RQE Report: Model-Based Monitoring for Early Warning Flood Detection

Elizabeth Basha *
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
e_basha@mit.edu

Abstract

Predictive environmental sensor networks provide complex engineering and systems challenges. These systems must withstand the event of interest, remain functional over long time periods when no events occur, cover large geographical regions of interest to the event, and support the variety of sensor types needed to detect the phenomenon. Prediction of the phenomenon on the network complicates the system further, requiring additional computation on the microcontrollers and utilizing prediction models that are not typically designed for sensor networks. This paper describes a system architecture and deployment to meet the design requirements and to allow model-driven control, thereby optimizing the prediction capability of the system. We explore the application of river flood prediction using this architecture, describing our initial work on the prediction model, network implementation, component testing and infrastructure development in Honduras, deployment on a river in Massachusetts, and results of the field experiments. Our system uses only a small number of nodes to cover basins of 1000-10000 km^2 using an unique heterogeneous communication structure, incorporating self-monitoring for failure, and adapting measurement schedules to capture events of interest.

1 Introduction

Current work in sensor networks highlights the growing applicability of networks to everyday problems. In the area of monitoring and detecting environmental phenomena, work on habitat monitoring of birds [23], zebras [21, 43], and a redwood tree [37] exemplifies the usefulness of these systems. We are interested in developing systems to monitor large environmental events and to deal with system constraints required for real-world use of these networks.

Predictive environmental sensor networks require addressing several complicated design requirements. The network must cope with element exposure, node failures, limited power, and prolonged use. When the event damages the environment such as floods, hurricanes, forest fires, and earthquakes, this further complicates the requirements. This system must withstand the event, which usually poses a serious hazard to network survival especially those nodes directly measuring the event. Additionally, the system must operate throughout long disaster-free periods, measure a variety of variables contributing to the disaster, thereby requiring heterogeneous sensor support, and communicate over the large geographical regions in which these events occur.

Once the system meets these fundamental sensing design requirements, it then needs to actually predict the event of interest. Most algorithms for this do not conform easily to a sensor network, instead focusing on a centralized computing system with significant processing power and complex system models. This sort of computational power does not exist everywhere we might want to install such a prediction network, especially rural and developing countries, nor do we want to install such computing power. We instead would like to parsimoniously use the computing power on the sensor network to perform this prediction by adaptively sampling data from the network. Model computing on a network requires executing a simplified form of the underlying physical model and developing distributed implementations. Key to this process involves eventually connecting the model to the data collection such that the model drives when and what is measured. As specific measurements are required to reduce the prediction uncertainty, the model identifies this need and, within the network, requests the data from the sensors.

We examine the effects of these constraints and system architectures within the context of river flood prediction. We chose the Aguán River basin in northern Honduras as our test basin after learning about the seriousness of the flood problem there from a non-governmental organization in the region. Specifically, in 1998, Hurricane Mitch caused approximately 5000 deaths, 8000 injuries and 12000 missing after a wall of water swept down the river at night while everyone was sleeping (see Figure 1) [39]. Additionally, in the Aguán region, flooding is not limited to hurricanes, occurring annually due to heavy rainfall. Many lives and property could be saved if people knew the flood was coming and, after flooding occurred, could monitor the river to understand how to best concentrate relief efforts. With this application in mind, we further define the system requirements. The system must withstand river flooding and the severe storms causing the floods, monitor and communicate over a 10000 km^2 river basin, predict flooding autonomously, and limit costs allowing feasible implementation of the system in a developing country.

*Research supervised by Daniela Rus and Sai Ravela. Preliminary draft.

While our eventual goal is Honduras, for practicality reasons and speed of debugging, we also chose a local test site on the upper Charles River at Dover, Massachusetts. The Charles encompasses a basin of 1000 km^2, only one order of magnitude less than our proposed basin, and provides support from the USGS along with verification of our measurements through their sensors.

Figure 1. Aftermath of Hurricane Mitch in 1998 in Northern Honduras

This paper describes a system architecture to meet these requirements along with describing our initial work on the flood prediction model that will drive the network, system implementation, field experiments in Honduras, deployment of the network in Massachusetts, and the result of modeling the Massachusetts field experiment data.

2 Previous Work

Previous work covers a wide variety of topics including sensor networks for environmental monitoring, sensor networks for flood detection, and operational flood detection systems.

Sensor Networks for Environmental Monitoring

Several sensor network systems have been designed for outdoor monitoring purposes especially animal monitoring. While this work does not directly relate to ours, implementations sharing some interesting characteristics including cattle ranch monitoring [31], cattle control [5], sheep monitoring [36], zebra herd monitoring [21, 43], seabird nests [23], and frog vocalizations [17]. Of greater relevance is work in environment monitoring where several projects have implemented related systems.

Tolle [37] developed a sensor network to monitor a redwood tree. Installing nodes throughout the height of the 70 m tree, the system measured air temperature, relative humidity, and solar radiation over a 44 day period. The system logged data every five minutes and transmitted it via GPRS modem to an external computer. All analysis was performed off-line after the test period.

Selavo [29] created a sensor network for measuring light intensity. Each node can connect to 8 resistive or voltage-based sensors, communicating data locally via Zigbee and remotely via a single Stargate at 2.4 GHz with delay tolerance of the data arrival at the base station. They performed a field experiment of 1 day with 7 nodes and have installed 19 sensor nodes in another experiment (but no results were available at time of publication). No data analysis occurred on the nodes.

Guy [14] built a sensor network system that has been installed in four different locations to date. In the James Reserve, a forest setting, the system measured temperature, humidity, rain, and wind using up to 27 nodes over 1.5 years. 2 nodes were installed for 1.5 years in a high-desert farm and 24 nodes in the UCLA Botanical Gardens for 3 months. Finally, a 12-node system was installed in a Bangladesh rice paddy for 2 weeks to measure nitrate, calcium, and phosphate (this experiment also described in [26]). These nodes used 433 MHz communication systems to share the data measured and a base station sent the data for offline analysis. The goal of the researchers for the system was portability and rapid deployment, focusing on a very different set of requirements than our system.

Werner-Allen [42] installed a wireless sensor network on a volcano in Ecuador, running 16 nodes for a 19 day test. Their system focused on scientific effectiveness, specifically the quality of the data and quantity measured allowing for delays in data gathering as long as correctly timestamped. The nodes measured seismic and acoustic data, transmitting to each other at 2.4 GHz and back to the base station through a single repeater node at 900 MHz. Detection of recordable signals did occur on the system, but no further data analysis within the network.

While the above systems do share some characteristics to the system and problem we describe, none envision the level of heterogeneity our system requires, the minimalistic number of sensors available for the extensive network area, the real-time need for the data, or the computational autonomy and complexity to perform the prediction operation.

Sensor Networks for Flood Detection

Previous work on sensor networks for flood detection is sparse with only two different examples discovered in the literature. Castillo-Effen [6] suggests an architecture for a system, but is unclear on the basin characteristics and no hardware details are suggested. Closest to our work is a paper by Hughes [18], describing a flood-predicting sensor network that uses Gumstix sensor nodes, which require significant power but allow for a Linux operating system to run on the node. As described, the system had been tested in the lab, but no field tests were performed by time of the paper. The planned field test would consist of 13 nodes along 1 km of the river. It is unclear what flood prediction model they are using and if it is currently running on their lab test system. Given lack of information on the flood prediction side, the known details of the hardware platform dismiss it as an immediate solution to the problem introduced here as it has limited geographic range, high cost, and large power requirements.

