

Model-Based Monitoring for Early Warning Flood Detection

Elizabeth A. Basha
Computer Science and
Artificial Intelligence
Laboratory
Massachusetts Institute of
Technology
Cambridge, MA, USA
e_basha@mit.edu

Sai Ravela
Earth Atmospheric and
Planetary Science
Massachusetts Institute of
Technology
Cambridge, MA, USA
ravela@mit.edu

Daniela Rus
Computer Science and
Artificial Intelligence
Laboratory
Massachusetts Institute of
Technology
Cambridge, MA, USA
rus@csail.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 work on a centralized form of 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 km² using an unique heterogeneous communication structure to provide real-time sensed data, incorporating self-monitoring for failure, and adapting measurement schedules to capture events of interest.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—Wireless communication;
C.3 [Special-Purpose and Application-based Systems]: Real-time and embedded systems;
I.6.5 [Simulation and Modeling]: Model development;
J.2 [Physical Sciences and Engineering]: Earth and atmospheric sciences

General Terms

Design, Experimentation, Measurement

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SenSys'08, November 5–7, 2008, Raleigh, North Carolina, USA.
Copyright 2008 ACM 978-1-59593-990-6/08/11 ...\$5.00.

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 [25], zebras [23, 45], and a redwood tree [40] exemplifies the usefulness of these systems. We are interested in developing and deploying 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 the case with floods or hurricanes, this further complicates the requirements. This system must withstand the event, which usually poses a 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. Our specific application is river flooding with a main deployment target for the system of rural and developing regions. With this application in mind, we can further define the system requirements. The system must withstand river flooding and the severe storms causing the floods, monitor and communicate over a 10000 km² river basin, predict flooding autonomously, and limit costs allowing feasible implementation of the system in a developing country.

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. Computing models on a network suggests 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 data drives when and what is measured, how often the model computes predictions, and

when the system communicates predictions; project goals not fully achieved in this paper.

In this paper we present a sensor network architecture and instantiation for developing regions, a statistical modeling algorithm for river flood prediction, and evaluations of both. The flood prediction algorithm is based on a regression model that performs significantly better than current hydrology research versions at 1 hour predictions and nearly as good as these same research versions when our model predicts 24 hours and those models predict 1 hour. For prototyping and validation purposes, we tested this model using 7 years of data from the Blue River in Oklahoma. We split the 7 years into 1 year of training data and 6 non-intersecting years of verification data. Figure 1 demonstrates our modeling results for a 24 hour prediction on a portion of the verification data set, showing how our predicted peaks coincide with the observed reality. The implementation used for this paper is centralized computation of sensor data from multiple nodes.

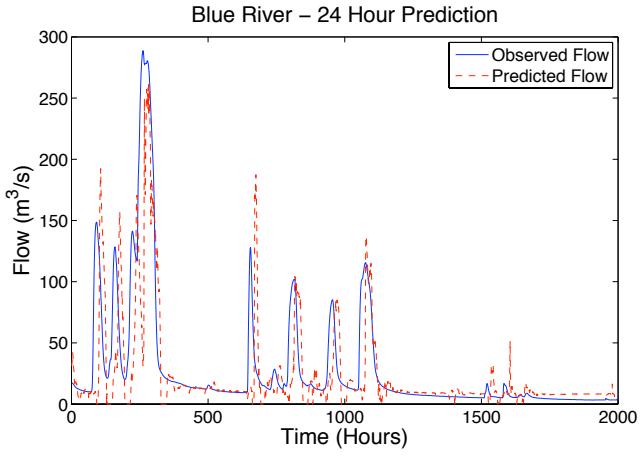


Figure 1: Blue River 24 Hour Prediction Results on Verification Data Set

We built a sensor network for flood prediction that consists of 9 nodes. An instantiation of the sensor network consisting of 3 nodes was installed on the Charles River at Dover, Massachusetts and gathered 5 weeks of data. This data was later run through our model. Chosen primarily for practicality reasons and speed of prototyping, the Charles encompasses a basin of 1000 km^2 , only one order of magnitude less than our proposed basin, and provides support from the United States Geological Survey (USGS) along with verification of our measurements through their sensors. A simplified version of the 9-node sensor network was also deployed on the Aguán River basin in northern Honduras, which we use as our test basin as this region provides a representative case of a developing region with serious flood problems (see Figure 2). This installation was used to test the sensing, networking, deployment, and maintenance issues in rural Honduras. We plan a followup deployment that runs the flood prediction algorithm in-situ during January 2009. This paper describes our experiments in these two deployments.



Figure 2: Aftermath of Hurricane Mitch in 1998 in Northern Honduras

The rest of the paper is organized as follows. Section 2 presents previous work. Section 3 describes the prediction algorithm. Section 4 discusses issues related to designing and building a sensor network capable of running the flood prediction algorithm from Section 3 in-situ. Section 5 discusses the field installation and experimental results.

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 include cattle ranch monitoring [34], cattle control [8, 31], sheep monitoring [39], zebra herd monitoring [23, 45], seabird nests [25], and frog vocalizations [19]. Of greater relevance is work in environment monitoring where several projects have implemented related systems.

Tolle [40] 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 5 minutes and transmitted it via GPRS modem to an external computer. All analysis was performed off-line after the test period.

Selavo [32] 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 [16] 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. The project installed 2 nodes 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 [29]). 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 [44] 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 occurred 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 necessary 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 [9] 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 [20], 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 power requirements that may be, in the long-term, unsustainable.

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 policies of environmental monitoring systems [3]. The US Emergency Alert System provides communication of the alerts throughout the nation us-

ing 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 [13]. 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 USGS and create policies 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 described below.

