7th ISE & 8th HIC
Chile, 2009

REAL-TIME ANOMALY DETECTION IN PRECIPITATION SENSORS

DAVID J. HILL
National Center for Supercomputing Applications, University of Illinois
Urbana, IL 61820, USA

BARBARA S. MINSKER
Department of Civil and Environmental Engineering, University of Illinois
Urbana, IL 61820, USA

EYAL AMIR AND JAESIK CHOI
Department of Computer Science, University of Illinois
Urbana, IL 61820

Recent advances in sensor technology are facilitating the deployment of sensors into the environment that can produce measurements at high spatial and/or temporal resolutions. Not only can these data be used to better characterize systems for improved modeling, but they can also be used to improve understanding of the mechanisms of environmental processes. With large volumes of data arriving in near real time, however, there is a need for automated anomaly detection to identify data that deviate from historical patterns. These anomalous data can be caused by sensor or data transmission errors or by infrequent system behaviors that may be of interest to the scientific or public safety communities. This study develops an automated anomaly detection method that employs a Dynamic Bayesian Network to assimilate data from multiple heterogeneous sensors into an uncertain model of the current state of the environment. Filtering (e.g., Kalman filtering) can then be used to infer the likelihood that a particular sensor measurement is anomalous. Measurements with a high likelihood of being anomalous are classified as such. The method developed in this study performs fast, incremental evaluation of data as they become available; scales to large quantities of data; and requires no *a priori* information regarding process variables or types of anomalies that may be encountered. The performance of the anomaly detector developed in this study is demonstrated using a precipitation sensor network composed of a NEXRAD weather radar and five telemetered rain gauges deployed by the USGS. The results indicate that the method performs well at identifying anomalous data caused by a real sensor failure.

INTRODUCTION

This study addresses anomaly detection in tipping-bucket rain gauge (TBRG) data. This type of data is used to correct bias in radar-rainfall estimates from the Next Generation Weather Radar (NEXRAD) system [10] and is increasingly being used for (near) real-

time applications (e.g., flood forecasting [2]). However, to the authors' knowledge, no studies have yet addressed automated methods for quality analysis and control (QA/QC) in TBRG data streams. Anomaly detection is the process of identifying data that deviate markedly from historical patterns [4]. Anomalous data can be caused by sensor or data transmission errors or by infrequent system behaviors that may be of interest to scientific and regulatory communities. Anomaly detection performed in real time has many practical applications for environmental sensors such as real-time QA/QC, adaptive sampling, and anomalous event detection.

Successful real-time anomaly detection in environmental streaming data must surmount four key challenges. First, because new data are collected continuously (usually at high frequencies), the entire data set cannot usually be held in memory nor can all existing data be reprocessed when new measurements become available. Second, real-time decisions can only use previous observations; thus, future observations cannot be used for anomaly classification. Third, environmental sensors go off line frequently; thus, if a significant number of specific historical measurements are necessary to process a new measurement, then many measurements will not be able to be processed. Finally, sensors deployed in the natural environment behave in unexpected ways; thus, no *a priori* definition of the types of anomalies that may be encountered is available. Recently, Hill et al. [3] presented a Dynamic Bayesian Network (DBN)–based method for anomaly detection that addresses these challenges and demonstrated its efficacy for identifying anomalies in data from a network of sensors deployed in Corpus Christi Bay, Texas, which measure windspeed, wind direction, barometric pressure, and air temperature. These data differ from TBRG data because they vary more smoothly in time than do rainfall measurements.

This study extends the method of Hill et al. [3] by introducing a method to learn the DBN model parameters on line while performing anomaly detection, thus facilitating the rapid deployment of the anomaly detection method on new sensor networks and eliminating the need for the DBN model to be retrained to accommodate temporal changes in the system dynamics. The new anomaly detection method is then demonstrated using data from a precipitation sensor network composed of a NEXRAD weather radar and five telemetered rain gauges.