Current Operational Systems for Flood Detection

While not specifically sensor network installations, understanding the current operational systems helps clarify the problem space in which we are working. The lack of published information on operational flood systems makes generalizations difficult, but three systems seem to summarize the approaches currently taken.

One type involves a highly technical solution with significant resource support such as seen in the US. For this system, companies develop sensor, communication, and computation technology based on the ALERT protocol, which defines the data structure and polices of environmental monitoring systems [2]. The US Emergency Alert System provides communication of the alerts throughout the nation using television and radio channels by creating special technology and policies, requiring the installation of the technology in stations across the country along with weekly testing, and ensuring protocol compliance at all levels [11]. Implementation of specific systems trickles through each level of government: federal, state, and county. Given the large number of counties in the US, systems and policies do vary, but the majority rely on large numbers of personnel (some highly technical) and significant technical resources. Usually, counties implement the direct measurement system with help from the United States Geological Survey and create polices on how their county defines a disaster and evacuation procedures. Actual prediction usually depends on qualified hydrologists examining the data (thus removing measurement errors) and running it through a complicated physical model called the Sacramento model, which requires calibration of several unmeasurable parameters using years of historical data.

The other type is the system commonly seen in Central America, especially Honduras [3], relying on volunteers and limited technology. Sensors to measure river state include river level markings painted on bridges and water collecting rain gages. Volunteers read the river level and rain level (also emptying the rain gage) at several intervals during a day, radioing that information to a central office run by the government. In that office, a person listens to the radio, records the values in a book, and compares them to a defined policy whereby the river level measured corresponds to a color alert. This color alert is radioed to the head office of the government branch, which then decides on the need for an evacuation alert in that region and implements some form of emergency alert procedures. Overall this system relies on very little technology and extensive policies to warn communities, working best in small river basins where measurements indicate flooding in that area (as opposed to downstream of the measurement area).

A third solution exists in Bangladesh, a country regularly devastated by flooding due to its low sea level and large rivers. To combat this, the Danish Hydraulic Institute initially outfitted the country with local telemetry stations in 1995 and created a MIKE 11-based flood forecasting system [20]. However, this system experienced sustainability problems along with issues due to the fact that the headwaters of its major flood-causing rivers originate in India, creating complexities with monitoring. A solution to this was created by a global community of researchers and government institutions, collating all of the satellite information and forecasts generated by the US to provide short, medium, and long-term flood predictions of the major basins [15, 41]. A system called the Flood Forecasting and Warning Response System distributes the alert through reports submitted to various government agencies along with internet, e-mail, fax, telephone, radio and TV sources [8]. This takes advantage of the ubiquity of satellite information, which looks to provide input data for flood forecasting systems of the future [16]. The success of the system does rely on very regular satellite passes, still not common in all parts of the world, and a large amount of US resources, also not available everywhere.

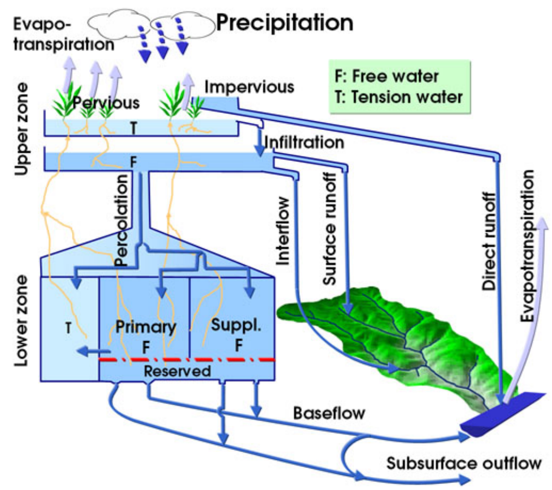

Figure 2. Example of Current Physically-based Model: Sacramento Soil Moisture Accounting (SAC-SMA)

Computation Requirements of Current Operational Flood Prediction Model

The current operational model works by modeling the different methods of rainfall surface runoff to determine how much water will enter the river, thus increasing the level. Called the Sacramento Soil Moisture Accounting model (SAC-SMA), it predicts runoff out to 12 hours based on rainfall over the area. It creates three different water compartments (see Figure 2 [7]): a zone describing the direct runoff from rain falling on impervious soils, a zone describing water flowing into the river after exceeding the soil moisture capacity of pervious soils, and a zone describing runoff occurring after soil moisture capacity is exceeded above water impervious regions [13]. The model describes each zone using several differential equations; all are too many to list, but just those governing one zone are:

$$\frac{dx_F(t)}{dt} = -p_f(t) - u_f(t) - e_F(t)$$
$$x_T(t) < x_T^0 \quad and \quad x_F(t) < x_F^0$$

$$\frac{dx_F(t)}{dt} = r_f(t) - p_f(t) - u_f(t)$$
$$x_T(t) = x_T^0 \quad and \quad x_F(t) < x_F^0$$

$$\frac{dx_F(t)}{dt} = 0; \quad x_T(t) = x_T^0 \quad and \quad x_F(t) = x_F^0$$
$$with$$
$$u_f(t) = \alpha x_F(t)$$
$$p_f(t) = \frac{x_F(t)}{x_F^0} p_0$$
$$e_F(t) = e_p(1 - \frac{x_T(t)}{x_T^0}); \quad x_F(t) > 0$$
$$e_F(t) = 0; \quad x_F(t) = 0$$

where e_F = evapotranspiration rate from upper zone free water storage, p_f = percolation rate to lower zone storage, r_f = excess flow rate from upper zone tension water to upper zone free water storage, u_f = interflow rate from upper zone free water storage, x_F = upper zone free water volume per unit area of catchment, x_F^0 = upper zone free water capacity, x_T = upper zone tension water volume per unit area of catchment, x_T^0 = upper zone tension water capacity, and α = rate of interflow production (in inverse time units).

A similar set of equations describes the other zones and several more complex equations describe the interconnection of these zones into a single surface runoff value. These clearly cannot easily run on a sensor network.

Determining the actual computation time of these equations is a bit uncertain. Two papers exist that outline some information on this although the information seems to differ between them. Experiments by Vrugt [40] on autocalibration methods using this model resulted in 25 minutes for calibration on a Pentium IV 3.4 GHz computer. This calibration appears to involve running the model. Other work by Ajami [1], also in the area of parameter autocalibration, includes Figure 3, displaying the run time and calibration time for a number of calibration methods where it appears that running the model requires on the order hours. However the paper doesn't specify numbers. Of the two papers, the first paper is more specific regarding computational numbers but much less detailed on procedure whereas the second is clearly using the SAC-SMA model over the same data set we also use. Either way, the information from both suggests that the model requires more computational power and time than available on a sensor network or in rural and developing country locations.