Another type is the system commonly seen in Central America, especially Honduras [4], relying on volunteers and limited technology. Volunteers read the river level off of markings painted on bridges and the rain level from water collecting gages (also emptying the 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).

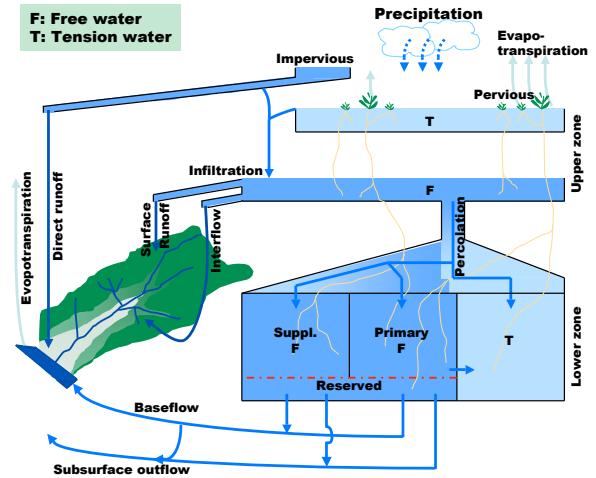


Figure 3: Example of Current Physically-based Model: Sacramento Soil Moisture Accounting (SAC-SMA)

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 [22]. 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 [17, 43]. A system called the Flood Forecasting and Warning Response System distributes the alert through reports submitted to various government agencies along with a variety of public media sources [11]. This takes advantage of the ubiquity of satellite information, which looks to provide input data for flood forecasting systems of the future [18]. 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.

Computation Requirements of Current Operational Flood Prediction Model We discuss flood prediction models in general in Section 3; however of most relevance to this work are current operational models, especially that used on our prototype data sets. In the US, 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 3 [10]): 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 [15]. The model describes each zone using several differential equations along with several more complex equations describing 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 difficult as examining two papers that outline some information on this provides different information. Experiments by Vrugt [42] 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 [2], also in the area of parameter autocalibration includes a figure displaying the run time and calibration time for a number of calibration methods where it appears that running the model requires on the order of 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.

Additionally, these equations use 11 parameters, not all corresponding to actual physical, measurable quantities [27]. 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 [14]. The resulting model operates only on that basin; model creation for a different basin requires 8 years of calibration

data for 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).

3. PREDICTION MODEL

In this section we describe a model and an efficient algorithm for flood prediction that uses data from the nodes of a spatially distributed sensor network. This approach is computationally leaner than conventional approaches to flood modeling and prediction, utilizing real-time data from multiple sensor nodes.

Rainfall driven floods¹ are the most common seasonal events. They occur when the soil no longer has the capacity to absorb rainfall. To predict flooding, a model requires knowing how much rain falls and what the soil's time-dependent response to the rainfall will be.

Current physically-based models deduce the rainfall-induced above and below ground flows of water into the river, and subsequent stream flow using numerical implementations of the equations governing transport through the soil and the river channels [30, 35] (see Figure 3 for an example [10]). Modeling these 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 [21].

In the development of these rainfall-runoff models, ongoing work covers a range of models from lumped to spatially-distributed variations [30, 35]. 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 computational complexity, 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 [6, 7, 33, 36, 37, 38] 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) [12] and are used for guidance in operational cyclone forecasting [24].

¹Storm surges are not considered in this work.

The simplest set of statistical models is that of linear regressions [5], appearing in various forms for hydrological modeling [6]. Within this category, multiple linear regression models assume that a linear equation can describe the system behavior, weighting the past N observations of all relevant input variables taken at time t to produce a prediction of the output variable at time $t + T_L$. We can also consider adding past predictions and/or past prediction errors as inputs within this model. To determine the weighting factors, some amount of data is designated as the training set for the model, defined here as the data seen in time T_T (an application-defined parameter), and a simple inversion-multiply operation provides the coefficients from this data, which is the prediction model until recalibration occurs, defined as a time window of length T_R . In case the data provided contains local perturbations limiting the