METHODS

State space models assume that there is an underlying hidden state of the environment that is observed via noisy sensors and that this hidden state evolves in time. DBNs provide a framework for performing Bayesian inference on state space models. For example, consider the first order Markov state space model

$$\begin{aligned} X_t &= A(X_{t-1}) + \varepsilon_X \\ Z_t &= C(X_t) + \varepsilon_Z \end{aligned} \tag{1}$$

where X_t is the system state vector at time *t*; Z_t is a vector of observations of the state variables at time *t*; $A(\ldots)$ is the system model, which describes the dynamics of the state variables from time *t*-1 to *t*; $C(\ldots)$ is the observation model, which describes the relation of the sensor output to the state variable vector; and ε_X and ε_Z are random variables representing the noise in the system dynamics and the sensor measurements, respectively. This model can be represented by the DBN shown in Figure 1.

To apply the state space model of Equation 1 to a particular sensor network, it is necessary to specify the distributions of ε_X and ε_Z as well as the system and observation models. Often the form of these distributions (e.g., multivariate Gaussian) and models (e.g., linear) is known or can be assumed, but the parameters are unknown and must be inferred from the data. The next subsection presents a method of learning these parameters on line as new measurements are made.

Learning DBN Parameters On Line

The parameters of the DBN can be learned via Expectation-Maximization (EM), a two-step iterative process that calculates the maximum likelihood estimate of the unobserved parameters by maximizing the expectation of the joint probability of the system state and observations given the sensor measurements and the maximum likelihood estimate of the model parameters.

For the DBN in Figure 1, the expected likelihood function takes the following form:

$$E\left[P\left(\{X\},\{Z\}\right)|\,\{Z\},\Theta\right]=E\left[P\left(X_0\mid\Theta\right)\prod_{t=2}^{T}P\left(X_t\mid X_{t-1},\Theta\right)\prod_{t=1}^{T}P\left(Z_t\mid X_t,\Theta\right)\right] \quad (2)$$

where $\{X\}$ is the set of unobserved system states, $\{Y\}$ is the set of observations of the system states given by the sensors, and $\Theta=\{\theta_1,\ldots\theta_n\}$ are the model parameters [9]. The value of the parameters that maximize the likelihood of the observations can be found by setting the partial derivative of the likelihood function with respect to each parameter to zero. To simplify the math, the logarithm of the likelihood function can be used in place of the likelihood function. Because Equation 2 requires the entire set of observations, this is inherently a batch learning method and thus cannot be applied directly to streaming data, since these data series' have no defined endpoint. However, Neal and Hinton [8] have shown that incrementally maximizing a factored decomposition of Equation 2, namely

$$E\left[P\left(\{X\},\{Z\}\right)|\,\{Z\},\Theta\right]=\prod_{t}E\left[P\left(X_t,Z_t\mid\{Z\}_t\Theta\right)\right] \quad (3)$$

where $\{Z\}_t$ is the set of observations made prior to and including time *t*, produces the same maximum likelihood estimates as maximizing Equation 2. This allows for an on-line learning method where the parameters are updated sequentially as new data arrive.

Because this incremental method is performed using only the observations prior to and including time *t*, it is well suited for on-line learning of DBN parameters. The next

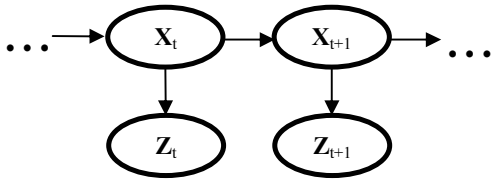

Figure 1. Dynamic Bayesian Network representation of first-order Markov state space model. Arrows represent the conditional dependence relationships, and ellipses indicate a repeated graphical pattern.

section demonstrates the application of the incremental version of EM to a particular type of DBN: the Kalman filter.

On-Line Learning Using a Kalman Filter

If the state variables and their measurements are Gaussian, and the transition and observation models are linear, then Equation 1 can be rewritten as

$$\begin{aligned} X_{t+1} &= AX_t + N(0,Q) \\ Z_t &= CX_t + N(0,R) \end{aligned} \tag{4}$$

where $N(\mu,\Sigma)$ represents a multivariate Gaussian distribution with mean μ and variance Σ, and Q and R are the process and measurement noise covariance matrices, respectively. Equation 4 is the Kalman filter model [5], which can be solved incrementally for new measurements using the well-known Kalman update equations

