and Perlman, M. D. 2002. The size distribution for markov equivalence classes of acyclic digraph models. *Artificial Intelligence 141,* 1-2, 137 – 155.

Hill, D. J. and Minsker, B. S. 2006. Automated fault detection for in-situ environmental sensors. In *7th International Conference on Hydroinformatics.* Research Publishing, Nice, France.

Hill, D. J., Minsker, B. S., and Amir, E. 2007. Real-time bayesian anomaly detection for environmental sensor data. In *Proceedings of the 32nd conference of IAHR.* International Association of Hydraulic Engineering and Research, IAHR, Venice, Italy.

Hodge, V. and Austin, J. 2004. A survey of outlier detection methodologies. *Artif. Intell. Rev. 22,* 2, 85–126.

Ibarguengoytia, P., Sucar, L., and Vadera, S. 1996. A probabilistic model for sensor validation. In *Twelthth Conference on Uncertainty in Artificial Intelligence*, E. Horvitz and F. Jensen, Eds. Morgan Kaufmann, Portland, Oregon, 332–333.

Isermann, R. 2005. Model-based fault detection and diagnosis: Status and applications. In *Annual Reviews in Control*. Vol. 29. Pergamon Press Ltd., St. Petersburg, Russia, 71–85.

Kalman, R. E. 1960. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering 82,* Series D, 35–45.

keen Wong, W., Moore, A., Cooper, G., and Wagner, M. 2002. Rule-based anomaly pattern detection for detecting disease outbreaks. In *In Proceedings of the 18th National Conference on Artificial Intelligence*, K. Ford, Ed. AAAI Press, Edmonton, Alberta, 217–223.

Lauritzen, S. 1992. Propogation of probabilities, means, and variance in mixed graphical association models. *Journal of The American Statistical Association 87,* 420, 1098–1108.

Lauritzen, S. and Wermuth, N. 1989. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The Annals of Statistics 17,* 1, 31–57.

Matheron, G. 1963. Principles of geostatistics. *Economic Geology 53,* 8 (December), 1246–1266.

Mehranbod, N., Soroush, M., Piovos, M., and Ogunnaike, B. A. 2003. Probabilistic model for sensor fault detection and identification. *AIChe Journal 49,* 7, 1787–1802.

Mourad, M. and Bertrand-Krajewski, J. 2002. A method for automatic validation of long time series of data in urban hydrology. *Water Science & Technology 45,* 4-5, 263–270.

Murphy, K. P. 1998. Inference and learning in hybrid Bayesian networks. Tech. Rep. UCB/CSD-98-990, University of California, Berkeley, California. January.

Nicholson, A. E. and Brady, J. M. 1992. Sensor validation using dynamic belief networks. In *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 207–214.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Rabiner, L. R. 1990. A tutorial on hidden markov models and selected applications in speech recognition. In *Readings in speech recognition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 267–296.

Reis, B. Y., Pagano, M., and Mandl, K. D. 2003. Using temporal context to improve biosurveillance. *Proceedings of the National Academy of Science 100,* 4, 1961–1965.

Russell, S. and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Pearson Education, Inc., Upper Saddle River, New Jersey.

Schmidt, M., Nicolescu-Mizil, A., and Murphy, K. 2007. Learning graphical model structure using L1-regularization paths. In *Proceedings of the Twenty-Seceond AAAI Conference on Artificial Intelligence*. AAAI Press, Vancouver, British Columbia, 1278–1284.

Sensirion 2005. *SHT1x / SHT7x Humidity & Temperature Sensor*. Sensirion, Sensirion AG, Laubisrütistr. 50, CH-8712 Stäfa ZH, Switzerland.

Szalay, A. and Gray, J. 2002. The world-wide telescope, an archetype for online science. Tech. Rep. MSR-TR-2002-75, MSR.

Tsamardinos, I., Brown, L. E., and Aliferis, C. F. 2006. The max-min hill-climbing Bayesian network structure learning algorithm. *Mach. Learn. 65,* 1, 31–78.

Viera, A. J. and Garrett, J. M. 2005. Understanding interobserver agreement: The kappa statistic. *Family Medicine 37,* 5, 360–363.

Wang, L., Ramoni, M. F., Mandl, K. D., and Sebastiani, P. 2005. Factors affecting automated syndromic surveillance. *Artificial Intelligence in Medicine 34,* 3, 269–278.

Yuan, M. and Lin, Y. 2007. Model selection and estimation in the Gaussian graphical model. *Biometrika 94,* 1, 19–35.

Received Month Year; Revised Month Year; Accepted Month Year