Algorithm 1 Flood Prediction Algorithm

```

1:  $\phi$  : past flow
2:  $\theta$  : air temperature
3:  $\rho$  : rainfall
4:  $N$  : # past flow values used
5:  $Q$  : # rainfall values used
6:  $P$  : # air temperature values used
7:  $Y$  : predicted flow
8:  $e$  : prediction error
9:  $T_T$  : training time window
10:  $T_L$  : prediction lead time
11:  $T_R$  : recalibration time window
12:
13:  $T_{TL} = T_T - T_L$ ;                                ▷ Compute initial coefficients and prediction
14:  $\phi_N \leftarrow [\phi(1 : T_{TL} - N), \dots, \phi(1 + N : T_{TL})]$ 
15:  $\theta_P \leftarrow [\theta(1 : T_{TL} - P), \dots, \theta(1 + P : T_{TL})]$ 
16:  $\rho_Q \leftarrow [\rho(1 : T_{TL} - Q), \dots, \rho(1 + Q : T_{TL})]$ 
17:  $X \leftarrow [\phi_N, \theta_P, \rho_Q]$ 
18:  $C = ((X * X^T)^{-1} * X^T) * Y(1 + T_L : T_T)$ 
19:  $Y(1 + T_L : T_T) = X * C$ 
20:  $Y(1 + T_L : T_T) = X * C$                                 ▷ Recompute using prediction error
21:  $e = Y(1 + T_L : T_T) - \phi(1 : T_T - T_L)$ 
22:  $X \leftarrow [\phi_N, e, \theta_P, \rho_Q]$ 
23:  $C = ((X * X^T)^{-1} * X^T) * Y(1 + T_L : T_T)$ 
24:  $Y(1 + T_L : T_T) = X * C$ 
25:
26:
27: for  $t = T_T + 1$  to ... do                                ▷ Forecast
28:   if ( $t \% T_R$ ) == 0 then
29:      $e = Y(t - T_T : t) - \phi(t - T_T - T_L : t - T_L)$       ▷ Recalibrate coefficients
30:      $\phi_N \leftarrow [\phi(t - T_T : t - N), \dots, \phi(t - T_T + N : t)]$ 
31:      $\theta_P \leftarrow [\theta(t - T_T : t - P), \dots, \theta(t - T_T + P : t)]$ 
32:      $\rho_Q \leftarrow [\rho(t - T_T : t - Q), \dots, \rho(t - T_T + Q : t)]$ 
33:      $X \leftarrow [\phi_N, e, \theta_P, \rho_Q]$ 
34:      $C = ((X * X^T)^{-1} * X^T) * Y(t - T_T : t)$ 
35:   end if
36:    $e = Y(t) - \phi(t - T_L)$                                 ▷ Compute Forecast
37:    $\phi_N \leftarrow [\phi(t - N), \dots, \phi(t)]$ 
38:    $\theta_P \leftarrow [\theta(t - P), \dots, \theta(t)]$ 
39:    $\rho_Q \leftarrow [\rho(t - Q), \dots, \rho(t)]$ 
40:    $X \leftarrow [\phi_N, e, \theta_P, \rho_Q]$ 
41:    $Y(t + T_L) = X * C$ 
42: end for

```

effectiveness of the coefficients, we can smooth the data using a low-pass filter.

We developed a model using this technique, with inputs of past flow (ϕ), air temperature (θ), and rainfall (ρ), defining their orders as N , P , and Q respectively, and a single output, predicted river flow (Y) (see Algorithm 1). This model, as we implemented it, self-calibrates, can use very little training data (on the order of weeks), performs a very simple set of operations, and requires 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.

Test Data and Setup

To analyze our algorithms, we use 7 years of rainfall, temperature, and river flow data for the Blue River in Oklahoma [26, 41]. This river and data come from an ongoing project called the Distributed Model Intercomparison Project (DMIP) run by the National Oceanic and Atmospheric Administration to compare hydrological models [30, 35]. The DMIP test provides more hydrometeorological data for the models than our model uses, allows for calibration based on 1 year of data, and requires a 1 hour prediction of river level for assessment [30, 35].

We define three different criteria for determining the quality of our algorithms: the modified correlation coefficient (taken from DMIP [35]), the false positive rate of prediction, and the false negative rate of prediction. For the modified correlation coefficient, as with the standard form used in probability, the value ranges from -1 to 1 with 1 meaning the two data series are identical. With this metric, 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. To define the false rates, we determine the flood level for the Blue River from the National Weather Service's online prediction work.

Model Calibration

We implemented the model as described above in Matlab, starting with defining the training window and recalibration. To help define the proper training window, T_T , we ran our experiments over several time windows: one month, one season (3 months), two seasons (6 months), three seasons (9 months), and a complete year. This covers all reasonable time periods for any generic river and any greater period of time becomes intractable for our system. For now, either we do not recalibrate the coefficients after the initial training or we recalibrate after we observe a new full training time window. Figuring out the optimal value for this we leave for future work.

Given these two parameter definitions, we analyzed the remaining parameters describing the models to determine optimal values. To pick the best values, we sweep the order (the number of past values used) for each of the three input variables with and without including the error of the past prediction. We find the optimal for each of the three metrics: the highest modified correlation coefficient, the lowest number of false positives, and the lowest number of false negatives.

In addition to the model with these parameters, we computed predictions using two 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). We also compare to the DMIP results, choosing the models that had the best modified correlation coefficient value for the Blue River (described in Section 5.1). The LMP model offers the best results; however this model is an instantiation of the SAC-SMA model described earlier (see Section 2), demonstrating the current operational centralized method. For this reason, we include the model built by the National Weather Service Office of Hydrologic Development, called the OHD model, in order to demonstrate the best distributed model in current research. Our testing computes predictions for time periods of 1 hour (for comparison with DMIP) and 24 hours (as a more realistic prediction window).

Section 4 discusses the requirements for implementing Algorithm 1 on a sensor network.

4. SENSOR NETWORK ARCHITECTURE

Algorithm 1 requires rainfall, air temperature, and water flow data collected and transmitted in real-time over the entire river basin area multiple times per hour. Since the flow of a river may change significantly over a period of several minutes, this suggest a sampling rate on the order of minutes. In order to support distributed, robust, real-time data collection, transmission, and, eventually, processing for large geographic regions corresponding to real river basins, we further define the following system requirements:

- Monitor events over large geographic regions of approximately 10000 km²
- Provide real-time communication of measurements covering a wide variety of variables contributing to the event occurrence
- Survive (on the order of years) long-term element exposure, the potentially devastating event of interest, and minimal maintenance
- Recover from node losses
- Minimize costs
- Predict the event of interest using a distributed model driven by data collected
- Distribute among nodes the significant computation needed for the prediction

The distance requirement, the inability to populate the entire area with sensors, and cost limitations, suggest a two-tiered approach with a small number of long-range nodes surrounded by a cluster of sensing nodes². The long-range nodes communicate over long distances on the order of 25 km using 144 MHz radios, have more power, and, therefore, can provide more online time for computation. The sensing nodes operate at 900 MHz, cover a more dense area, use

²With these three requirements, it is impractical to implement a homogeneous scattered sensor network.

low power, and have a smaller physical footprint. In addition to the field nodes, we integrate office and community nodes with the 144 MHz network. These nodes provide user interfaces to the system. Our goal is the development of a system capable of running a distributed version of Algorithm 1 in-situ. We have designed such a system; however, currently the flood prediction computation is centralized, but uses data from distributed sources. We want to avoid centralized prediction for two reasons: (1) to avoid a single point of failure and (2) to enable data-driven parsimonious calibration on the lower power 900 MHz network. Figure 4 shows a concept overview for this collection of four node types and how they communicate.

