A DATA-DRIVEN APPROACH TO SOIL MOISTURE COLLECTION AND PREDICTION USING A WIRELESS SENSOR NETWORK AND MACHINE LEARNING TECHNIQUES

BY

ZHIHAO HONG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Advisers:

Professor Ravishankar K. Iyer
Professor Zbigniew T. Kalbarczyk

Abstract

Agriculture has been one of the most underinvestigated areas in technology, and the development of Precision Agriculture is still in its early stages. This thesis proposes a data-driven methodology that aims to address some of the current problems in Precision Agriculture development. Soil moisture, a key factor in the crop growth cycle, is selected as an example to demonstrate the effectiveness of our data-driven approach. The success of the data-driven approach depends on two factors: (1) the quality of the data gathered and (2) the effectiveness of its analysis and interpretation. Previous studies have focused on addressing these factors separately, by either developing hardware for collecting soil moisture data or building efficient data analysis models.

In our work, we take a holistic approach by addressing problems on both ends and designing an integrated system for Precision Agriculture that uses a wireless sensor network and machine learning techniques. On the collection side, a reactive wireless sensor node is developed that aims to capture the dynamics of soil moisture while sampling at relatively low frequency to save energy. The sensor node dynamically adjusts its sampling frequency based on soil moisture readings and can be easily configured to meet the specific needs applications. The hardware is prototyped using MicaZ mote and VH400 soil moisture sensor. On the data analysis side, a site-specific soil moisture prediction framework is proposed based on models generated by the statistically sound machine learning techniques SVM (support vector machine) and RVM (relevance vector machine). The framework can integrate inputs from other reliable data sources to improve its accuracy. The proposed framework is evaluated under a historical dataset on 9 sites across Illinois. It achieves low error rates (15%) and high correlations (95%) between predicted values and actual values when forecasting soil moisture about 2 weeks ahead.

To my family and friends, for their love and support

Acknowledgments

I would like to express my deepest appreciation to my advisers, Professor Ravishankar K. Iyer and Professor Zbigniew T. Kalbarczyk, for their support and guidance. It has been a great experience and an honor to work in the DEPEND group, where I have learned much about research during the past two years. I believe this experience will benefit me greatly in my future career. I would also like to thank Professor Prasanta K. Kalita of the Agricultural Engineering Department for his great insights and feedback on my research.

I would also like to thank the University of Illinois for giving me the opportunity to finish both my bachelor's and master's degrees here. During my six years in Urbana, I have made friends and pals with whom I can share sadness and happiness. I am grateful for all the things that I have experienced here. They all helped me to learn and grow. Particularly, I thank the following members of the DEPEND group for the insights and support that I have received: Heidi Leerkamp, Daniel Chen, Catello Di Martino, Arjun Prasanna Athreya, Cuong Manh Pham, Homa Alemzadeh, Zachary Estrada, Phuong Minh Cao, Zachary Stephens, Qingkun Li, Subho Sankar Banerjee, Hui Lin, Yogatheesan Varatharajah, Key Whan Chung, Eric Badger, and Saurabh Jha.

Last but not least, I would like to express my deep gratitude toward my family members. I thank them for their understanding, patience, and love. Although we are apart most of the time and communication can be difficult, they always provide support and encouragement when I need it.

Table of Contents

Chapter 1

INTRODUCTION

The average farm size in the U.S. is increasing every year despite a continuously decreasing farmer population [1]. As a result, more and more cropland is shifting to large farms. A report from the USDA [2] predicts that over 20-25 years the average farm size will double. A large farm relies on a more structured and automated management system to realize better financial returns and use of resources. Globally, the demand for food has skyrocketed, especially in developing countries such as India and China. The prices of wheat and corn have tripled, and the price of rice climbed fivefold in three years from 2005 to 2008 [3], pushing 75 million people into poverty in nearly two dozen counties. Established 20th century solutions to meet food demand—clearing more land and using more fertilizer, pesticide, insecticide, and water—may no longer work [4].

Precision Agriculture (PA) promises to deliver the next generation of agriculture by actively using technology to collect various types of data and applying **site-specific, sensor-based treatment** to the farm. Figure 1.1 illustrates the PA vision. Data-driven agriculture is still at an early stage of development and faces many challenges. As pointed out in [5, 6, 7], the major problems for PA to become reality include:

- Crop management decisions and data collection systems need to be designed to meet the needs of specific farms.
- Automated and user-friendly systems need to be developed for users with less software experience.
- The introduction of expert knowledge must be possible. Systems should allow the inclusion of new automated methods for user-defined terms.
- Devices need to be affordable and scalable for large farm deployment.

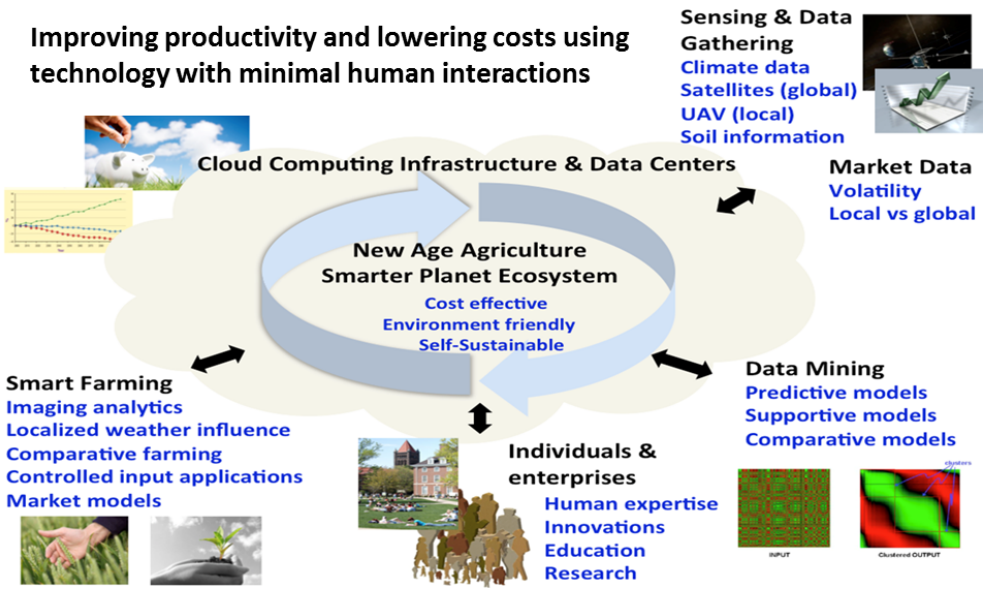


Figure 1.1: The proposed vision of Precision Agriculture

In our vision for PA or next-generation agriculture, farms would be built on a **data-driven or data-centric approach**. The objective of PA is to use data to improve productivity and yield, lowering resource and labor costs. As the world moves into the era of the Internet of Things (IoT), data are collected in various forms from different types of devices. A unified platform is needed to ensure that the data formats are consistent and that data are readily analyzable.