Additionally, these equations use 11 parameters, not all corresponding to actual physical, measurable quantities [25]. To calibrate these parameters and the model requires at least 8 years of rainfall and runoff data for calibration, ideally 8 years of further data for verification, detailed topographic maps, and hand-calibration by trained hydrologists [12]. The resulting model operates only on that basin; model creation for a different basin requires 8 years of calibration data for

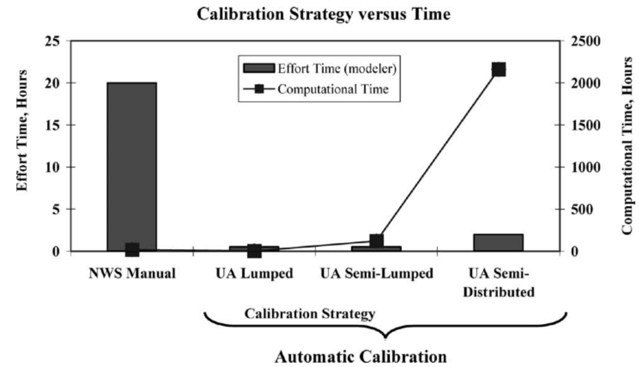

Figure 3. Computation and Calibration Time for SAC-SMA Model Using Manual and Automated Methods

the new river and expert hand-calibration. This again does not work for a sensor network nor regions where such data does not exist (and putting sensors in for 8 years to gather enough information is impractical). Simpler models with self-calibration and small data requirements that can run on a sensor network need to be created.

3 Prediction Model

Rainfall driven floods are the most common seasonal events[1]. They occur when the soil no longer has the capacity to absorb rainfall. Surface runoff and sub-surface discharge processes are the primary contributors to river flow and arise as a direct (but nonlinear) response of the basin to infiltration. To predict flooding, a model requires knowing how much rain falls and what the soil's time-dependent response to the rainfall will be.

Physically-based models deduce the runoff, discharge, and subsequent stream-flow using numerical implementations of the equations governing transport through the soil medium and the river channels [28, 32] (see Figure 2 for an example [7]). Modeling runoff processes using physics creates a challenge from a simulation point of view. The model requires details of the topography, soil composition, and land cover, along with meteorological conditions and hydrometeorological quantities such as soil moisture [19].

Ongoing work exists in the development of rainfall-runoff models, ranging from lumped to spatially-distributed variations [28, 32]. Although popular in academic research, the need for calibrating spatially-distributed models to individual basins, model sensitivity to basin conditions, and the tremendous computational burden involved in running them makes wide-spread application complicated and, in resource-strapped underdeveloped areas, nearly impossible.

In contrast, statistics gleaned from the observed record can lead to the development of low-dimensional distributed models, which are local in the sense of being valid for a given site. Such models intrinsically self-calibrate because the evolving record of observations allows them to adapt to the latest conditions. This creates portability from one locality to the next, from one season to the next, and from one climate regime to the next. Statistical models can yield low computa-

[1] Storm surges are not considered in this work.

tional complexity (as we propose), making them well suited for on-site and real-time implementations. Several of such statistical models running on different portions of the basin can collaborate in a distributed inference network to estimate flow at unobserved portions of the basin. Thus statistical models can also yield spatially-extended estimates. These benefits cut across the traditional justification for physically-based models and motivate their use in our work.

A growing body of evidence indicates that statistical models are useful in earth systems. This is true of flood prediction and, although the evidence [4, 30, 33, 34, 35] here is sparse, we can see mature applications in other areas. In particular, statistical-models have proven among the best in forecasting hurricane intensity (which presents similar challenges to flood forecasting) [9]. Indeed, statistical models are used for guidance in operational cyclone forecasting [22]. We believe that machine learning combined with physically-based features for estimation and inference can produce effective, highly-efficient, self-calibrating, distributed or lumped models of physical phenomena.

We can approach creating a model for predicting flow time-series using a set of predictors in many ways [4, 33]. A single global model (one with time invariant parameters) can operate either using functional (autoregression, neural networks, support vector machines) or distribution-oriented (Bayesian) means. However, global models, we think, may be insufficient due to the non-linearity of the processes we are modeling, requiring a more local (time varying) model to best capture that behavior. Localized modeling approaches include model trees, mixture of experts, and example-based methods. Ultimately, judging by other applications [10], we think that using local statistical models with simplified physical models may be the best stochastic modeling approach for flood prediction. We set out to encompass the model design space, beginning with simple models.

One of the simplest models is an autoregression model, appearing in various forms for hydrological modeling. This model assumes that a linear equation can describe the system behavior, weighting the past N observations of all relevant input variables taken at time $t - T_{lead}$ to produce a prediction of the output variable at time t. N describes the order of the model and is a parameter determined by the application. To determine the weighting factors, some amount of data designates the training set for the model (another application-defined parameter) and a simple inversion-multiply operation provides the coefficients from this data, which is the prediction model until recalibration occurs. In case the data provided contains local perturbations limiting the effectiveness of the coefficients, we can smooth the data using a low-pass filter.

While our experiments showed that an autoregression model of order 3 seems to work best over a broad range of prediction times, it is unclear if low-frequency time variability of predictors can enhance this performance. A natural way to test this, common in many signal processing problems, involves building autoregression models in wavelet space. We do this using multiscale differential features [27] and call it the features plus autoregression model (hereby called features). This captures information in the signals at several bandwidths (scale) and frequencies (order of derivative).

We developed two models using these techniques, with inputs of rainfall, temperature, and flow-history, and a single output, river flow. Using these models allowed us to gain insight into the system while working with limited data resources. These models, as we implemented them, self-calibrate, use very little training data (on the order of weeks), perform a very simple set of operations (an inversion-multiply and convolution), and require storing only the amount of data necessary for training. Considering the complexity of the current model as explained in Section 2, and the goal of computation on the sensor network, the use of such simple models is easily motivated. However, as observed (see testing section following), these forms of models tend to obscure rare behaviors such as floods by over-weighting the small-scale perturbations occurring in the longer timeseries of low flow observations. This suggests the use of local models, which we can construct more intelligently through our modeling work, thereby reducing the amount of data needed. Our future work will explore this direction, especially model-trees, defining coarse-to-fine organization of the model space in terms of diurnal, seasonal, annual, and decadal time cycles.

Testing

To analyze our algorithms, we use seven years of rainfall, temperature, and river flow data for the Blue River in Oklahoma [24, 38]. This river and data come from an on-going project called the Distributed Model Intercomparison Project (DMIP) run by the National Oceanic and Atmospheric Administration to compare hydrological models [28, 32]. The DMIP test provides more hydrometerological data for the models than our models uses, allows for calibration based on 1 year of data, and requires a 1 hour prediction of river level for assessment [28, 32]. We define three different criteria for determining the quality of our algorithms: the modified correlation coefficient (taken from DMIP [32]), the false positive rate of prediction, and the false negative rate of prediction. For the modified correlation coefficient, since we use the definition from DMIP, we can also compare our models to those listed as a reference of quality. False positive and negative detections provide a more common sense criteria as minimizing these increases the confidence of the end user in the system predictions.

We implemented the models as described above in Matlab, defining the training window as 4 weeks of data and recalibrating after each observation. Given these two parameter definitions, we analyzed the remaining parameters describing the models to determine optimal values. We discovered that an Order=3 autoregression model with smoothing using $\sigma^2 = \sqrt{2}$ suffices for a variety of prediction lead times. When adding in the feature computation, computing features at a single point using $scls = [1, \sqrt{2}, 2\sqrt{2}]$ outperforms other scale factors and feature convolution over a range of data points. In addition to the two models with these parameters, we computed predictions using two

	Modified Correlation Coefficient	False Positive Rate	False Negative Rate
DMIP LMP Uncalibrated	0.77	-	-
DMIP OHD Uncalibrated	0.71	-	-
DMIP Average Uncalibrated	0.58	-	-
DMIP LMP Calibrated	0.86	-	-
DMIP OHD Calibrated	0.86	-	-
DMIP Average Calibrated	0.70	-	-
Climatology 1 Hour	-0.009	17	13
Climatology 16 Hour	-0.009	17	13
Persistence 1 Hour	0.997	0	0
Persistence 16 Hour	0.76	2	10
Autoregression 1 Hour	0.98	3	0
Autoregression 16 Hour	0.63	23	0
Feature-Based 1 Hour	0.96	4	0
Feature-Based 16 Hour	0.63	36	3

Table 1. Overall Results: DMIP Best Cases and Four Linear Cases

naive approaches: climatology (or predicting the average of all previously seen flow observations at that hour and date in past years) and persistence (or assuming that the flow will stay at its currently observed value). Using all four approaches, for comparison with the DMIP results, we run our test over all seven years of data, computing the modified correlation coefficient at the end for the complete set. The test computes predictions for time periods of 1 hour (for comparison with DMIP) and 16 hours (as a more realistic prediction window).