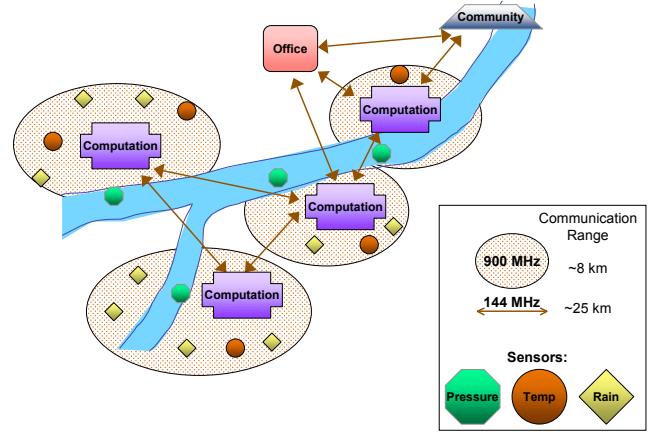


Figure 4: Idealized Sensor Network Consisting of Two Communication Tiers and Four Node Types; Communication Ranges Not to Scale

All four node types have a common base board and architecture that we then expand by daughter-boards as appropriate. The rest of this section describes our design and implementation for these system components. Figure 5 gives a more detailed overview.

4.1 Base System

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 [28]. The LPC2148 has a limited number of physical serial ports, which we extend to 8 by adding a Xilinx CoolRunner-II CPLD to the system and configure it as a serial router. The base board sends all free pins to the daughter-board connectors allowing for a variety of operations (e.g. digital input/output and analog conversion) and potential multiplexing of each I/O on specialized boards. A mini-SD Card and FRAM (Ferroelectric Random Access Memory) supply data and configuration storage.

In terms of power, most nodes operate in a simple pattern of wake-up, transmit, perform some operations (such as sensor measurements), and then sleep. We quantify some of these operations in Table 1, showing the power budget for a sensing node where transmission occurs every 10 minutes and measure every 5.

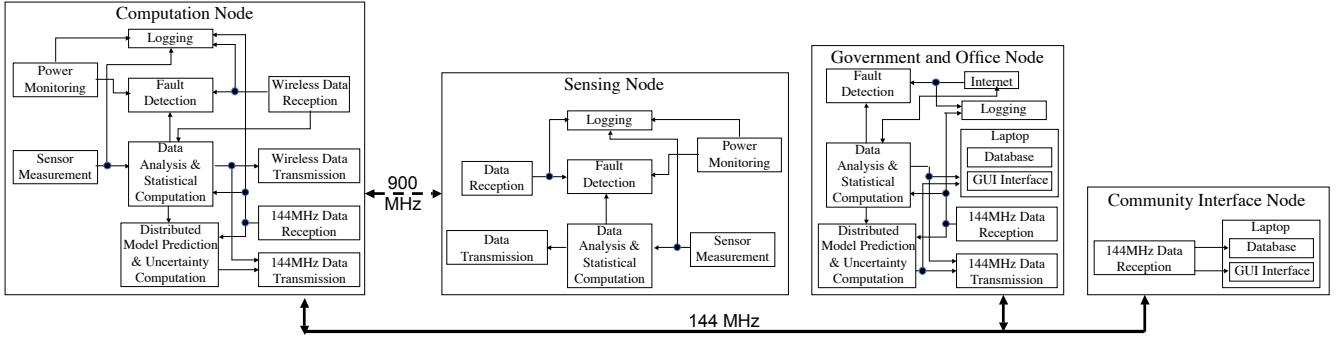


Figure 5: Generic Predictive Environmental Sensor Network Architecture Consisting of Sensing, Computation, Government Office, and Community Interface Nodes

Finally, a charging circuit on the board allows photovoltaic charging of lithium-polymer batteries, which power the system at 3.7 V, along with measurement of charge current.

Component	Time Active (min per 1 hr)	Current(mA)	mAh
LPC2148 Transmit	12	40	8
LPC2148 Measure	3	40	2
LPC2148 Power-Down	45	0.04	1.8
AC4790 Receive	3	30	1.5
AC4790 Transmit	1.5	1300	32.5

Table 1: Sensing Node Power Budget

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, power regulation, and a scheduler system based on the real-time clock and internal timers.

4.2 Communication

The two-tier communication structure uses 900 MHz and 144 MHz systems. As the 900 MHz system is used by almost all nodes, the base system provides that communication using a AC4790 900 MHz wireless module [1]. This wireless module provides RF and interface protocols, handling issues such as retries, error detection, and peer-to-peer communication. It operates at a fixed data rate, optimally 76.5 kbytes/sec but dropping to approximately 500 bytes/sec once the internal Aerocomm messaging overhead is considered. For the software, much of the transmission protocol is provided by Aerocomm, but we did implement several wrapper libraries for the AC4790 in order to interface with it and add our own simple packet structure.