A combination of different types of data-gathering technologies should be applied together to allow site-specific data to be collected on a large scale without granularity loss. Data accuracy and integrity are crucial to a data-driven approach, as they impact the effectiveness of corresponding analysis tools. Collecting similar types of data using different methods can enhance the accuracy and integrity of the data. Also, mixing different data collection methods allows methods to complement each other's shortcomings. For example, remote sensing techniques are good at collecting large-scale data with coarse granularity, while wireless sensor networks promise to deliver data with fine granularity. *As a result, the inclusion of data from other sources should be considered from the system design and modeling perspective.* Once data are collected, data mining techniques can be applied to extract patterns and build estimation and prediction models that are valuable to farm management. The data can also be applied back to collection systems to make

them more efficient and reliable. In this way, next-generation agriculture becomes a feedback system where data not only optimize decision-making but also reshape data collection to meet specific needs. Figure 1.1 further highlights our approach and shows different components that contribute to our data-centric agriculture approach.

In this thesis, a site-specific data collection and data mining system is designed and implemented to fulfill part of the proposed vision using a data-driven approach to maintaining soil moisture. Some of the current problems in Precision Agriculture are addressed in our work by using a wireless sensor network and machine learning techniques in collection and prediction, respectively. Soil moisture, a key factor in plant growth, is closely related to irrigation, which consumes about 70% of the world's accessible freshwater each year [8]. We demonstrate that using a data-driven approach, the system can incorporate user-defined inputs and efficiently collect fine-grained soil moisture data and related meteorological data. Two regression supervised machine learning algorithms—support vector machine (SVM) [9] and relevance vector machine (RVM) [10]—are used to show the effectiveness of data-driven tools in building a site-specific soil moisture model. In our work, an integrated system is presented that addresses both ends of the data-driven approach: data gathering and data analysis.

Existing studies and research have provided separate solutions on the collection and analysis ends. In [11], a wireless sensor network has been deployed in large fields to collect soil moisture and meteorological data. The data acquisition procedure starts every 10 min for monitoring soil moisture dynamics in the field. A reactive sensor node was developed in [12] that samples at high frequency during rainfall. On the analysis end, soil moisture modeling has been studied for decades. Soil moisture analysis includes topics on physically based modeling, data-driven modeling, geostatistical analysis, and more. While designing physically based models requires significant in-depth knowledge of soil water and a statistics background, machine learning techniques can efficiently generate site-specific models, once the training methodology and respected dataset are set. Past research has applied neural networks [13], vector machines [14, 15], polynomial regression [16], and more on historical soil moisture datasets in the hydrology domain. *However, none of them built a system from the* ***Precision Agricultural perspective*** *that took a holistic approach by addressing problems in both collection and analy-*

sis. The success of the data-driven approach relies on the quality of the data gathered and the effectiveness of its analysis and interpretation. By looking at problems on both ends, a more unified and optimized system can be designed from the types of data gathered and the data granularity required for modeling tools.

1.1 Contributions

The specific contributions of this thesis are summarized as follows:

- We propose a data-driven methodology to address some of the current problems in Precision Agriculture. Soil moisture is at the core of plant growth and has effect on irrigation scheduling, yield forecasting, fertilizers use estimating, etc. Large volumes of data related to soil moisture and climate have been collected decades which are preferable in the context of using data-driven tools, and are easy to retrieve. For the above reasons, soil moisture is selected as an example to demonstrate the effectiveness of the data-driven approach. Using a wireless sensor network and machine learning techniques, we provide solutions for data gathering and data interpretation.

- We prototype a reactive wireless sensor node that can efficiently capture soil moisture dynamics using insights from historical data. A framework is proposed to let users easily configure the device to be application-specific. The prototyped device is tested on field soil to demonstrate its functionality and the responsiveness of the sensors. Using historical data from the Illinois Climate Network (ICN) [17], we demonstrate the effectiveness of our reactive sampling algorithm in capturing soil moisture dynamics compared with static sampling methods.

- We present a unique soil moisture prediction framework and evaluate the framework based on Illinois statewide historical data. The proposed framework is built on models generated by the SVM and the RVM algorithms. It achieves low error rates (15%) and high correlations (95%) between predicted values and actual values when forecasting soil moisture about 2 weeks ahead. Our experimental results show that

prediction outputs can remain accurate over a long period of time (one year) when models are corrected by reliable data from other sources every 45 days. A unified, well-formatted statewide dataset about soil moisture profiles and meteorology in Illinois is produced and can be used for further research.

1.2 Thesis Organization

The rest of the thesis is organized as follows: Chapter 2 provides the background of Precision Agriculture, other methods of estimating soil moisture, and the existing work using wireless sensor networks and machine learning techniques in agriculture. Chapter 3 briefly introduces the overall system and dataset used for our data analysis. Chapter 4 presents the detailed design of our smart wireless sensor hardware and its software algorithm. The implementation and evaluation of the collection system are presented in Chapter 5. The prediction model is described in Chapter 6. Chapter 7 discusses the experimental results obtained. The thesis concludes and future work is described in Chapter 8.

Chapter 2

BACKGROUND AND RELATED WORK

In this chapter, we provide background information on Precision Agriculture, techniques used in this work, and existing work in the field. First, an overview of Precision Agriculture is presented, followed by a discussion of the importance of soil moisture in agriculture. Then, we discuss the existing approaches to soil moisture measurement, estimation, and prediction. Background information on our techniques and related work is presented at the end.

2.1 Precision Agriculture

Precision Agriculture, or Precision Farming, is a farming management concept that was first introduced around the 1980s in the United States. The idea of Precision Farming is to bring automated technology into the agriculture industry to improve the effectiveness of agricultural practices and to increase the crop yield. The United States Department of Agriculture (USDA) calls this kind of agriculture "as needed" farming, in which farming decisions are made based on **data-driven approaches** [18]. The closed cycle in figure 2.1 helps to illustrate the concept in detail by breaking the concept down into a set of activities.

Within the Precision Agriculture framework, site-specific data on soils, crops, nutrients, pests, or yield are collected for analysis. By processing those raw but site-specific data, farmers are able to gain fine-grained insights into their farms and make smart decision accordingly. For instance, the agriculture sector consumes about 70% of the planet's accessible freshwater [8]. With the help of fine-grained, on-site soil moisture information, a farmer can apply water to places where it is needed instead of applying the same amount of water across the farm. Fine-grained information reduces the financial cost

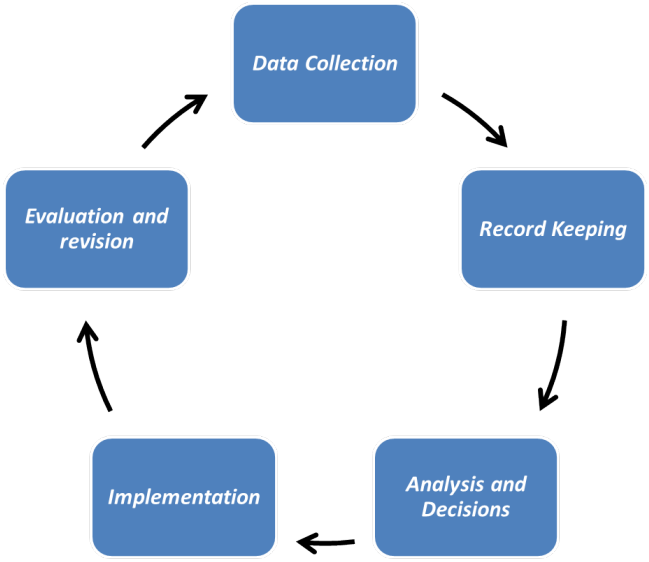


Figure 2.1: Precision Agriculture decision making cycle [18]

of irrigation and avoids the problems of over-irrigation.