$$\begin{aligned} \widetilde{X}_{t+1} &= A\widetilde{X}_t + K_{t+1}\left(Z_{t+1} - CA\widetilde{X}_t\right) \\ \widetilde{P}_{t+1} &= \left(I - K_{t+1}C\right)\left(A\widetilde{P}_t A^{\mathrm{T}} + Q\right) \\ K_{t+1} &= \left(A\widetilde{P}_t A^{\mathrm{T}} + Q\right)C^{\mathrm{T}}\left(C\left(A\widetilde{P}_t A^{\mathrm{T}} + Q\right)C^{\mathrm{T}} + R\right)^{-1} \end{aligned} \tag{5}$$

where $\widetilde{X}_t$ is the expected system state given the observations up to and including time *t*, $\widetilde{P}_t$ is the covariance matrix of the expected system state at time *t*, I is the identity matrix, $^{\mathrm{T}}$ indicates the matrix transpose, and K_t is the Kalman gain matrix at time *t*.

From Equation 4, it can be seen that this model has four parameters: A, C, Q, and R. In cases where the sensors directly measure the environmental state variables (as is typical), C can be specified as the identity matrix. Thus,, only A, Q, and R need to be learned. Because the state variables and their measurements are multivariate Gaussian, their conditional distributions can be expressed as

$$\begin{aligned} P\left(X_{t+1} \mid X_t, A, Q, R\right) &= \alpha\left|Q\right|^{-1/2}\exp\left(-\frac{1}{2}\left(X_{t+1} - AX_t\right)^T Q^{-1}\left(X_{t+1} - AX_t\right)\right) \\ P\left(Z_t \mid X_t, A, Q, R\right) &= \alpha\left|R\right|^{-1/2}\exp\left(-\frac{1}{2}\left(Z_t - CX_t\right)^T R^{-1}\left(Z_t - CX_t\right)\right) \end{aligned} \tag{6}$$

where α is a normalization constant. If Equation 6 is used in Equation 4, and the expectation is taken with respect to the measurements and the current value of the model

parameters (using Equation 5), then the incremental value of Equation 3 can be determined at any time *t*. Maximizing this equation by setting its partial derivatives with respect to A, Q, and R to zero leads to the following incremental updates

$$A^{new} = \sum_{t=2}^{T}\left(\widetilde{P}_{t,t-1} + \widetilde{X}_t \widetilde{X}_{t-1}^{\mathrm{T}}\right)\left[\sum_{t=2}^{T}\left(\widetilde{P}_{t-1,t-1} + \widetilde{X}_{t-1}\widetilde{X}_{t-1}^{\mathrm{T}}\right)\right]^{-1}$$

$$Q^{new} = \frac{1}{T-1}\left(\sum_{t=2}^{T}\left(\widetilde{P}_{t,t} + \widetilde{X}_{t-1}\widetilde{X}_{t-1}^{\mathrm{T}}\right) - A^{new}\sum_{t=2}^{T}\left(\widetilde{P}_{t-1,t-1} + \widetilde{X}_{t-1}\widetilde{X}_{t-1}^{\mathrm{T}}\right)^{\mathrm{T}}\right) \tag{7}$$

$$R^{new} = \frac{1}{T}\left[\left(Z_t - C\widetilde{X}_t\right)\left(Z_t - C\widetilde{X}_t\right)^{\mathrm{T}} + C\widetilde{P}_t C^{\mathrm{T}}\right]$$

Equations 5 and 7 can be used together to sequentially infer the system state and the Kalman filter parameters, given newly observed data.

DBN Modeling of Precipitation

In this study, the Kalman filter derived in the previous section is used to nowcast the Δt-minute precipitation accumulation at the location of gauges within a precipitation gauge network. Thus, the variable X_t in Equation 1 can be represented as

$$Z_t = \begin{bmatrix} a_t^1 & \cdots & a_t^n \end{bmatrix}^{\mathrm{T}} \tag{8}$$

where a_t^i is the rainfall accumulation at location *i* and time *t*. Two sets of observations will be compared for assimilation into the DBN model: observations from the gauges alone, and combined gauge and radar-rainfall measurements derived from NEXRAD. Thus, in the first case (hereafter referred to as the "gauge filter"), the measurement vector Z_t in Equation 1 will take the form

$$Z_t = \begin{bmatrix} g_t^1 & \cdots & g_t^n \end{bmatrix}^{\mathrm{T}} \tag{9}$$

where g_t^i is the gauge measurement at location *i* and time *t*. In the second case (hereafter referred to as the "combined filter"),

$$Z_t = \begin{bmatrix} g_t^1 & \cdots & g_t^n & r_t^1 & \cdots & r_t^n \end{bmatrix}^{\mathrm{T}} \tag{10}$$

where r_t^i is the radar measurement at location *i* and time *t*.