Table 1 shows the overall results, comparing our two models and two naive approaches as well as to the best cases for calibrated and uncalibrated DMIP models. The DMIP models listed performed best in the project testing for the category of modified correlation coefficient for the Blue River. LMP offers the best results, but this model (the SAC-SMA model described earlier) demonstrates the current operational centralized method. For this reason, we include the OHD (or NWS Office of Hydrologic Development) model to demonstrate the best distributed model in current research. Examining the modified correlation coefficient, as the table demonstrates, persistence performs the best for 1 hour predictions with the autoregression model closely following. At 16 hours, persistence performs better than the average DMIP model, calibrated or uncalibrated, while autoregression outperforms the average uncalibrated DMIP result and performs only slightly worse than the average calibrated result.

Does this then mean the best approach to flood prediction is to assume the current state will hold? Examining the other two metrics for the 16 hour prediction claims otherwise. While persistence tracks the observed data closely in the aggregate sense over the seven years of data, it fails to detect 10 flood events. Autoregression, on the other hand, detects all flood events and adds 23 new ones. In this case, the failure to detect a flood is a significantly worse event than the false prediction of a flood. The first causes loss of life and property while the second reduces confidence in the model; the former cannot be remedied, but the latter can be alleviated through training and external verification methods. The low correlation value for autoregression, however, suggests that the large number of low flow data points saturates the model with examples of minor perturbations such that it mispredicts actual flood events. This suggests the need for more local modeling efforts and future work will examine the use of multiple models to reduce this error, although that may require at least one significant flood event to occur for the training set to properly anticipate the large increases.

In addition to multiple model creation, future work on this algorithm includes exploration of online definition of the uncertainty of the prediction to better indicate the potential of false positives and negatives, and increasing the timeframe of recomputing of coefficients from each hour to something more reasonable. Next we would like to do this on a sensor network platform with in-situ data collection and computation. The following sections describe our work on the sensor network platform to achieve this direction.

4 System Description

Complex system requirements constrain the system, forcing careful examination of the architecture and design of the network. We defined the following requirements for the overall system:

- Monitor events over large geographic regions of approximately 10000 km^2
- Measure a wide variety of variables contributing the occurrence of the event
- Survive long-term element exposure (on the order of years)
- Recover from node losses
- Power system for years
- Withstand the event of interest such as a river flood
- Minimize costs

Looking deeper at event prediction results in the following additional requirements:

- Detect and predict event of interest
- Handle significant computation requirements

- Allow for the distribution of the model across the network and ability of model to drive data collection

Due to the distance requirement, the inability to populate the entire area with sensors, and cost limitations, the network will consist of a small number of nodes, communicating over long distances on the order of 25 km with no recurring costs as that violates the cost requirement and limits the sustainability of the system. Yet the system also needs a variety of sensor measurements around any one location. This suggests a two-tiered architecture with multiple nodes, or a mini-network, at each location with a super-network connecting the locations. Other constraints affect the node design and system protocols and, taken with the above, result in Figure 4, showing an idealized overview of our system. We intend this system architecture as a generic architecture for model-based computation in sensor networks.

Based on the combination of mini-networks and long-range links, as demonstrated in Figure 4, the system communicates via two different methods. To cover the long-range communication links of approximately 25 km range, the system uses 144 MHz radios. In the US, this band covers the amateur radio operators and thereby provides many mobile radio systems for possible use. These systems focus on voice communication, however, so we designed a modem to allow data communication within this frequency over Kenwood TM271A VHF radios. The modem uses a MX614 Bell 202 compatible integrated circuit to convert 1200 baud serial signals to FSK modulated signals for radio transmission. This allows for data transmission across a cheap, long-range communication method without the recurring costs of a satellite or mobile telephone system. For short-range communication links within a 8 km range such as required by the mini-networks, the system operates within the 900 MHz band, using Aerocomm AC4790 off-shelf modules.

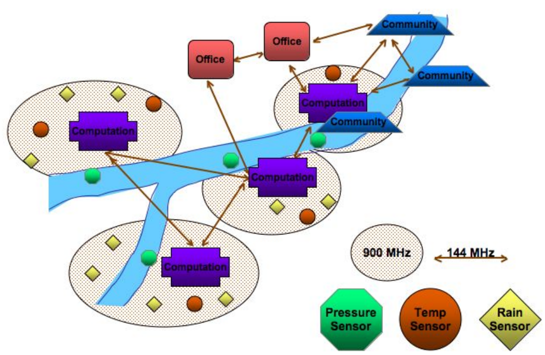

Figure 4. Idealized Sensor Network Consisting of Two Communication Tiers: 144 MHz Computation, Government Office, Community, and 900 MHz Sensor Nodes

In addition to two communication methods, the system consists of four different regimes of operation: sensing, computation, government office interface, and community interface. Each of these regimes requires a slightly different instantiation of hardware and software. Some regimes may overlap so each hardware/software instantiation needs to contain enough similarity to allow for this overlap within one node; we achieve this by starting from the same base system electronics and add an expansion daughter-board if a regime instantiation requires any extra hardware components to perform its specialized operations. In the following, the paper overviews each of these different nodes of the system, discussing the design decisions in more detail. The paper describes these sections within the context of a generic architecture for predictive environmental sensor networks (see Figure 5). Section 5 discusses all details specific to a flood prediction instantiation.

4.1 Base System

To allow easy interchange of node types and the combination of regimes within one node, all nodes begin with the same base electronics designed to provide for a variety of options. An ARM7TDMI-S microcontroller core, specifically the LPC2148 from NXP, provides the necessary computation power for the board. The LPC2148 operates at a high CPI (cycles per instruction) rate and provides many useful internal components including a real-time clock, 10-bit analog to digital converters, and I2C communication. However, it limits the number of physical serial ports to 2, which cannot cover all needed operations so we add a Xilinx CoolRunner-II CPLD to the system and configure it as a serial router. This allows for up to 8 different serial devices and all possible connections between them. In addition to limited serial ports, the small package size of the LPC2148 requires multiplexing of the operations on the digital input/output (I/O) pins thereby limiting the actual operations available. The base board sends all free pins to the daughter-board connectors allowing for a variety of operations and potential multiplexing of each I/O on specialized boards.

The base system also contains the AC4790 900 MHz wireless module as most nodes will require one. A mini-SD circuit and FRAM (Ferroelectric Random Access Memory) supply data and configuration storage. Finally, a charging circuit on the board allows photovoltaic charging of lithium-polymer batteries, which power the system at 3.7 V.

In addition to the base hardware constructed, the system runs a custom base software package developed in C using the WinARM libraries. This package consists of: serial libraries which hide the underlying CPLD serial router, a custom EFAT file system for SD-Card logging, sensor access libraries, AC4790 radio libraries, power regulation, and a scheduler system based on the real-time clock and internal timers.