As pointed out in [19], Precision Agriculture should be a holistic approach to reorient the total system of agriculture towards low input, high efficiency, and sustainability. This encourages us to take a systems approach to our topic of soil moisture and design a system that covers all the components of the decision-making cycle shown in figure 2.1.

2.2 Soil Moisture

The soil moisture value represents the fraction of the total volume of soil that is occupied by liquid (water), as expressed in the equation below. Quantitatively, *wfv* (water fraction by volume) or *vwc* (volumetric water content) are used as units to describe soil moisture level within a range of 0 to 1.

$$vwc = \frac{V_w}{V_t}$$

where V_w represents liquid phase (water) in the soil sample and V_t is the total volume of the sample.

Soil moisture plays a crucial role in crop growth and final yield because plant roots extract water from soil and react quickly to the environmental changes. In the work of [20, 21, 22], a strong correlation is found between

soil moisture and crop yields of wheat and corn.

The impact of soil moisture on crop growth varies significantly with depth. In [23, 24], water extraction at different depths is studied. Although corn roots typically extend to 90 cm, 96% of the water absorption from the crop seeding stage to the growing stage happens at depth of 5 cm to 30 cm, which is called the root zone. This information helps us to identify target depths for our soil moisture system.

Aside from the agriculture sector, soil moisture is frequently studied in hydrology and environmental science. It can be a critical factor in flooding and erosion. Therefore, soil moisture estimation and prediction methods can be useful in crop production, irrigation scheduling, flood prevention, and more. Agricultural soil moisture study should focus on the root zone, from depth of 5 cm to 30 cm, where crops extract the most water.

2.3 Methods of Soil Moisture Estimation

Conventionally, the most accurate method of obtaining soil moisture is the gravimetric method [25]. The gravimetric method requires a person to physically go to the field and take soil samples. The soil samples are kept in closed containers and proceed to the drying phase. In the drying stages, several steps are carefully taken to turn the soil into a solid via either a microwave oven or conventional oven. The soil moisture content can be calculated based on the mass loss of the sample during the overall process. Although accurate soil moisture value can be obtained, it is obvious that this method is time consuming and cost prohibitive. Furthermore, the gravimetric method can present the problems of choosing sample sites, and it is not suitable for large-scale farms.

Another method used to estimate soil moisture is modeling of soil profiles. In [26, 27, 28, 29], various models are proposed using physically-based empirical data. The model parameters are estimated together with environmental factors such as precipitation, temperature, etc. While these models show good results in their respected papers, designing a mathematical model requires significant in-depth knowledge of soil water and a statistics background. The physically based parameters require specific devices to measure. The models derived are very general, and a "one size fits all" model is pro-

posed for a particular soil type. We believe that obtaining a good "one size fits all" model can be very challenging due to the spatial and temporal variations in precipitation, soil property, temperature, vegetation characteristics, and many other environmental factors.

Remote sensing techniques have drawn lots of attention in this research field in recent decades due to advancements in satellite sensing and imaging classification technology. It has been proven that there is a strong correlation between soil moisture values and microwave emissivity and infrared data [30]. Satellites equipped with large diameter antennas and microwave sensors enable us to capture large-scale microwave images with relatively good spatial resolution. The collected data are then used to estimate large-scale soil moisture for the purpose of modeling the interaction between land and atmosphere with higher accuracy. To capture images with higher resolution requires that satellites stay at lower elevation and be equipped with larger diameter antennas, which in turn require more fuel to maintain in space [31]. With the advancement of unmanned aerial vehicles (UAV), remote sensing has become less expensive and more accessible to farmers.

Active and passive microwave remote sensing shows great capability to obtain observation of soil moisture at global and regional levels and has become the main focus of researchers in hydrology. One ongoing, state-of-the-art NASA research that launched in 2014 is the Soil Moisture Active Passive (SMAP) [32]. The SMAP utilizes a very large antenna and combined radiometers/radar to measure soil moisture at higher resolution than current radiometers can achieve. The passive radiometer will have a nominal spatial resolution of 36 km, and the active radar will have a resolution of 1 km [31]. SMAP uses high-resolution radar observation to disaggregate coarse resolution radiometer observation and produces a soil moisture value of 3 km resolution. These methods of integrating the use of active and passive sensors to downscale the passive microwave estimation of soil moisture has shown promising results in the work of [33, 34].

Remote sensing methods still have many shortcomings when applied in agriculture. First, the measurement is usually in macro-scale with each pixel in the 8-10 km range, which is too coarse. The large pixel size does not reflect the variations within the pixel and fails to provide fine-grained information by only forecasting the "average" value within the pixel. Secondly, the depth of soil that remote sensing is capable of forecasting is limited to the surface

area. As pointed out in [31, 35], the penetration depth is on the order of one tenth of the wavelength. The typical sensing depth of soil is around 2 cm to 5 cm, which does not cover the full depths of plant roots. The majority of water extraction for corn and wheat happens within the range of 5 cm to 30 cm, as mentioned in [23, 24]. Lastly, the raw collected data can be noisy. The electromagnetic response of the land surface is modified by surface roughness, vegetation canopy effects, and other factors that interact with atmosphere before the data is received by the sensor. With the increase in sensing frequency, more noise is added to the data. These factors influence the accuracy of the raw data and thus make the modeling result more inaccurate.

2.4 Wireless Sensor Network

In our work, we apply a wireless sensor network and machine learning techniques to soil moisture in order to overcome or compensate for the disadvantages of the methods discussed in the previous section. Background information about these two techniques and related work are presented in the following section.

2.4.1 Wireless Sensor Node

Wireless sensor networks (WSNs) have been widely used in data collection and monitoring. The data are collected automatically and promise to reflect fine-grained, dynamic changes. A wireless sensor node consists of four main parts: sensing unit, processing unit, transmission unit, and power unit, as shown in figure 2.2. A sensing unit comprises sensors and analog-to-digital converters (ADC). A wireless sensor node usually provides multiple ports where analog sensors can be attached. The analog signals produced by the sensor are fed to an ADC, which converts the reading into digital format. The processing unit is a lightweight microcontroller with a small amount of memory to process data. The transmission unit, often referred to as a RF chip, is responsible for sending and receiving data to the network. The RF chip can be in either sending state or receiving state, but not both. The power unit receives power from an external power supply and provides power

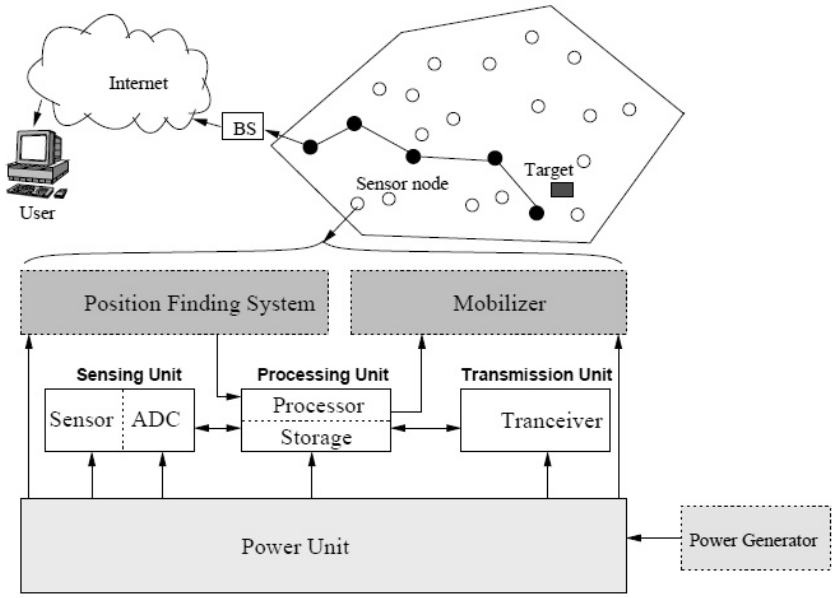


Figure 2.2: Wireless sensor node structure [36]

to other units.