In this paper, the on-line learning Kalman filter described above will be used to model the precipitation at the gauge locations, assuming that the rainfall is linear Gaussian. Although the intermittency of rainfall can lead to nonlinearities and non-Gaussian behavior, Kalman filters have been successful at modeling time series in a wide variety of applications and, as will be shown shortly, do provide reasonable estimates of rainfall accumulations. Thus, Kalman filtering is a good benchmark for comparing more complex DBN-based precipitation models in the future.

Anomaly Detection in Precipitation Gauge Data Streams

Using the DBNs described above, anomalies can be detected in the precipitation gauge data streams by using filtering to sequentially infer the posterior distribution of the state variables and their corresponding observations as new measurements become available from the gauges. The posterior distribution of the observed variables can then be used to construct a Bayesian Credible Interval (BCI) for the most recent set of measurements. The p% BCI indicates that the posterior (i.e., adjusted for the available observations) probability of the observed state variables falling within the interval is p; thus, the BCI delineates the range of plausible values for sensor measurements. For this reason, any measurements that fall outside of the p% BCI will be classified as anomalous. The 100(1-α)% BCI for a new measurement can be calculated as

$$\bar{x} \pm z_{\alpha/2} * \sqrt{\Sigma} \tag{8}$$

where $z_{\alpha/2}$ is the 100(1 − α/2)th percentile of a normal distribution, and Σ is the variance of the posterior distribution of the measurement prediction. The BCI method of anomaly detection has been successfully demonstrated for identifying anomalies in windspeed, wind direction, barometric pressure, and air temperature data streams by Hill et al. [3] using DBNs with off-line parameterization.

CASE STUDY

To test the efficacy of the anomaly detection method developed in this study for QA/QC of rain gauge data, the method was applied to a network of five TBRGs operated by the USGS and located in the Salt Creek Watershed in Du Page County, Illinois. The data from these gauges are used by the USGS Illinois Water Science Center for real-time forecasting of floods in the Salt Creek. These data can be corrupted by errors caused by failed transmissions, false tips, and missed tips, which are currently identified by manual inspection of the real-time data. The first type of error, failed transmission, is identified automatically by the telemetry system, whereas the second type (false tip) is rare and does not affect the modeling significantly. The third type of error (missed tip), however, is difficult to identify with the current QA/QC strategy (personal communication Audrey Ishii, USGS). Thus, the goal of this case study is to deploy the rain gauge anomaly detection methods described above for the purpose of real-time identification of missed tip errors in the telemetered TBRG data.

Rainfall in the Salt Creek Watershed is also monitored by NEXRAD, which has three radar installations in close proximity. In this study, radar-rainfall estimates are derived only from the closest radar station (KLOT) using the Level II base reflectivity data and the method outlined by [6]. In order to identify gauge errors in a timeframe suitable for facilitating real-time modeling of floods in the Salt Creek, a time step Δt of 10 minutes was selected. This time step was chosen because it is a multiple of the 5-minute gauge data and it is equivalent to the time of the longest NEXRAD radar scan [1].

Also, gauge errors are intermittent, so longer time intervals would likely result in the misclassification of large numbers of valid gauge data.