4.2 Sensing

Sensing nodes measure the variables needed to detect and predict the event of interest. In addition to the measurements, the nodes log the raw data, compute data statistics over each hour and inter-transmission time period, analyze data for indications of incoming events and potential sensor failures, and scan for faults based on current operational status and

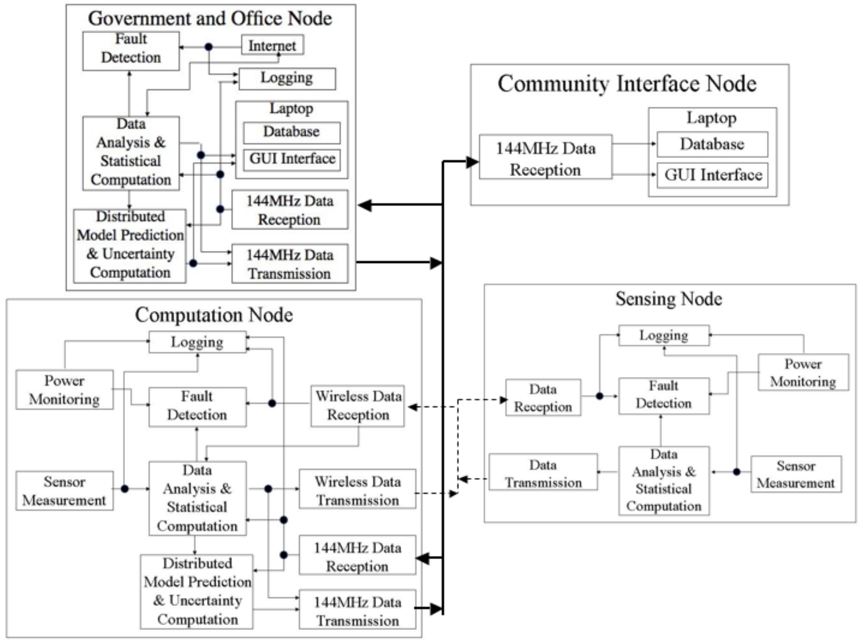

Figure 5. Generic Predictive Environmental Sensor Network Architecture Consisting of Sensing, Computation, Government Office, and Community Interface Nodes

power. These nodes also regularly transmit the computed data to the computation nodes via the underlying AC4790 protocols internal to the radio module. For these types of systems, a node may monitor multiple sensors and multiple sensor types requiring hardware and software support for both scenarios. Additionally, these nodes survive out in the field for many months so cannot use much power and should self-monitor for the warning signs of node failure.

Our nodes accomplish this through a daughter-board attached to the base system that expands the available I/Os through an I2C integrated circuit and creates several ports for sensors ranging from resistive to interrupt to voltage. In case the sensor requires a more complicated interface, we include RS485 and RS232 circuits for external communication to sensors. The nodes also measure charge current supplied by small photovoltaic panels to determine the power available, a helpful indicator of the remaining life of the node.

These nodes communicate via 900 MHz with all nodes in immediate range, creating mini-networks of sensors within the greater system (the combination of few nodes and large areas ensures each node only joins one mini-network). Nodes regularly transmit data measurements to all other nodes, providing a way to monitor each other through examining the data for errors and immediately noticing the failure of any node not transmitting within an appropriate window. To both capture the event of interest and monitor the state of the sensor, the system modifies the measurement rate and transmission rate based on criteria supplied by the model through the computation nodes as well as allowing the computation nodes to request measurements outside of the regular measurement schedule. Repeated measurements of odd values such as the maximum possible value of the sensor or rapid rates of change trigger a warning that the sensor may not function anymore, which the node then transmits via the 900 MHz network to other nodes nearby.

4.3 Computation

Computation nodes connect the mini-networks of sensors and provide the computational muscle (see Figure 6). These nodes run the algorithms to detect and predict the event of interest, focusing on their local region. As data arrives from nearby sensors, the node combines the data from each sensor, including information from other mini-networks' related sensors (such as upstream sensors in the case of river flooding), and examines the correctness of the data using information received from the government office interface nodes. The node runs the data through the distributed model, computing the prediction along with the uncertainty of that prediction over the time window specified. To reduce the uncertainty, the computation node requests additional data from the sensing nodes and modifies the parameters affecting the measurement schedule for the sensing nodes.

To achieve this, nodes communicate both via the 900 MHz network and to each other via the 144 MHz network. Because these nodes use the 144 MHz network, they require a different daughter-board from the sensing nodes. This board includes the modem and power switching circuitry for radio control while also including the various sensing ports and RS485 communication. With the second radio and differing daughter-board requirements, this node add several libraries to the existing base software structure, most specifically wrapper functions to use the modem at a higher level than raw serial data.

The system cannot use lithium-polymer batteries and small photovoltaic panels due to the power requirements of the radio when transmitting so this system uses leadacid batteries along with 6 W photovoltaic panels. To power the radio, the system needs 12 V instead of the base board requirement of 3.7 V so we use the daughter-board to power the base board for these nodes. To ensure radio communication over the 25 km range, these systems ideally need antennas located at least 5 m high in the air, requiring antenna towers for the system with added benefits of ensuring proper sunlight for the photovoltaic panels and theft protection for the system (see Figure 7).

4.4 Government Office Interface

These nodes provide a user interface to the network. This interface will focus on the government and relief agencies who will maintain the system, providing data and predictions regarding the event of interest along with detailed information to monitor the system and display those nodes no longer functioning.

The office nodes communicate via 144 MHz with the computation nodes to provide any external requests or data and receive all of the existing network data. This allows the office nodes to perform prediction and detection in a similar manner to the computation nodes, but with a focus on the entire geographic area instead of a local region, creating a secondary prediction computation as a redundant measure for the local computation nodes. Additionally, with the possibility of internet access in an office, these nodes could provide external verification for data through online information, using satellite and other remote data available to verify the computation results, checking for errors such as a flood prediction when no rain has fallen or a forest fire during a rain storm.

4.5 Community Interface

These nodes provide an interface to the communities interested in the detection and prediction of the events. In case of disasters, this would allow time to evacuate, but the system could also provide information such as environmental monitoring, agricultural information, or animal predator detection. The interface displays the known state of the geographic area, event predictions, event detections, and post-event monitoring. To avoid confusion, the interface does not supply detailed information regarding the network, such as node status or the data underlying the computations. Based on the location of the communities within the network, these nodes may also double as any of the other node types.

Figure 7. 5 Meter Antenna Tower for Computation Nodes

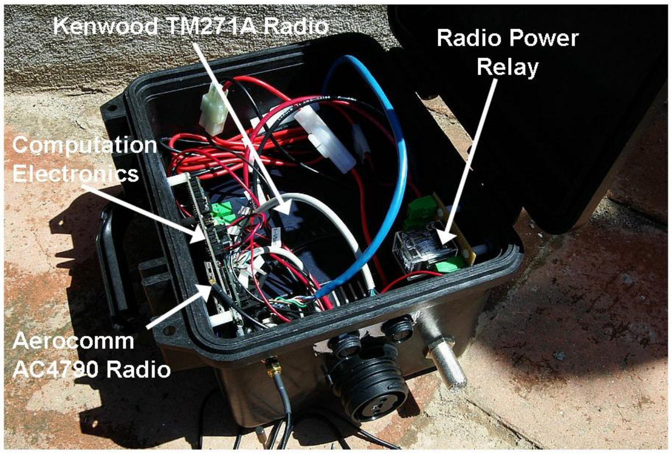

Figure 6. Computation Node

5 Installation and Results

We implemented this design, focusing on the application of river flooding. We have traveled nine times to Honduras over the course of this project with the goal of understanding deployment issues and testing the system components. Additionally, we have a local test site in the Charles River basin, specifically the region of the river located around Dover, Massachusetts. This site allows us both to quickly identify any system issues without a trip to Honduras and to run longer tests, collecting data for our prediction modeling work and discovering any long range system issues.