2.4.2 Wireless Sensor Network

A wireless sensor network comprises spatially distributed autonomous sensors and a base station. Each sensor node is pre-loaded with some routing protocols that route data packets to the base station. The commonly used protocols are the star topology protocol, the tree-based protocol, and the cluster-based protocol. In star network protocol, all sensing nodes are considered peers, and communication happens only between the base station and peer nodes. This ensures minimal overhead to maintain the infrastructure. Tree-based protocol performs well if nodes are spatially distributed, while cluster-based protocol is more suitable in situations where the distribution of sensors is dense. The job of the base station is to collect data from the network and communicate with other networks or services, such as logging, Internet, or satellite.

2.5 Machine Learning

Data analysis tools can be applied to extract useful information, once field data are collected. In our work, machine learning techniques are used as tools to establish prediction models.

Machine learning is a scientific discipline that explores the design of algorithms that can learn from the data [37]. The main objective is to find an unknown relationship or to infer the function dependency between input and output data. Depending on the target or output type of the algorithm, the problem can be either classification (if the targets are nominal) or regression (if the targets are numerical). Based on the label status of the target value, machine learning techniques can be categorized as supervised, semi-supervised, and unsupervised. For building prediction models, supervised learning is used, since the training targets are well labeled.

2.5.1 Supervised Learning

In supervised learning, the objective of the learning algorithms is to infer a function from labeled training data. Pairs of training data consisting of input and output data are fed to the learning algorithm. The algorithm optimizes parameters of underlying functions by minimizing some objective function, such as error rate or boundary margin, and produces an optimized inferred function. Varieties of supervised learning have been developed, such as perceptrons, neural networks, and vector machines. This work uses two vector machine learning algorithms—support vector machine (SVM) and relevance vector machine (RVM)—to build models based on a preliminary study in [16], which explored various techniques for soil moisture prediction.

2.6 Related Work

In the next two sections, related work on WSNs and machine learning techniques is presented. In the last section, work that takes a systems approach and integrates both techniques is discussed.

2.6.1 WSN in Precision Agriculture

The development of WSN applications in Precision Agriculture makes it possible to increase efficiency, productivity, and profitability while minimizing unintended impact on wildlife and agricultural production systems. Real-time information from the fields provides a solid base for farmers to adjust strategies at any time.

The following are several successful projects using wireless sensor techniques.

- An automatic irrigation controller for Precision Agriculture was presented in [38]. The controller is an embedded sensor node that regulates the desired soil moisture level based on sensor readings. The sensor node takes soil moisture samples periodically, and the irrigation scheme is adjusted based on the readings.

- Wireless sensors have also been deployed to monitor the temperatures in vineyards [39]. The temperature information is used for predicting two important factors that impact the wine quality: head summation and potential frost damage.

- Akyildiz and Stuntebeck [40] developed an underground sensors system for monitoring soil conditions by deploying sensor nodes completely underground. The system can provide irrigation and fertilization information based on the measured water and mineral content.

- Sensors were also applied in the greenhouse environment, which is relatively stable and protects devices from harsh weather. Liu et al. [41] developed a wireless sensor network in a greenhouse that integrates a variety of sensors to measure substrate water, temperature, electrical conductivity, daily photosynthetic radiation, and leaf wetness in real time. The result shows an improvement in plant growth yield and in water and fertilizer schemes while reducing plant diseases related to over-watering. Wang et al. [42] developed a specialized wireless sensor node for monitoring temperature, relative humidity, and light inside greenhouses.

2.6.2 Machine Learning in Precision Agriculture

The following projects have shown promising results for applying machine learning techniques to agriculture.

- In terms of soil moisture, most of the machine learning techniques have been explored in the area of hydrology. Techniques such as neural networks [13], SVM [15], and multivariate relevance vector machines (MVRVM) [14] have been applied at sites such as rangeland and watershed. In [15], SVM techniques were used to predict soil moisture at the Little Washita River watershed in Oklahoma. In [14], MVRVM (a variation of RVM) was applied to predict deep root zone soil moisture based on surface parameters. In all of the work mentioned above, the prediction period is set to around 7-10 days ahead.

- There are some projects that focus on irrigation scheduling. In [43], the authors used neural networks to identify nonlinear relationships between plant water status and the textural features of pictorial information of the plant canopy.

- In [44], the authors use genetic machine learning approaches by running the WEKA [45] workbench to identify the status of strawberry plants. The system gathers environmental data related to lights and soil moisture to determinate the plants' health status. A number of rules are generated by the machine learning tools in WEKA to determine the threshold at which the plants' health status changes.

2.6.3 Work on Integrated Approach

While most of the research addresses sensing and analysis separately, there are a few projects that integrate the two approaches and build a complete system.

- In [46], a system is built for nitrogen fertilizer. A nitrogen sensor is designed and built to examine the water condition in the plant and soil online. Appropriate amounts of nitrogen fertilizer are then applied based on the reference number. The reference value of a crop-specific nitrogen requirement is established based on the data fusion of remote sensing data, real-time sensing data, and established knowledge.

- A Precision Agriculture application is presented in [47] that has a decision-making layer on top of the sensor deployment. In this work, the system learns by analyzing logging data from the sensors. Machine learning techniques are used to introduce new rules for water, pesticides, or fertilizers.

- In [48], a real-time feedback system is designed for personal health. The data is collected in real time from mobile sensors on people and sent to a server for analysis. The analysis engine is able to extract information from raw data and give real-time recommendations about certain actions to both the data source and people who share similar patterns in their data.

Chapter 3

SYSTEM OVERVIEW

The goal of this chapter is to give a brief overview of our soil moisture system and the data used in our data-driven approach. Detailed descriptions of the collection and prediction systems are presented in the following chapters.

3.1 System Overview

Figure 3.1 shows an overview of the system, which can be separated into two parts: collection and prediction. Using data-driven approaches, solutions in each part address the problems mentioned in the Introduction and are evaluated using historical data. The design principle is to create frameworks where the system can be configured in a site-specific way and be able to take inputs from other sources. Following the Precision Agriculture decision-making framework in section 2.1, our work provides solutions for each part in the decision-making cycle, from "data collection" to "analysis and decision" and then to "evaluation and revision." Using our sensor node, the data can be collected and received by the base station for real-time monitoring and data analysis purposes (data collection). We demonstrate that the collected data can be used to provide data pattern insights and train prediction models (analysis and decision). The analysis results on data patterns are then applied back to the collection system for configuring fine-tuned device parameters (evaluation and revision). We believe the same methodology can be applied to designing systems in other areas of agriculture as well. Due to the large data size required by the machine learning algorithms, the field collected data from our sensor node are not used in our prediction experiments. However, the two sub systems are connected in the sense that the same soil and environmental attributes are collected and used in collection and prediction, respectively.