A small subset of the nodes require 144 MHz communication. This frequency usually provides voice communication,

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 the radio, we added several libraries to the existing base software structure, most specifically wrapper functions to use the modem at a higher level than raw serial data. We developed our own very simple packet structure and communication protocol, defining addresses based on the pre-existing unique base board identifiers and setting a fixed number of retries.

Transmitting, the 144 MHz radio uses 25 W, although, in our system, this only occurs 6 times every hour for a period of 15 seconds. To power the radio, the system needs 12 V instead of the base board requirement of 3.7 V so we use a daughter-board to power the base board for these nodes. This allows us to completely power-down the radio when not transmitting, so the power budget does not change significantly from Table 1, adding 52 mAh for 144 MHz transmission. The 12 V power requirement of the radio means system cannot use the same lithium-polymer batteries as the sensing nodes so this system uses lead-acid batteries along with 6 W photovoltaic panels. Additionally, 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.

4.3 Sensing Nodes

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, and analyze data for indications of potential sensor failures. These nodes regularly transmit 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). The two operations of transmitting and measuring occur independently, enabling easy modification of the time windows for both the transmission time and the measuring time. By

regularly transmitting, nodes provide monitoring of each other through examining the data for errors and noticing the failure of any node not transmitting within an appropriate window. 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 can then transmit via the 900 MHz network to other nodes nearby.

In addition to the standard base board hardware features, this node requires hardware supporting multiple sensors and multiple sensor types. 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.

We now consider the specific sensors necessary for the application of river flooding: rainfall, air temperature, and water pressure. Other measurements could aid the prediction of river flooding; however, we chose only these three sensor types because of the ease of finding them, connecting to them, and installing them. Additionally, these sensors tend to be inexpensive. So far our modeling work supports using only these three; should this change, our focus on design generality allows for the easy addition of other sensors both from a hardware and a software standpoint.



Figure 6: Rainfall Sensor Node Consisting of Electronics, Sensor, and Photovoltaic Board

Rainfall 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. 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 6).

Measuring water pressure allows us to compute the water level. While our simulation work described in Section 3 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 we use the two values interchangeably. 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 7) 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.

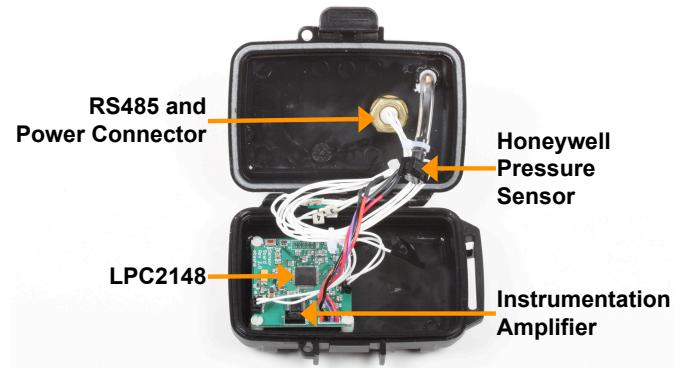


Figure 7: Pressure Sensor Box to Communicate with Sensor Node

4.4 Computation Nodes

Computation nodes connect the mini-networks of sensors, providing the communication backbone of the system (see Figure 8). These nodes will also perform the distributed computation of the prediction. Currently, as data arrives from nearby sensors and other computation nodes, the node maintains a record of all values, computes some data statistics, and examines the data correctness. Eventually, it will run this data through the model, compute the uncertainty of the model prediction, and request additional data from sensing nodes to reduce the uncertainty.

On the communication side, nodes communicate both via the 900 MHz network and to each other via the 144 MHz network. The 144 MHz modem and the power switching circuitry for radio control require a different daughter-board from the sensing nodes although we also include the various sensing ports and RS485 communication seen on the other daughter-board.

The communication range requirement drives the separation of these nodes from the sensing nodes with the additional focus of the computation due to the extra power available from the radios and in order to enable data-driven model calibration within the lower power 900 MHz network. Currently the node uses the same microprocessor as the sens-

ing nodes for prototyping purposes. As we further develop the prediction models and distributed algorithms, should we discover a need for more computational power, we could easily add an additional microprocessor to the daughter-board or even a GumStix.

4.5 Government Office Interface Nodes

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 for data and receive all of the existing network data. In addition to providing information to the office, receiving all the data will allow the office nodes to predict for the entire region using a centralized algorithm as a redundancy mechanism to the local distributed predictions. 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.

4.6 Community Interface Nodes

An effective use of this system requires an intuitive community interface. We have not developed this system component yet as user-interface issues are not our current focus. However, our design requirements for these nodes can be summarized as follows. These nodes provide an interface to the communities interested in the detection and prediction of the events. They utilize the same hardware and base software as the government nodes, but will provide a simpler user interface. The interface will display the known state of the geographic area, event predictions, event detections, and post-event monitoring. To avoid confusion, the interface will 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.

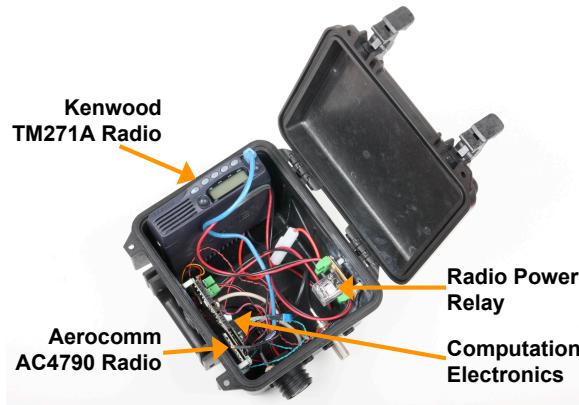


Figure 8: Computation Node

5. INSTALLATION AND RESULTS

We designed and performed three sets of experiments with the following goals: (1) test the flood prediction algorithm using a large set of physical river flow data (Section 5.1), (2) demonstrate long-term data collection of river flow data with a sensor network (Section 5.2), and (3) test the networking capabilities of our two-tier sensor network in a rural setting (Section 5.3). These are the three key components necessary to demonstrate integrated system performance for flood prediction. We are currently working on this integration and expect a comprehensive test in Honduras in January 2009.