5.1 Implementation

For the particular application of river flooding, the sensors consist of precipitation, air temperature, and water pressure. Precipitation sensors measure using reed magnetic switches, which cause an interrupt after every 1 mm of rainfall. Temperature sensors measure resistively, modifying an ADC level, which translates into a temperature after calibration. While the sensor types differ, the system construction is similar and requires nothing additional to the sensing nodes

Figure 8. Precipitation Sensor Node Consisting of Electronics, Sensor and Photovoltaic Board

described in 4.2. We placed the electronics within Otter boxes to ensure protection from the elements and added Bulgin connectors for the sensor, antenna, and photovoltaic boards (see Figure 8).

Measuring water pressure allows us to compute the water level. While our simulation work with the Blue River data uses river flow since that is the data available from the USGS, measuring flow requires several sensors to get a cross-sectional profile of the river in order to understand flow at a single location on the river. Level, however, requires only one measurement to understand the state of the river yet relates to flow through easily defined and understood curves (the USGS actually measures level as well and performs this conversion prior to posting the data online). Therefore the two values are interchangeable and we use river level for our discussion of the modeling results from the Dover site data. To perform the level measurement requires a special underwater installation. In order to maintain solar power and wireless communication, we developed an external pressure sensor box (see Figure 9) to communicate via RS485 with the sensing node. Our pressure board consists of another LPC2148 microcontroller, RS485 interface, and instrumentation amplifier. The LPC2148 is much more powerful than necessary, but allows us to maintain a consistent software system. We complete the box by attaching a Honeywell 24PCDFA6A pressure sensor, and output the RS485 lines along with power and ground through a Seacon underwater connector. Honeywell's pressure sensor measures 13.8-206.8 MPa of water pressure directly instead of the more typical air pressure, allowing us to bypass the use of extensive tubing to ensure no water touches the sensor.

5.2 Dover Field Test

We used the Dover site to test the long-term behavior of the system, specifically the sensing and 900 MHz communication. The installation consists of 3 distinct sensor nodes: 1 precipitation, 1 temperature, and 1 pressure sensor.

We installed the three nodes within 900 MHz radio communication range at the locations shown in Figure 10. The pressure sensor we placed within a USGS sensing

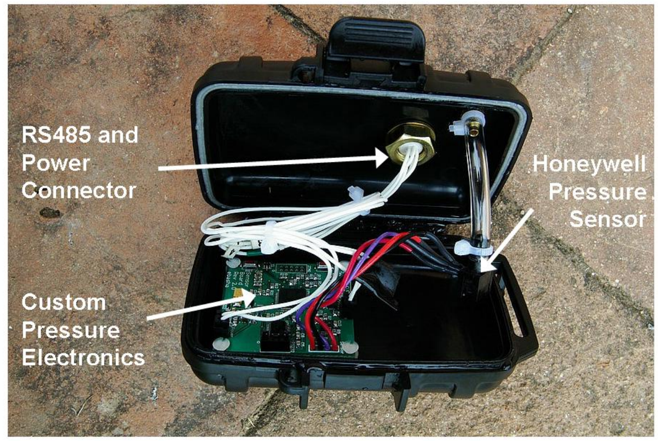

Figure 9. Pressure Sensor Box to Communicate with Sensor Node

station, using their concrete shed as a base for our system. The other sensors we located across the river, with the precipitation sensor across from the pressure sensor and the temperature sensor upstream of both.

Figure 10. Dover Site with Three Sensors

With the system we gathered 5 weeks worth of data before ending the field experiment. Figure 11 shows the hourly precipitation, pressure, and temperature measured by the nodes over the complete experiment. While no flood occurred during this time period, we do see a variety of interesting behaviors such as a large amount of rainfall occurring at hour 251 and a period of no change occurring right before from hours 90 through 250. We then run this data through three of the models discussed in Section 3 for three prediction times and compare use of one versus two weeks of training data. The results shown in Table 2 demonstrate the expected conclusion that more training data improves the prediction, which we can explain by noting that this benefit derives from the rainfall now present within

	Modified Correlation Coefficient	
	One Week Training Data	Two Weeks Training Data
Persistence 1 Hour	0.992	0.995
Persistence 8 Hour	0.901	0.925
Persistence 16 Hour	0.749	0.797
Autoregression 1 Hour	0.957	0.978
Autoregression 8 Hour	0.743	0.916
Autoregression 16 Hour	0.422	0.800
Features 1 Hour	0.927	0.970
Features 8 Hour	0.613	0.902
Features 16 Hour	0.374	0.779

Table 2. Comparison of Model Results for Dover Data

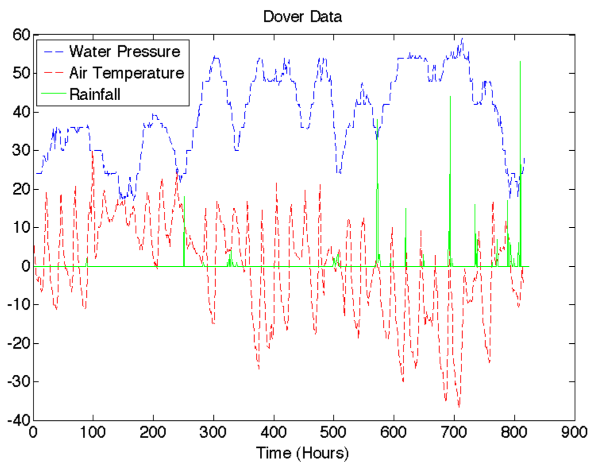

Figure 11. Data Collected from Dover Test Site

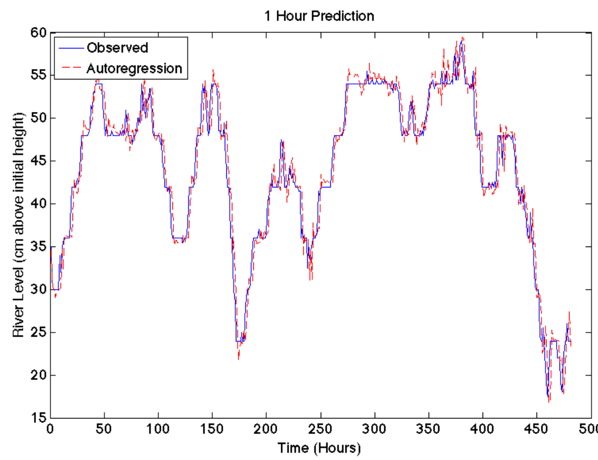

Figure 12. 1 Hour Forecast of Dover Data

the two weeks of training data where, at one week, no rain had occurred. Additionally, comparing to the average from DMIP's calibrated results to define reasonable (see Table 1), all three models function reasonably, outperforming DMIP out to 16 hour predictions with two weeks of training data and out to 8 hours with only one week. We also see an interesting result occurring for the 16 hour prediction where autoregression performs slightly better than persistence. Figures 12 and 13 demonstrate the performance of the autoregression model for prediction hours 1 and 16 with two weeks of training data. At 1 hour, the system tracks the observations quite well with only a minor lag. At 16 hours, the system lags the observations more noticeably with a jaggedness suggesting an attempt to compensate for the chaotic behavior of the data and two overshoots corresponding to the two large rainfall events captured in the 5 weeks. The apparent phase delay of the prediction results from the order of the autoregression; a larger order would remove this delay, but results in a worse correlation coefficient.