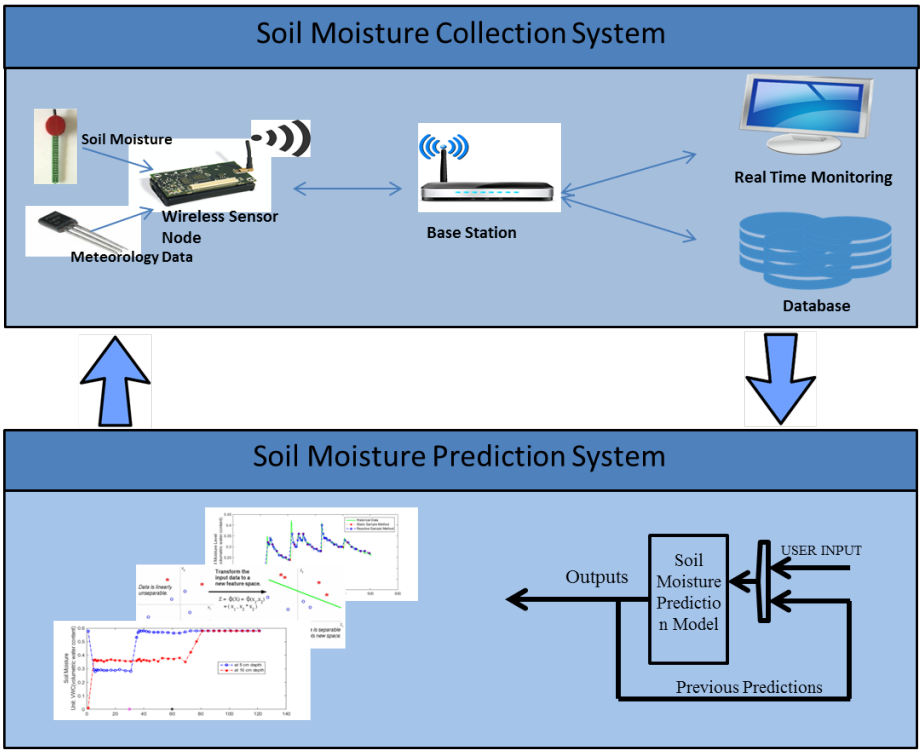


Figure 3.1: Proposed soil moisture collection and prediction system

In the collection system, a wireless sensor system is designed based on data analysis from the Illinois Climate Network (ICN). The wireless sensor node is prototyped using MicaZ mote [49] and collects soil moisture and other meteorological data. The sensor node is an intelligent reactive device that focuses on collecting soil moisture dynamics data with respect to surrounding environment changes. To address the problems of site-specific and user inputs in Precision Agriculture, the sensor node is programmed using open source platform TinyOS [50] and offers two user-defined variables regulating the level of data granularity and sample intervals. A reliability layer is added on top of the sensor node to increase overall robustness. Our wireless sensor network can be used for applications such as in-field soil moisture collection and other kinds of remote site data collection, since it is specifically designed for applications that require a long lifetime.

In the prediction system, machine learning techniques are applied on 9 different sites across the ICN, and a prediction framework is built on top of the machine learning models to predict soil moisture n days ahead. The models predict the soil moisture value based on meteorological parameters

including temperature, humidity, wind speed, solar radiation, precipitation, and soil temperature together with the previous days' soil moisture values. The sparse and well-studied machine learning techniques SVM and RVM are applied on the historical data to derive mathematical models. Designed from a Precision Agriculture perspective, the site-specific model is able to incorporate data from other sources at the granularity of one day. In contrast to soil moisture as studied in hydrology, where variations in soil attributes are caused mainly by environmental changes, cropland is regularly maintained by people. Hence, the variations can come from both meteorologic changes and human interventions. The feature of taking user-provided data at fine granularity makes the system more robust by allowing the model to interact with human knowledge or real soil moisture data from other sources. The proposed framework achieves low error rates (15%) and high correlations (95%) between predicted values and actual values when forecasting soil moisture about 2 weeks ahead. It should be noted that some factors affecting soil moisture, such as leaf area index and root water extraction, are not included in our current work, due to the lack of these data. Depending on the crop type and its growth stage, these types of data vary. Our machine learning method can incorporate these parameters by treating them as new features to improve model accuracy, once these data become available.

3.2 Data Source

The data used in this study are from the Illinois Climate Network (ICN) program, which is one of the main programs under the Illinois State Water Survey (ISWS). ISWS operates as a Division of the Prairie Research Institute of the University of Illinois at Urbana-Champaign. It is responsible for collecting, analyzing and archiving high-quality, objective data related to water sources in Illinois. ICN monitors weather and soil conditions at the 19 locations identified in figure 3.2. Historical data from 1989 to 2012 are available to the public upon request.

The meteorological dataset from ICN consists of data from the 19 automated weather stations scattered across Illinois and maintained by ISWS staff. The weather stations collect temporal weather observations on numerous weather variables such as temperature, precipitation, relative humidity,

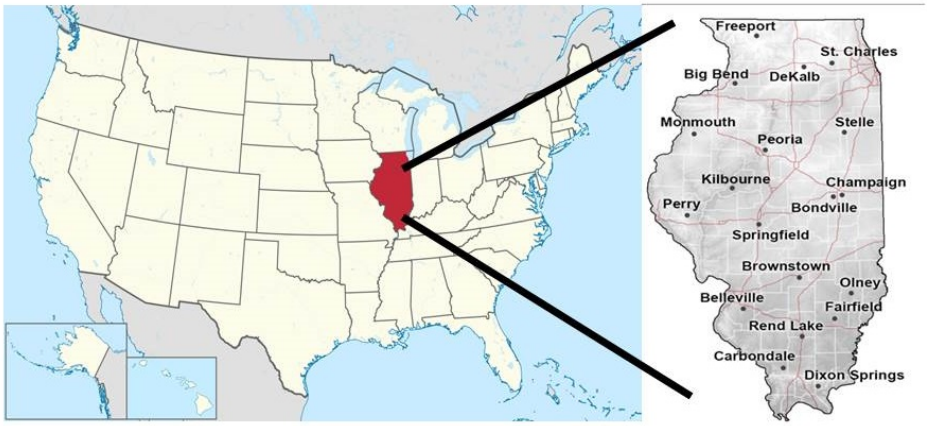


Figure 3.2: Statewide map of ICN sites

barometric pressure, wind speed, wind direction, and solar radiation. Most sensors are polled every 10 seconds and are averaged by the hour. All the weather data are downloaded to the ISWS database once a day.