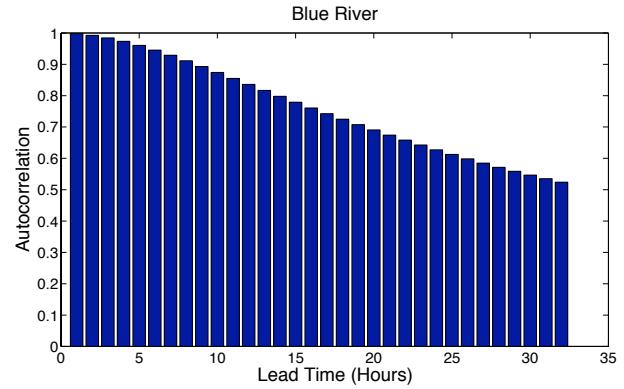


Figure 9: Autocorrelation of Blue River Data

5.1 Blue River

In this section, we use a large data set of real river data to test the validity of our prediction algorithm. For our river we examine the Blue River in Oklahoma, which encompasses 1233 km² at 153 m above sea level. No watersheds exist above this basin. The area has an average summer temperature of 82 °F and an average winter temperature of 36.7 °F with the occasional blizzard and tornados.

The DMIP data set for this river consists of 7 years of data from May 1993 to July 2000, measured from 1 river flow sensor, 6 rainfall sensors, and a weather station. We use only temperature from the weather station as we discovered that the other measurements are highly correlated with temperature. For the rainfall measurements, we average them before entering them into the model, as we expect to do when collecting the data on the sensor network.

Training Window (Weeks)	Order (Memory) of Flow Data	Order (Memory) of Temperature Data	Order (Memory) of Rainfall Data
4	5	1	14
12	5	1	14
24	5	1	14
36	2	2	16
52	2	2	16

Table 2: Order Calibration Results for Blue River

	Training Window (Weeks)	Modified Correlation Coefficient	False Positives	False Negatives
Our Model	4, 12, 24, 36, 52	0.998	0	0
DMIP Ave Uncalibrated		0.58	-	-
DMIP OHD Uncalibrated		0.71	-	-
DMIP LMP Uncalibrated		0.77	-	-
DMIP Ave Calibrated		0.70	-	-
DMIP LMP Calibrated		0.86	-	-
DMIP OHD Calibrated		0.86	-	-
Climatology 1 Hour	52	0.32	0	13
Persistence 1 Hour	0	0.998	0	0

Table 3: 1 Hour Prediction Results for Blue River

To determine the viability of 24 hour predictions, we examine the autocorrelation of the data set. As Figure 9 demonstrates, while the values decrease, at 24 hours the river correlates to itself at a value of 0.627, which implies a reasonable amount of data exists for using past information to predict that range. This also provides room for improvement in the predictability of the river; if the river had autocorrelation values of 0.9 or so at 24 hours, we could simply use persistence to predict, but here clearly have room for bettering the prediction.

We then calibrated the model as described in Section 3 to determine the optimal number of past measurements (or order) for each variable for the different training windows. For the 1 hour prediction, all time windows performed equally well using the last flow value, last temperature value, and last rainfall value. In fact, examination of the resulting coefficients demonstrates that only the latest flow value is used; basically, at one hour, persistence provides the best approach. Table 2 lists the orders that resulted in the best modified correlation coefficient for the 24 hour prediction case. We determined that better results always occurred when including as an input the error of the prediction associated with the latest observation and that our simple recalibration scheme did not improve results. Finally, maximizing the modified correlation coefficient provided the best overall results compared to minimizing the false rates.

Tables 3 and 4 show the overall results for this river, comparing our model and two naive approaches as well as the best cases for calibrated and uncalibrated DMIP models. In Table 3, examining the modified correlation coefficient, persistence and our model perform the best for 1 hour predictions. At 24 hours, our model performs better than persistence at all training time windows and both clearly perform better than climatology. Figure 1 demonstrates these results, showing our model with 52 weeks of training data and no recalibration. Additionally, although the DMIP results only apply to 1 hour predictions, comparing DMIP to the 24 hour predictions shows our model and persistence outperforming the average uncalibrated DMIP model. The average calibrated and best models from DMIP outperform our model with the average calibrated better than our model by 3%. This may seem like an unusual comparison; however, the lack of 24 hour prediction DMIP models limits us to this comparison, which does show that our 24 hour prediction is competitive with the 1 hour DMIP predictions.

5.2 Dover Field Test

We tested the long-term behavior of the system, specifically the sensing and 900 MHz communication, at Dover, Massachusetts on the upper Charles River from October through November 2007. 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. The data gathered at Dover allows us to connect the system and modeling work by running the information gathered there with our sensor network system through our Matlab model.

We installed 3 distinct sensor nodes (1 rainfall, 1 temperature, and 1 pressure sensor) within 900 MHz radio communication range at the locations shown in Figure 10. At the time of testing, we could not achieve the claimed 32 km range of the Aerocomm radios, seeing a maximum fully functional range of 1.6 km. While not ideal it is still sufficient for testing purposes; we continue to explore this issue for future field experiments. The pressure sensor we placed within a USGS sensing station, using their concrete shed as a base for our system. The other sensors we located across the river, with the rainfall sensor across from the pressure sensor and the temperature sensor upstream of both. While we could have collocated the rainfall and temperature sensors, we chose to keep them separate in order to maximize our testing of the 900 MHz network and better understand any problems related to the specific sensors.