Overall, the field experiment was a useful indicator of the potential to predict flooding using statistical methods as we suggest, demonstrating reasonable results up to 16 hour prediction windows with only two weeks of training data. Larger flow variations would help further demonstrate this, although we would prefer to avoid that sort of flooding in Massachusetts. Additionally, connecting the Dover site to MIT using the communication nodes would provide a good test of a larger portion of the system and, given that we can use the amateur radio band, should occur over the next few months.

5.3 Honduras Field Tests

During our trips to Honduras, we explored various problems associated with the system deployment, especially infrastructure issues, and tested those components associated with infrastructure. A local non-governmental organization, the Centro Técnico San Alonso Rodríguez (CTSAR), initiated the project in the river basin and aids our work on this problem.

On the communication side, we verified the usability of the 144 MHz radios. We tested the various ranges necessary for the system, ensuring that they can communicate over those ranges. To communicate at these ranges reliably, the radio antennas need line-of-sight high in the air, which requires antenna towers and limits the ability to test this portion of the system in the US. With CTSAR help, we arranged access to land and built 5 meter antenna towers at two river sites where we plan to install water level sensors for 144 MHz radio communication along with 10 meter towers at the CTSAR office and the government emergency

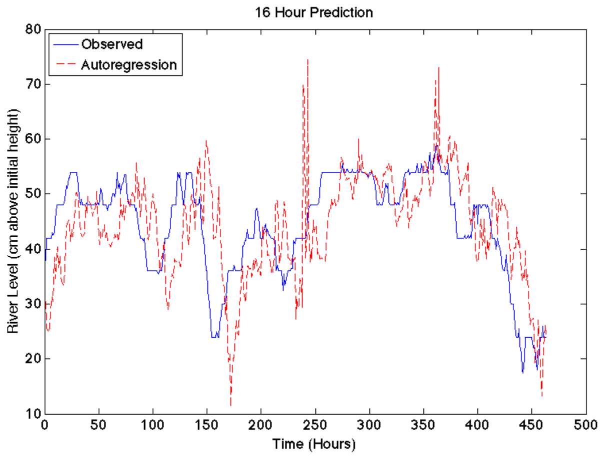

Figure 13. 16 Hour Forecast of Dover Data

Figure 14. Installation of 10 Meter Tower at Office

management office in Tocoa (see Figure 14). With these towers, we verified both the communication range and the ability of our modems to communicate data over this range. Thanks to hurricanes in 2005, we also proved that the towers and antennas will survive hurricane force winds.

At the offices, in addition to the towers, we worked to design and install secondary solar power systems. We would prefer to use grid power if it exists, but need solar power backup for the daily fluctuations of that system along with the major outages associated with disasters. CTSAR worked with a local company to purchase panels, batteries, and a charge controller. We added an off-the-shelf inverter, a power strip, and very simple custom electronics to switch to solar at the absence of grid power. We installed these systems at both offices and are running long-term usage tests. At the government office, we also installed a permanent radio and laptop for development of that interface, using it both for longer term radio tests and exploring issues with the interface.

Another area of testing has been the water measuring

Figure 15. Installation of Water Level Prototype

system. We have created five different prototypes of this system installing each for several months in Honduras (see Figure 15). Through these prototypes, we settled on measuring water pressure as a method of obtaining river level. Other options such as resistive water level sensors were rejected due to corrosion issues, while ultrasonic sensors were rejected due to the indirect nature of the measurement along with reduced ability in high winds. These prototypes allowed us to understand the complexities of installing something in a flooding river since box movement reduces the efficacy of the measurement. Structures must hold the sensor in a fixed spot while ensuring the system does not sink in the soft ground of the river and that it is retrievable for maintenance. We developed two different solutions allowing us to install the system on a bridge for greater reliability and also in the middle of the river when the situation necessitates.

All of this work has helped create the infrastructure necessary to achieve our goal of a demonstration system. Next, we plan to perform a long-term test of the communication system in Massachusetts with the algorithm distributed on the system and then install everything in Honduras in time for hurricane season.

6 Conclusion

We described in this paper an architecture for predictive environmental sensor networks over large geographic areas. These systems are node-limited due to region size and cost constraints. They also have significant system requirements due to the outdoor installation, destructive events, and long operational lifetime.

Our sensor network solution addresses these requirements, consisting of two communication tiers, four node types, and support for a variety of different sensor types. We focused on the event of river flooding, specifically in Honduras. The paper describes our early work on the flood prediction algorithm that will drive the system and the implementation of the sensor network architecture for this application. Locally, we installed 3 nodes on the upper Charles river at Dover and gathered 5 weeks of data, which we ran through our prediction algorithm, demonstrating both our system functionality and algorithmic functionality. In

Honduras, we built several key pieces of infrastructure, including the radio antenna towers, and tested several system components.

Future work involves adding the flood prediction algorithm to the network and connecting the Dover sensors through the computation nodes to MIT. This will provide a sufficient enough test for us comfortably plan a permanent system installation in Honduras, a further test of the practicality and robustness of the system.

Acknowledgments

This work was supported in part by NSF and by the Microsoft Digital Inclusion program. We are very grateful for this support.

We would also like to thank the following people for their help in the construction of the sensor nodes: Alex Bahr, Carrick Detweiler, Andrea Llenos, and Iuliu Vasilescu.

7 References

[1] N. K. Ajami, H. Gupta, T. Wagener, and S. Sorooshian. Calibration of a semi-distributed hydrologic model for streamflow estimation along a river system. *Journal of Hydrology*, 298:112–135, October 2004.

[2] ALERT Systems Organization. Alert history. `http://www.alertsystems.org`.

[3] E. Basha. Interview with COPECO officials in La Masica, Honduras, January 2004.

[4] A. Brath, A. Montanari, and E. Toth. Neural networks and non-parametric methods for improving real-time flood forecasting through conceptual hydrological models. *Hydrology and Earth System Sciences*, 6(4):627–639, 2002.

[5] Z. Butler, P. Corke, R. Peterson, and D. Rus. Dynamic virtual fences for controlling cows. In *Proceedings of the Ninth International Symposium on Experimental Robotics*. ISER, June 2004.

[6] M. Castillo-Effen, D. H. Quintela, R. Jordan, W. Westhoff, and W. Moreno. Wireless sensor networks for flash-flood alerting. In *Proceedings of the Fifth IEEE International Caracas Conference on Devices, Circuits and Systems*, pages 142–146. IEEE, Nov 2004.

[7] Center for Hydrometeorology & Remote Sensing, University of California, Irvine. Hydrologic predictions - on-going activities. `http://chrs.web.uci.edu/research/hydrologic_predictions/activities07.html`.

[8] R. Chowdhury. Consensus seasonal flood forecasts and warning response system (FFWRS): an alternate for nonstructural flood management in Bangladesh. *Environmental Management*, 35:716–725, May 27 2005.

[9] M. DeMaria and J. Kaplan. An updated statistical hurricane intensity prediction scheme (ships) for the atlantic and eastern north pacific basins. *Weather and Forecasting*, 14(3):326–337, 1999.