The gauge and combined filters were run using historical data from May 2007–Aug. 2007. This period of historical data was selected because it fell during the warmer months (thus, issues related to radar and gauge quantification of frozen precipitation could be ignored) and because a gauge failure, which was identified for the first time by this study, occurred on Aug. 23. Anomalies were classified using a 99% BCI. Figures 2 and 3 show the data, the rainfall estimates, and the anomaly classifications for the gauge and combined filters, respectively, for the period of Aug. 23 06:00–Aug. 24 04:00. During this time period, it can be seen that three discrete rain events occurred at 08:00, 20:00, and 21:00. Except for gauge B, all of the gauges and the radar capture these rainfall events. Gauge B captures the first two events but reports that no rain is falling during the 21:00 event. Analysis by the USGS Water Science Center concluded that the data reported by gauge B during the 21:00 event was the result of several missed tip errors. Comparing the filtered estimates from both the gauge and combined filters with the gauge data demonstrates that, despite the linear Gaussian assumption, both filters appear to provide reasonable predictions of the 10-minute rainfall at the gauge site, though the filters tend to underestimate the rainfall peaks, which is likely a symptom of the assumption of Gaussian dynamics. Comparing the filtered rainfall estimates from the gauge and combined filters reveals that the addition of the radar data improves the combined filter's ability to capture the peaks in the rainfall data. Finally, it appears from these figures that both the gauge and the combined filter correctly classify most of the data resulting from the failure of gauge B. The combined filter misclassifies fewer data, indicating that the inclusion of the radar data helps to better classify sensor failures. However, both DBNs misclassify quite a few valid data points during rain events.

CONCLUSION

This case study introduces a method for learning DBN parameters on line and tests the efficacy of an on-line learning Kalman filter for anomaly classification in a precipitation gauge network. On-line learning of DBN parameters is valuable because it allows the DBN-based anomaly detector to be deployed immediately on a new sensor data stream and it permits the parameters to evolve as the process dynamics evolve, thus eliminating the need to periodically retrain the DBNs to refresh model parameters. Additionally, the results indicate that the assimilation of co-located estimates of rainfall from rain gauges and radar improves the rainfall estimate of the DBN model.

The results presented here focus on the ability of an on-line learning Kalman filter to identify anomalies when the assumption of Gaussian dynamics may not be fully valid, which may lead to misclassifications during storm events. Work is under way to extend the analysis to DBNs that do not rely on the assumption of linear Gaussian dynamics, such as assumed density filters or Rao-Blackwellized particle filters [7].