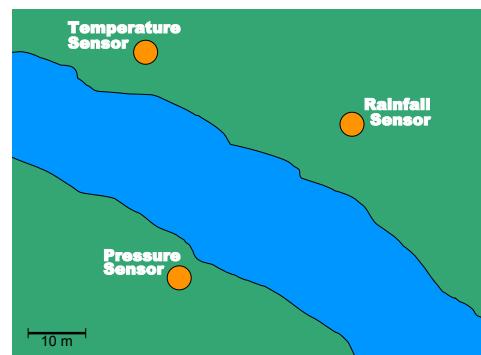


Figure 10: Locations of Sensors at Dover Site; Map Based on GPS Measurements and Surveying

	Training Window (Weeks)	Modified Correlation Coefficient	False Positives	False Negatives
Our Model	52	0.64	25	9
Our Model	36	0.61	25	9
Our Model	24	0.59	18	14
Our Model	12	0.59	18	14
Our Model	4	0.59	18	14
Climatology	52	0.32	0	13
Persistence	0	0.58	17	7

Table 4: 24 Hour Prediction Results for Blue River

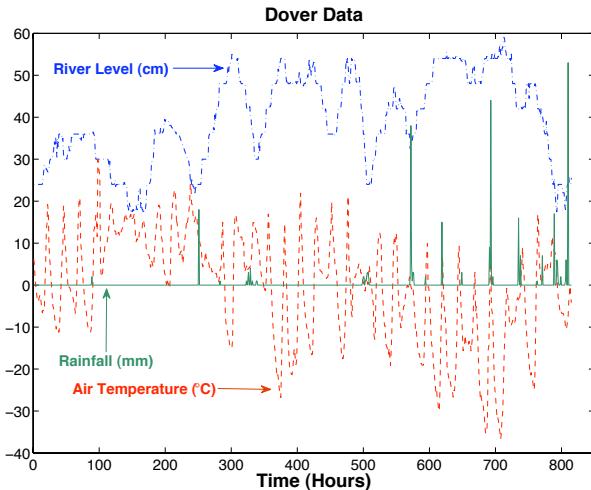


Figure 11: Data Collected from Dover Test Site

With the system we gathered 5 weeks worth of data before ending the field experiment due to winter weather. Figure 11 shows the hourly rainfall, 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 two of the models discussed in Section 3 for three prediction times using 4 weeks of training data. With 4 weeks of training, we computed the ideal orders of each of the variables as shown in Table 5. Ideally we would have more data in order to use more reasonable test windows and a larger range of training windows as we did with the Blue River, but, with the onset of winter, the upper Charles River freezes, limiting us to this amount of data until spring. Table 6 shows the results of these modeling runs, demonstrating again that with regression models we can improve over inherent autocorrelation of the river state.

We also verified our power information. Over a 5 day period, we saw a total discharge of 3153 mAh, which results in an average current of 26 mA. Charging over the same period resulted in a total of 1248 mAh. Two factors contributed to this less than ideal result: the extensive tree cover in the area limited effective panel placement, and many days saw heavy cloud cover and rain. We plan to increase the battery

Prediction Time	Order of Site Pressure Data	Order of Temperature Data	Order of Rainfall Data
1	1	0	3
16	3	0	5
24	2	0	5

Table 5: Order Calibration Results for Dover Site

capacity and develop some tree installation strategies for future deployments.

Overall, the field experiment was a useful indicator of the potential to predict flooding using statistical methods as we suggest with a sensor network system, demonstrating reasonable results up to 24 hour prediction windows with four 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

Our experiments in Honduras tested the two-tier architecture, deployment and maintenance issues, and issues specific to implementing these systems in developing countries. The work there began in January 2004 with the tower installation occurring in August 2005, the first communication test in March 2006, the test with sensing nodes and the office installations in March 2007, and the water prototypes throughout. All infrastructure remains with only the electronics removed to MIT for further work. We collaborate with a local non-governmental organization, the Fundación San Alonso Rodríguez (FSAR), to install the systems and understand deployment issues.

On the communication side, we verified our two-tier approach. We first focused on the usability of the 144 MHz radios. 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 FSAR 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 (see Figure 12) along with 10 meter towers at the FSAR office and the government emergency management office in Tocoa. With these towers,

	Prediction Time (Hours)	Modified Correlation Coefficient
Our Model	1	0.9925
Our Model	8	0.770
Our Model	16	0.596
Persistence	1	0.9923
Persistence	8	0.733
Persistence	16	0.554

Table 6: Comparison of Model Results for Dover Data

we verified both the communication range and the ability of our modems to communicate data over this range. Sending from our furthest tower 53 km away, with the radios set to the lowest 25 W transmission setting, we received all data packets transmitted. With no towers further away, we were unable to determine the maximum range possible, at this time. Due to hurricanes in 2005, we also proved that the towers and antennas will survive hurricane force winds.

Next, we added 4 sensing nodes to the system for a 4 day test. While no interesting weather occurred, this did verify collecting data from the sensing nodes, transmitting that data over 900 MHz to the computation nodes, and retransmitting that data over 144 MHz.



Figure 12: 5 Meter Antenna Tower for Computation Nodes

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. FSAR 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.