[10] K. Emanuel, S. Ravela, E. Vivant, and C. Risi. A statistical deterministic approach to hurricane risk assessment. *Bulletin of the American Meteorological Society*, 87:299–310, March 2006.

[11] Federal Communications Commission: Public Safety and Homeland Security Bureau. Emergency alert system. `http://www.fcc.gov/pshs/eas/`.

[12] B. D. Finnerty, M. B. Smith, D.-J. Seo, V. Koren, and G. E. Moglen. Space-time scale sensitivity of the sacramento model to radar-gage precipitation inputs. *Journal of Hydrology*, 203:21–38, December 1997.

[13] K. P. Georgakakos. Analytical results for operational flash flood guidance. *Journal of Hydrology*, 317:81–103, February 2006.

[14] R. Guy, B. Greenstein, J. Hicks, R. Kapur, N. Ramanathan, T. Schoellhammer, T. Stathapoulos, K. Weeks, K. Chang, L. Girod, and D. Estrin. Experiences with the extensible sensing system ESS. In *Proceedings of CENS Technical Report #60*. CENS, March 2006.

[15] T. M. Hopson and P. J. Webster. Operational short-term flood forecasting for Bangaldesh: application of ECMWF ensemble precipitation forecasts. 8, 2006.

[16] F. Hossain, N. Katiyar, Y. Hong, and A. Wolf. The emerging role of satellite rainfall data in improving the hydro-political situation of flood monitoring in the under-developed regions of the world. *Journal of Natural Hazards*, 43:199–210, March 9 2007.

[17] W. Hu, V. N. Tran, N. Bulusu, C. T. Chou, S. Jha, and A. Taylor. The design and evaluation of a hybrid sensor network for cane-toad monitoring. In *IPSN '05: Proceedings of the fourth international symposium on Information Processing in Sensor Networks*, pages 503–508. IEEE, April 2005.

[18] D. Hughes, P. Greenwood, G. Blair, G. Coulson, F. Pappenberger, P. Smith, and K. Beven. An intelligent and adaptable grid-based flood monitoring and warning system. In *Proceedings of the 5th UK eScience All Hands Meeting*, 2006.

[19] V. Y. Ivanov, E. R. Vivoni, R. L. Bras, and D. Entekhabi. Preserving high-resolution surface and rainfall data in operational-scale basin hydrology: a fully-distributed physically-based approach. *Journal of Hydrology*, 298:80–111, October 2004.

[20] G. H. Jørgensen and J. Høst-Madsen. Development of a flood forecasting system in Bangaldesh. In *Operational Water Management Conference*, 1997.

[21] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. In *ASPLOS '02: Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 96–107, New York, NY, USA, 2002.

ACM Press.

[22] T. N. Krishnamurti, C. M. Kishtawal, Z. Zhang, T. LaRow, D. Bachiochi, E. Williford, S. Gadgil, and S. Surendran. Multimodel ensemble forecasts for weather and seasonal climate. *Journal of Climate*, 13(23):4196–4216, 2000.

[23] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97, New York, NY, USA, 2002. ACM Press.

[24] N. Oceanic and A. A. N. W. Service. Distributed model intercomparison project. `http://www.nws.noaa.gov/oh/hrl/dmip/`.

[25] N. Oceanic and A. A. N. W. Service. Hl distributed modeling research. `http://www.nws.noaa.gov/oh/hrl/distmodel/abstracts.htm#abstract_7`.

[26] N. Ramanathan, L. Balzano, D. Estrin, M. Hansen, T. Harmon, J. Jay, W. Kaiser, and G. Sukhatme. Designing wireless sensor networks as a shared resource for sustainable development. In *ICTD '06: Proceedings of the International Conference on Information and Communication Technologies and Development*, pages 256–265, May 2006.

[27] S. Ravela. *On Multi-scale Differential Features and their Representations for Recognition and Retrieval*. PhD thesis, Department of Computer Science, University of Massachusetts at Amherst, 2002.

[28] S. Reed, V. Koren, M. Smith, Z. Zhang, F. Moreda, D.-J. Seo, , and D. Participants. Overall distributed model intercomparison project results. *Journal of Hydrology*, 298:27–60, October 2004.

[29] L. Selavo, A. Wood, Q. Cao, T. Sookoor, H. Liu, A. Srinivasan, Y. Wu, W. Kang, J. Stankovic, D. Young, and J. Porter. Luster: wireless sensor network for environmental research. In *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 103–116, New York, NY, USA, 2007. ACM.

[30] R. R. Shrestha and F. Nestmann. River water level prediction using physically based and data driven models. In *Zerger, A. and Argent, R.M. (eds) MODSIM 2005 International Congress on Modelling and Simulation*, pages 1894–1900. Modelling and Simulation Society of Australia and New Zealand, December 2005.

[31] P. Sikka, P. Corke, P. Valencia, C. Crossman, D. Swain, and G. Bishop-Hurley. Wireless adhoc sensor and actuator networks on the farm. In *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*, pages 492–499, New York, NY, USA, 2006. ACM Press.

[32] M. B. Smith, D.-J. Seo, V. I. Koren, S. M. Reed, Z. Zhang, Q. Duan, F. Moreda, and S. Cong. The distributed model intercomparison project (dmip): motivation and experiment design. *Journal of Hydrology*, 298:4–26, October 2004.

[33] D. P. Solomatine, M. Maskey, and D. L. Shrestha. Instance-based learning compared to other data-driven methods in hydrological forecasting. *Hydrological Processes*, 22:275–287, 2008.

[34] D. P. Solomatine and M. B. Siek. Modular learning models in forecasting natural phenomena. *Neural Networks*, 19(2):215–224, 2006.

[35] D. P. Solomatine and Y. Xue. M5 model trees and neural networks: Application to flood forecasting in the upper reach of the Huai River in China. *Journal of Hydrologic Engineering*, 9(6):491–501, November/December 2004.

[36] B. Thorstensen, T. Syversen, T.-A. Bjørnvold, and T. Walseth. Electronic shepherd - a low-cost, low-bandwidth, wireless network system. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 245–255, New York, NY, USA, 2004. ACM Press.

[37] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong. A macroscope in the redwoods. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 51–63, New York, NY, USA, 2005. ACM Press.

[38] tRIBS Development Team. tribs hydromet data. `http://www.ees.nmt.edu/vivoni/tribs/weather.html`.

[39] United States Geological Survey. USGS hurricane mitch program. `http://mitchnts1.cr.usgs.gov/overview.html`.

[40] J. A. Vrugt, B. O. Nualláin, B. A. Robinson, W. Bouten, S. C. Dekker, and P. M. Sloot. Application of parallel computing to stocahstic parameter estimation in environmental models. *Computers and Geosciences*, 32:1139–1155, October 2006.

[41] P. J. Webster and R. Grossman. Forecasting river discharge into Bangladesh on short, medium and long time scales. *Climate Forecasting Applications in Bangladesh*, January 2003. Online at `http://cfab.eas.gatech.edu/cfab/Documents/InfoSheets/CFAB_forecast.pdf`.

[42] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. Fidelity and yield in a volcano monitoring sensor network. In *OSDI '06: Proceedings of the 7th symposium on Operating systems design and implementation*, pages 381–396, Berkeley, CA, USA, 2006. USENIX Association.

[43] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi. Hardware design experiences in ZebraNet. In *SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pages 227–

238, New York, NY, USA, 2004. ACM Press.