The soil-related dataset is formed from 17 sites across Illinois. Most of the soil sites are co-located with one of the 19 weather stations mentioned above. At some sites, data are collected manually from site visits twice a month during the growing season (March to October) and once a month during the rest of the year. At the rest of the sites, the data are collected by sensors placed at different soil depths. The soil moisture data are measured at the site using Stevens Hydra Probe sensors that sample every hour at depths of 2, 4, 8, 20, 39 and 59 inches below the soil surface with accuracy of $\pm$ 0.03 vwc. The soil moisture information has a great impact on Illinois agriculture, offering potential insights into water resource management of the state. Also, across the U.S., there are approximately 220 remote sites collecting soil moisture and soil temperature along with precipitation, wind, and solar radiation data. The Soil Climate Analysis Network (SCAN) [51] offers those data to the public upon request, too.

Chapter 4

SOIL MOISTURE COLLECTION SYSTEM

This chapter introduces the design aspect of our wireless sensor network. The goal of this chapter is to provide a detailed description of our soil moisture data collection network with focus on design principles. It begins with an overview of the sensor node, its features and novelty. Then we discuss the design methodology used in the work, followed by descriptions of sensor node flow. We propose that data analysis can be applied to historical data to gain insights into soil moisture at monitoring locations and to create a framework in which the device can be configured to be more site-specific.

4.1 Collection System Overview

Our objective in the collection system is to design and prototype a wireless device that can efficiently collect the soil moisture data and related environmental data needed for training prediction models in Chapter 6. The prototyped sensor node is composed of environmental sensors and hardware devices from MicaZ platform. It is capable of collecting soil temperature, air temperature, humidity, and soil moisture at different depths. Data analysis is applied on the ICN data to help gain insights into data characteristics and patterns at monitoring locations.

The sensor node is a reactive device that focuses on collecting soil moisture dynamics with respect to changes in surrounding environmental parameters. Based on analysis of the ICN dataset, soil moisture does not change much during most non-rain days. A sampling rate adjustment algorithm is implemented and loaded on the hardware to adjust its sample rate according to the difference between previous and current readings. Reactive sensing allows the device to intelligently capture soil moisture dynamics with fine granularity while not spending unnecessary energy on sampling and communication

operations. The sensor node is modular and has two user-defined variables, making it a more application-specific device. By setting these two variables, users specify the level of data granularity and sample intervals for the sensor application. Lastly, we add a layer of reliability to the sensor to increase the robustness of the system. Our wireless sensor network can be useful for applications such as in-field soil moisture collection and other remote site data collection applications, since it is specifically designed for applications that require long life and remote site deployment. Compared to previous related work [52, 53, 54, 55], our sensor network has three distinctive features:

- **Reactive Sensing** extends the lifetime of the system by reactively sensing data based on environment changes.
- **Reconfigurable Devices** provide a framework in which user-defined variables can be easily used to configure different applications on an open source platform.
- **Robustness** is achieved by logging accurate data locally in the presence of network loss or power failure in harsh outdoor field conditions.

4.2 Design Methodology

Soil moisture data has been collected for decades in the U.S. Public websites such as the Soil Climate Analysis Network (SCAN) [51] and Illinois Climate Network (ICN) give data to the public upon request. With this huge amount of historical data, the problem becomes how to leverage that data and transfer the raw data into useful information.

We take a data-driven approach to gain insight into the data and use analysis results to shorten the development cycle. By analyzing historical data, we are able to gain knowledge about monitoring locations and monitoring subjects. This helps us design the system to be more application-specific and efficient in gathering high-quality data with minimal power consumption. Also, there is usually a long iterative process of hardware deployment in which several field tests need to be performed in order to fine tune the hardware parameters. The analysis results also shorten the process of evaluating the design and tuning hardware parameters.

The main roles of historical data can be summarized as follows:

- Present problems of existing solutions for the data collection process
- Provide insights into interested data characteristics
- Evaluate the effectiveness of sensor-sampling algorithms

The following work provides valuable insights into our hardware design and sensor deployment. In [52], a wireless sensor network has been deployed on buildings for structural health monitoring with high-frequency sampling. The performance of EEPROM flash memory on a Mica node is measured. In [53, 54], MicaZ motes have shown solid performance in collecting data in wild fields. It was found in [56] that, among all the MicaZ operations, transmitting a packet and writing to a flash memory take the most energy per operation. Bogena et al. [55] evaluated a low-cost soil water content sensor (EC-5) for wireless network sensor application. In their work, the sensor node took samples periodically, and the accuracy of the reading was studied given the changes of temperature.

4.3 Reactive Node Design

Our sensor node has three main advantages over previous designs. In this section, we present each advantage in detail.

4.3.1 Reactive Sensing

Power management is one of the most important aspects of wireless sensor networks. The sensor node is embedded hardware with only limited computational power and energy supply. The main contributors to power consumption are transmitting packets, writing to memory, and sampling data. In a MicaZ mote, packet transmission and flash write are the top two power consuming operations. Receiving a packet, transmitting a packet, and writing to flash memory cost about 8 nAh, 20 nAh, and 83 nAh, respectively [54]. Furthermore, in real deployment [55], it is shown that the battery maximum voltage supply starts degrading over time. The performances of sensor and radio transmission are affected when the power supply cannot meet their minimum requirements.

To extend the lifetime of a sensor node, it should be in *sleep* mode as long as possible. The power management in MicaZ allows the hardware to

alternate between *activity* and *sleep* mode. The power draw in *active* mode is 5 to 20 mA, while the *sleep* mode only draws 5 μA [55]. The sleep mode can significantly reduce power consumption by moving the microcontroller into lower power states and only keeping a few necessary components, such as the clock, in a functional state.

A sampling rate adjustment algorithm is implemented and loaded on the hardware to adjust sensing frequency based on soil moisture readings. Unlike other sensor applications in which the sensor samples at a high frequency (several samples per second), the soil moisture content in a wild field does not change much on an hourly basis, especially in non-rainy days. Hourly sampling is a waste of energy and adds no value to the collected information. A sample frequency adjustment algorithm is implemented on the sensor node to make the hardware collect data more intelligently and efficiently. Since our sampling decision algorithm runs on embedded systems, the algorithm should be straightforward and simple. In our design, the decision of whether to adjust the sample frequency is based on the difference between previous readings and current readings. If the difference exceeds some pre-set threshold, it is likely a rainy period at monitoring locations. In response, the sample intervals are exponentially decreased to a much shorter sample window to capture the variations. Once the soil moisture readings become stable, the sample intervals start to increase linearly until the maximum sample interval is reached. More detail on the algorithm is covered in section 4.4.

4.3.2 Reconfigurable Devices

A trade-off exists between obtaining high-resolution data and having an energy-efficient and sustainable sensor network. Obtaining high-resolution data requires hardware to take samples, perform computation, and transmit packets at short intervals and thus takes more energy. Depending on the particular situation at a real deployment site and the monitoring goal, the requirements for the monitoring period and the data granularity can be quite different. For example, if the sensor system is used for experiments on soil properties in a lab setting where environmental parameters such as temperature are manipulated by people, a higher resolution of soil properties

is needed for accurate analysis. However, if sensors are deployed on a remote island for a years-long monitoring project, the importance of system lifetime outweighs the importance data resolution. As a result, the system should be flexible for various applications and be site-specific according to the concept of Precision Agriculture.