Figure 13: Installation of Water Level Prototype by FSAR Employees

Another area of testing has been the water measuring system. We have created five different prototypes of this system installing each for several months in Honduras with the help of FSAR (see Figure 13). 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 real-time need for the data, 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 sensor types. We focused on the event of river flooding, specifically in Honduras. The paper describes our work on the flood prediction algorithm that will eventually run on 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, by the Microsoft Digital Inclusion program, and by Xerox Research. We are very grateful for this support.

We would also like to thank FSAR for their support, the MA USGS office for use of the Dover site, and the following people for their help in the construction of the sensor nodes: Alex Bahr, Carrick Detweiler, Andrea Llenos, and Iuliu Vasilescu.

We are grateful to Hari Balakrishnan, Sam Madden, Matt Welsh and the anonymous reviewers for providing very thoughtful and detailed feedback on earlier drafts of this paper. Their comments helped us improve the paper greatly.

7. REFERENCES

- [1] Aerocomm. *AC4790 900 MHz OEM Transceivers User Manual*, 1.3 edition.
- [2] 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.
- [3] ALERT Systems Organization. Alert history. <http://www.alertsystems.org>.
- [4] E. Basha. Interview with COPECO officials in La Masica, Honduras, January 2004.
- [5] G. E. P. Box and G. M. Jenkins. *Time series analysis: forecasting and control*. Holden-Day, San Francisco, CA, USA, 1976.
- [6] R. Bras and I. Rodriguez-Iturbe. *Random Functions and Hydrology*. Dover Publications, Inc, Mineola, NY, USA, 1993.
- [7] 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.
- [8] Z. Butler, P. Corke, R. Peterson, and D. Rus. From robots to animals: Virtual fences for controlling cattle. *Int. J. Rob. Res.*, 25(5-6):485–508, 2006.
- [9] M. Castillo-Effen, D. H. Quintela, R. Jordan, W. Westhoff, and W. Moreno. Wireless sensor networks for flash-flood alerting. In *Proceedings of the 5th IEEE International Caracas Conference on Devices, Circuits and Systems*, pages 142–146. IEEE, Nov 2004.
- [10] Center for Hydrometeorology & Remote Sensing, University of California, Irvine. Hydrologic predictions - on-going activities. http://chrs.web.uci.edu/research/hydrologic_predictions/activities07.html.
- [11] 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.
- [12] 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.
- [13] Federal Communications Commission: Public Safety and Homeland Security Bureau. Emergency alert system. <http://www.fcc.gov/pshs/eas/>.
- [14] 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.
- [15] K. P. Georgakakos. Analytical results for operational flash flood guidance. *Journal of Hydrology*, 317:81–103, February 2006.
- [16] R. Guy, B. Greenstein, J. Hicks, R. Kapur, N. Ramanathan, T. Schoellhammer, T. Stathopoulos, 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.
- [17] T. M. Hopson and P. J. Webster. Operational short-term flood forecasting for Bangladesh: application of ECMWF ensemble precipitation forecasts. *Geophysical Research Abstracts*, 8, 2006.
- [18] 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.
- [19] 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 4th International Symposium on Information Processing in Sensor Networks*, pages 503–508. IEEE, April 2005.
- [20] 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.
- [21] 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.
- [22] G. H. Jørgensen and J. Høst-Madsen. Development of a flood forecasting system in Bangladesh. In *Operational Water Management Conference*, 1997.
- [23] 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.
- [24] 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.

- [25] 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.
- [26] N. Oceanic and A. A. N. W. Service. Distributed model intercomparison project. <http://www.nws.noaa.gov/oh/hrl/dmip/>.
- [27] N. Oceanic and A. A. N. W. Service. Hl distributed modeling research. http://www.nws.noaa.gov/oh/hrl/distmodel/abstracts.htm#abstract_7.
- [28] Phillips. *LPC241x User Manual*, 2 edition, July 2006.
- [29] 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.
- [30] S. Reed, V. Koren, M. Smith, Z. Zhang, F. Moreira, D.-J. Seo, , and D. Participants. Overall distributed model intercomparison project results. *Journal of Hydrology*, 298:27–60, October 2004.
- [31] M. Schwager, C. Detweiler, I. Vasilescu, D. M. Anderson, and D. Rus. Data-driven identification of group dynamics for motion prediction and control. *Journal of Field Robotics*, 25(6-7):305–324, 2008.
- [32] 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.
- [33] R. R. Shrestha and F. Nestmann. River water level prediction using physically based and data driven models. In *Zenger, 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.
- [34] 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 5th International Conference on Information Processing in Sensor Networks*, pages 492–499, New York, NY, USA, 2006. ACM Press.
- [35] M. B. Smith, D.-J. Seo, V. I. Koren, S. M. Reed, Z. Zhang, Q. Duan, F. Moreira, and S. Cong. The distributed model intercomparison project (DMIP): motivation and experiment design. *Journal of Hydrology*, 298:4–26, October 2004.
- [36] 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.
- [37] D. P. Solomatine and M. B. Siek. Modular learning models in forecasting natural phenomena. *Neural Networks*, 19(2):215–224, 2006.
- [38] 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.
- [39] 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.
- [40] 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.
- [41] TRIBS Development Team. tRIBS HydroMet data. <http://www.ees.nmt.edu/vivoni/tribs/weather.html>.
- [42] J. A. Vrugt, B. O. Nualláin, B. A. Robinson, W. Bouten, S. C. Dekker, and P. M. Sloot. Application of parallel computing to stochastic parameter estimation in environmental models. *Computers and Geosciences*, 32:1139–1155, October 2006.
- [43] 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.
- [44] 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.
- [45] 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.