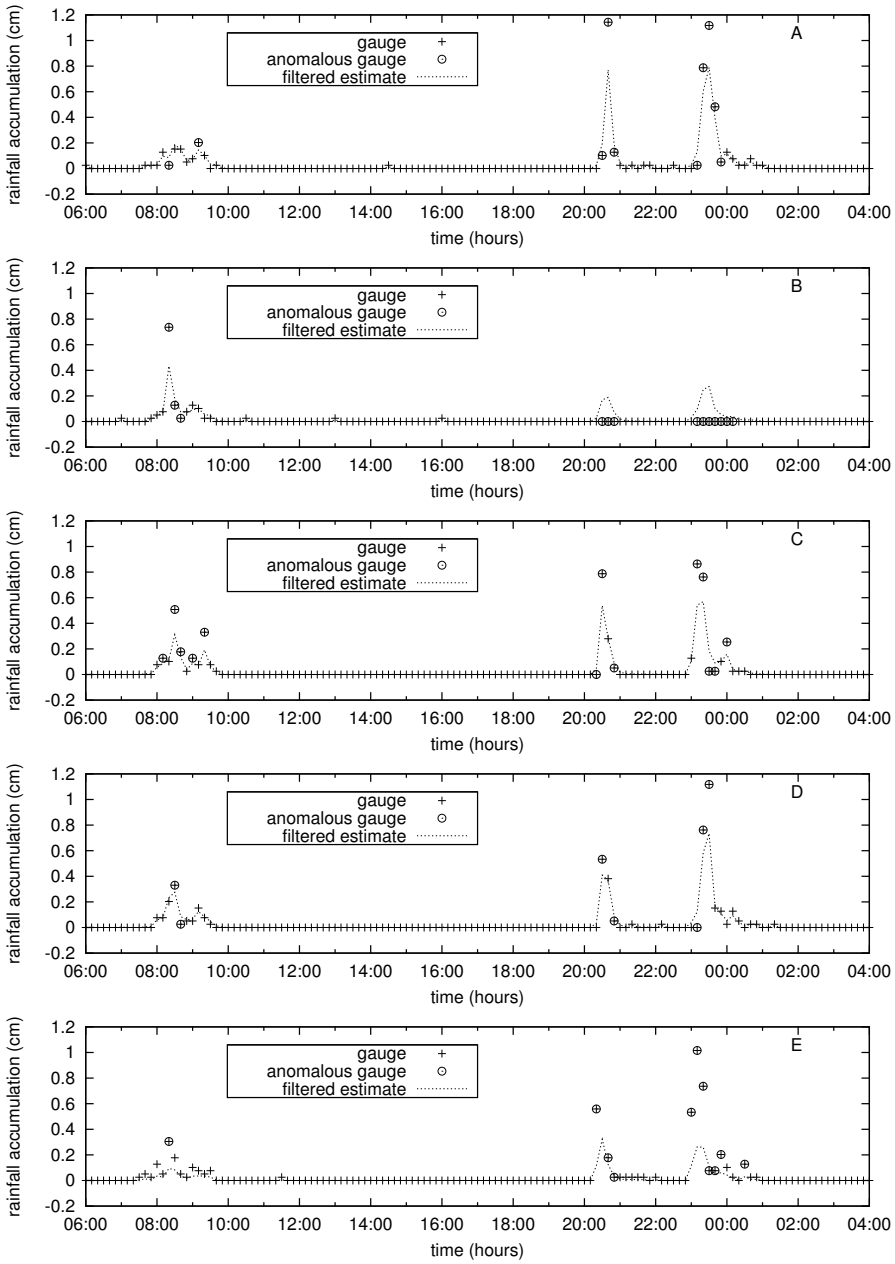

Figure 2. Gauge filter measurements, classification, and filtered estimates for the Aug. 23 06:00 – Aug. 24 04:00 data from gauges A, B, C, D, and E.