In our design, users can create a more customized, site-specific sensor node to better meet individual application requirements and goals. To achieve that, users need to specify two variables: *maximal sample interval* and *level of granularity*. The *maximal sample interval* is the largest sample rate that a node can hold. By default, if the soil moisture values stay the same, sensors continuously monitor soil at the *maximal sample interval* rate. The *level of granularity* specifies the threshold value, which triggers the sampling rate adjustment algorithm in the reactive sensing part. It can be understood as the level of sensor sensitivity to environmental changes. The lower the *level of granularity*, the more often the sample rate adjustment algorithm can be triggered and the more samples taken. These two variables let the users use their knowledge of the monitoring target and goal to create an application that meets their needs. The TinyOS platform is used for software implementation. The open source feature of the platform makes the porting of code much easier. After specifying those two variables, a more customized TinyOS image can be compiled and installed onto the mote.

The introduction of the two user-specified variables allows the sensor node to incorporate inputs from data analysis or expert knowledge. For instance, historical data can be used to determine optimal values for monitoring soil moisture at the ICN DeKalb site. It is observed that the hourly soil moisture values at 5 cm, 10 cm, and 15 cm do not vary much most of the time at the station and only change rapidly if there is precipitation. Figure 4.1 shows the distributions of time periods that soil moisture values at depth 5 cm take to have a variation greater than 0.04 vwc. The analysis examines the data in chronological order by checking if the current soil moisture value varies from the baseline value by more than 0.04 vwc. If so, the time span between the two values is added to the end of *soil_time_period* array. The baseline value is initialized to the first data point in the dataset and is refreshed to the current value every time new time span data is added to the *soil_time_period* array. The analysis is based on 2 years data—2*365*24=17,520 data entries—collected from the ICN DeKalb station, which provided hourly soil moisture

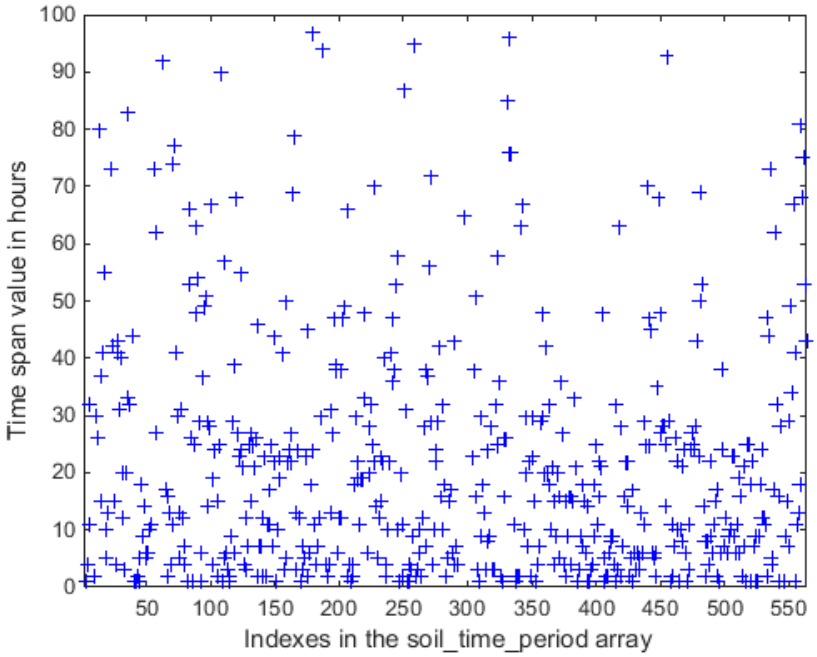


Figure 4.1: Soil moisture at 5 cm depth with *level granularity* 0.04

data. The mean of distribution is 44 hours, and most points are located at a 10-30 hour range, as shown. Compared to locations with frequent rain, the sample interval at DeKalb can be set to about 10-20 hours instead of 2 or 4. By extending the sample frequency from 2 and 4 hours to 12 and 16 hours, the lifetime of the sensor node can be significantly extended. Statistical methods such as linear interpolation and reactive sensing can further compensate for the loss of data granularity.

4.3.3 Robustness

In the process of data collection, the robustness of the wireless system can be affected by many factors. We categorize two main sources of impact on the overall robustness of the system: embedded devices and the wireless communication link.

Embedded systems (sensor node and sensor) are unreliable and error-prone due to their hardware computational constraints. When deployed in a remote area, the sensor node needs to withstand the harsh outdoor environment and maintain its power. It is common for sensor nodes to shut down due to

battery depletion. The data stored in volatile memory cannot survive the power cycle and are lost once the sensor node is shut down. Also, errors can be introduced into the data when samples are read. In practice, the sensor hardware, especially analog hardware, tends to give an inaccurate sample reading in its first use after long hours of inactivity. Inaccurate sensor reading hurts the accuracy of data modeling and of future analysis. The data error is hard to detect in the dataset and must be addressed from the source.

To prevent errors or inaccurate data readings from propagating to the analysis dataset, the sensor node is programmed to perform three consecutive sensor readings for each sample operation. The last reading of the sensor is considered the "accurate" reading and sent to base station. Based on our observations of sensor readings, we decided to program the sensor node to read three times, but the number of readings can be easily changed depending on the situation.

Wireless communication is unreliable and likely to lose data as well. In order to receive messages from the sensor node, the base station needs to be present and to remain in listening mode. Even with the presence of the base station, a sent message can also be lost during communication and fail to reach base station. The quality of the wireless link depends on environmental factors, such as air moisture level, but also on overall message traffic. In experiments on large-scale sensor networks, the loss rate usually worsens due to message conflicts at the base station.

We use a base station message acknowledgment mechanism from the RF chip, together with local EEPROM flash memory, to increase the system robustness. EEPROM is a configurable, non-volatile flash memory on MicaZ mote that is able to preserve the data with the power cycle of the sensor node. The MicaZ mote has about 4 Kbyte of EEPROM memory and can permanently store up to hundreds of data entries. When a message is received at the base station's transceiver end, the base station sends out an acknowledgment message to indicate the message has been successfully delivered. If the acknowledgment message is not received within a certain time window, the sensor node logs the message into EEPROM flash. There are two common approaches to deal with message delivery failures: either re-send the message immediately or store the data locally. We choose the second approach, since we assume the distribution of sensor nodes is sparse. Message delivery failures should be rare and are most likely due to absence of a base

station. Work in [12] uses EEPROM as a backup storage device, and sensor nodes log each sent message. Compared to that approach, our design is more energy-efficient. It has been shown that writing to flash is more expensive in terms of energy cost than transmitting a packet. The operation of flash writing should be performed only as needed.

Our sensor node is also able to perform properly as a logger alone, because of local EEPROM flash memory. Soil moisture loggers are often used in agriculture to log the events periodically, and real-time monitoring is not required.