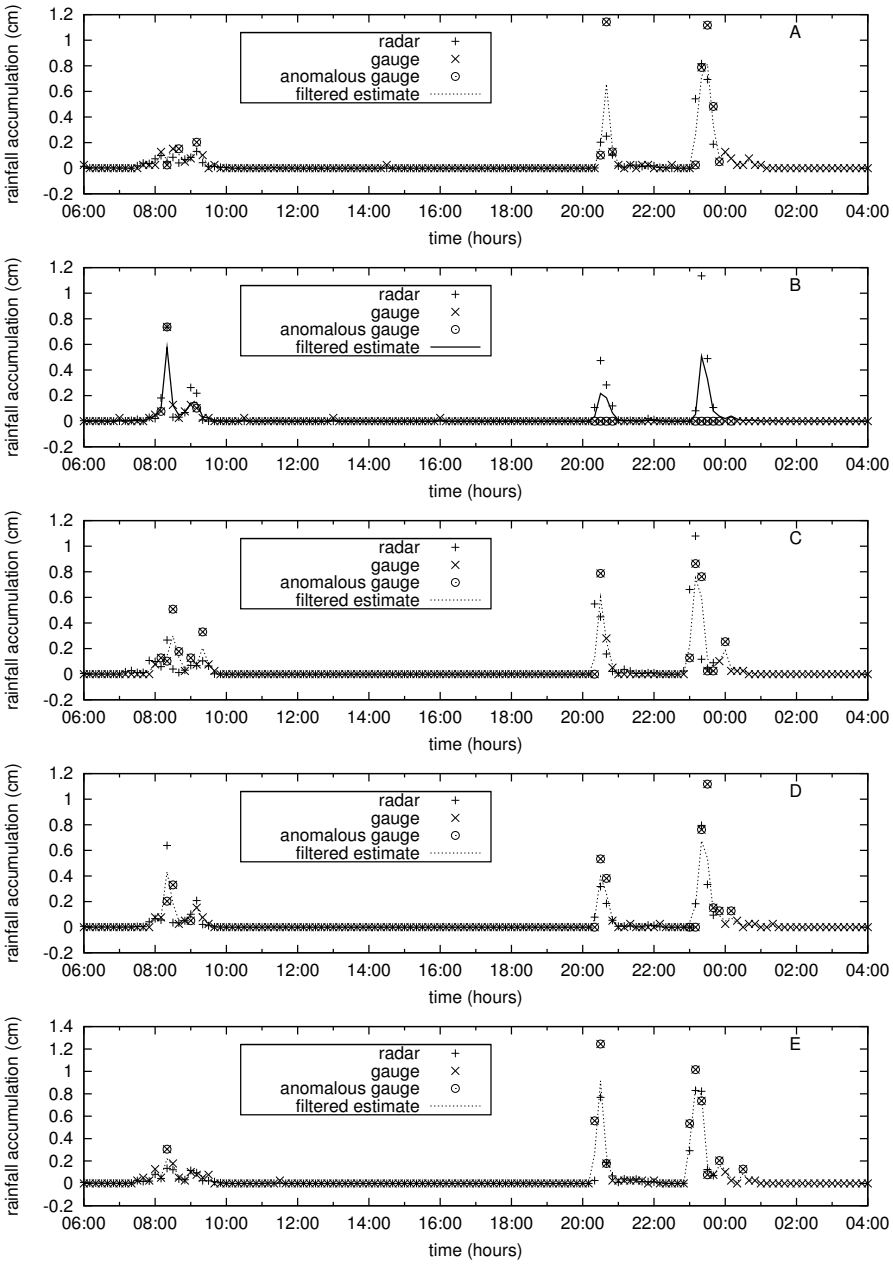

Figure 3. Combined filter measurements, classification, and filtered estimates for the Aug. 23 06:00–Aug. 24 04:00 data from gauges A,B,C,D, and E.

ACKNOWLEDGEMENTS

The authors would like to thank the Metropolitan Water Reclamation District of Greater Chicago and the City of Chicago, which provided data on the Chicago sewer system. This study was funded by the Adaptive Environmental Sensing and Information Systems (AESIS) initiative, which is supported by the University of Illinois at Urbana-Champaign Vice Chancellor for Research, the National Center for Supercomputing Applications (NCSA), and the Department of Civil and Environmental Engineering.

REFERENCES

[1] Fulton, R.A., Breidenbach, J.P., Seo, D.-J., Miller, D.A., and O'Bannon, T., "The WSR-88D rainfall algorithm", *Weather and Forecasting*, June, (1998), pp 377-395.

[2] Habib, E.H., Meselhe, E.A., and Aduvala, A.V., "Effect of local errors of tipping-bucket rain gauges on rainfall-runoff simulations", *Journal of Hydrologic Engineering*, June, (2007), pp 488-496.

[3] Hill, D.J., Minsker, B.S., and Amir, E. "Real-time Bayesian anomaly detection in streaming environmental data", *Water Resources Research,* under review.

[4] Hodge, V.J. and Austin, J., "A survey of outlier detection methodologies", *Artificial Intelligence Review*, Vol. 22, (2004), pp 85-126.

[5] Kalman, R.E. "A new approach to linear filtering and prediction problems", *Transactions of the ASME—Journal of Basic Engineering*, Vol. 82, Series D, (1960), 35-45.

[6] Liu, Y., Hill, D., Rodriguez, A., Marini, L., Kooper, R, Futrelle, J., Minsker, B., and Myers, J. "Near-real time spatiotemporal precipitation virtual sensor creation based on NEXRAD Level II Data in a semantically-enhanced digital watershed", Proc. 16th ACM SIGSPATIAL International Conference, Irvine, CA, November 5-7, (2008).

[7] Murphy, K. "Dynamic Bayesian Networks," In M.I. Jordan *Probabilistic Graphical Models*, forthcoming.

[8] Neal, R.M. and Hinton, G.E., "A view of the EM algorithm that justifies incremental, sparse, and other variants", In M.I. Jordan (Ed.) *Learning in Graphical Models*, Dordrecht: Kluwer Academic Publishers, (1998), pp. 355-368.

[9] Shumway, R.H. and Stoffer, D.S., "An approach to time series smoothing and forecasting using the EM algorithm", *Journal of Time Series Analysis*, Vol. 3, No. 4, (1982), pp 253-264.

[10] Steiner M., Smith, J.A., Burges, S.J., Alonso, C.V., and Darden, R.W., "Effect of bias adjustment and rain gauge data quality control on rainfall estimation", *Water Resources Research*, Vol. 35, No. 8, (1999), pp 2487-2503.