4.4 Work Flow

Figure 4.2 illustrates the work flow of the sensor node. Most of the time, the sensor node stays in *sleep* mode. When a timer interrupt happens, the sensor node first takes three consecutive readings from the data acquisition board on the analog ports. The last reading is set as the final reading. Data are packed into one message with the event timestamps and node ID. The sensor node sends the message to the base station and waits for the acknowledgment

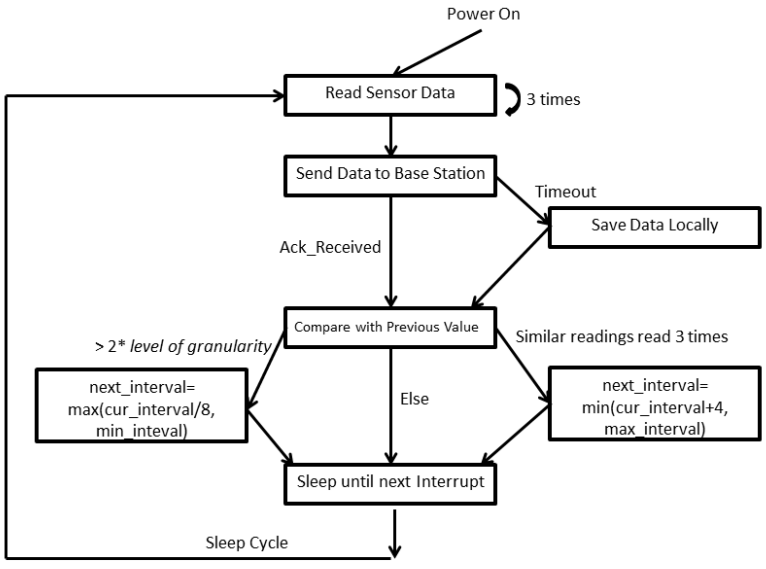


Figure 4.2: Sensor node work flow

signal. If the acknowledgment is not received within a certain time, the message is written to the flash memory for logging. If the acknowledgment is received, the sensor node moves on to the reactive sampling algorithm section.

4.4.1 Reactive Sampling Algorithm

The reactive sampling algorithm checks whether the sample interval should be adjusted. If in the latest 3 readings, all the values are within the *level of granularity* specified by the user, the sample interval is increased by 4 until it exceeds the maximal sample interval. If the difference between the previous and current readings is greater than 2 * *level of granularity*, the sample interval is shortened to one eighth of the current sample interval. Reactive sampling ensures that when a rapid change is detected, the system is able to sample at much shorter intervals. This increases the chances that interesting data is sampled. An evaluation of the algorithm is presented in the next chapter to test its performance on real collected data.

Reactive sampling does not guarantee that all the soil moisture fluctuation will be captured, but it increases the chances. The intuition behind reactive sampling is that when soil moisture varies, it is likely raining or during a rainy period that can last a few days. Reactive sampling does not work in the event of a sudden but short rain storm. Depending on the real deployment scenarios and requirements, the algorithm parameters can be further tuned. Another possible solution is to integrate the barometer sensor readings into the algorithm. Barometer sensor readings can serve as an indication of rain likelihood and trigger the sample intervals to change. A barometer sensor is available in our sensor node platform and can be studied in further work.

Chapter 5

SYSTEM IMPLEMENTATIONS AND EVALUATION

In this chapter, the sensor node's hardware components and software implementations are presented. First, the hardware platform and sensors used in this work are introduced, followed by a detailed description of the software application installed. At the end, demonstration results on the functionality of the sensor network and the effectiveness of our reactive algorithm are presented.

5.1 Hardware System

5.1.1 Hardware Platform

Aside from basic functionality, we mainly consider two aspects when selecting proper wireless hardware platform on which to build a sensor node: (1) peripheral hardware supports and (2) software supports. The following hardware components are chosen as our sensor hardware platform:

- Crossbow MicaZ is used as our sensor node platform and programmed in NestC. MicaZ Mote is designed specifically for deeply embedded sensor networks and operates in the 2.4 GHz ISM band. The node is compliant with IEEE 802.15.2 with 250kbps data rate [49]. It is based on Atmel ATmega128L low-power microcontroller and can be configured to run sensor application processing and a network communication stack simultaneously. The communication range is estimated at around 75m to 100m outdoor and 20m to 30m indoor [49].

- MDA300CA Data Acquisition Board [57] is used to collect analog readings from the sensor. MDA300CA is a multi-function data acquisition board with temperature and humidity sensors. It can be used as an interface

board between sensor and mote in various sensing applications. The hardware provides 2.5 V, 3.3 V, and 5 V excitation voltages to external sensors. There are 7 single-end ADC channels on the MDA300CA, which provides a sensor reading resolution of 12 bits. Signals with dynamic range of 0 to 2.5 V can be plugged into these channels. The formula to convert readings to voltage is as follows:

$$Voltage = 2.5 * \frac{ADC\ READING}{4096}$$

- MIB520CB [58] board is used as the base station and USB programming board. MIB520CB provides a USB interface to a PC, allowing data on the mote to be sent to a PC via serial port. To program the mote, it needs to be mounted on the MIB520CB via a 51-pin connector before a TinyOS image can be installed on it. MIB520CB can also act as an interface between a PC and a base station node. When doing data collection, MIB520CB can send the data from the mote to the PC in real time.

MSP430 Launchpad [59] from Texas Instruments (TI) is considered as a possible sensor node. MSP430 Launchpad is a popular microcontroller evaluation kit from TI for use in prototyping design. One additional radio chip needs to be mounted on top of the evaluation kit to enable wireless communication. However, MSP430 products from TI are not officially supported by the TinyOS system, and porting requires much effort. Unlike writing in TinyOS, which provides relatively high-level abstraction of the sensor component and radio, developers of Launchpad need to obtain detailed low-level hardware knowledge specific to Launchpad in order to program the board. Also, the supply power voltage Vcc from Launchpad is 3.6 V, which is below the minimum voltage required by our soil moisture sensor.

Another candidate sensor node is Waspmote [60] developed by Libelium. Waspmotes are commercial-use sensor nodes designed for real world applications. Libelium offers more than 70 sensors that can be integrated with Waspmote for various sensing applications. However, we chose MicaZ over Waspmotes because the latter is proprietary hardware and more expensive than the MicaZ hardware platform.

One advantage of using MicaZ is that it is officially supported by the TinyOS system and a power management system is provided. Power con-

sumption is an important factor when building a sensor node. Unlike other sensor nodes that sample at a high frequency, our application only requires taking samples hourly. Thus, power management is essential to ensure that the wireless radio chip, microcontroller, and sensor devices are in the *sleep* mode most of the time.

The TinyOS platform offers modules such as basic scheduler and MCUSleep to help manage the power states of a microcontroller and other devices automatically. Microcontrollers often have several power states, each with particular rules to determine the on/off state of several components. For example, the MSP430 microcontroller turns off ADC components at LPM3 mode. Microcontrollers should always be in the lowest possible power state that can fulfill the application requirements.

The basic principles of those power management modules are to estimate incoming tasks based on the task queue and other conditions. Every time a microcontroller handles an interrupt, the scheduler moves the state to *active*. However, if the task queue is empty, radio is off, and SPI interrupt is disabled, the scheduler moves the mote into *sleep* mode by moving the microcontroller into one of the low-power states. With the help of power management modules in TinyOS, the lifetime of the WSNs can be greatly increased.

5.1.2 Sensors

The following sensors are used in our soil moisture sensor node to collect various types of data:

- Temperature and Humidity Sensor

  The temperature and humidity sensor Sensirion SHT11 is included on the MDA300CA sensor board. It measures relative humidity (RH) with a resolution of 0.03 RH and accuracy of +-3.5% RH. The temperature sensor has a measurement range of -40°C to 123.8°C with a resolution of 0.01°C and accuracy of +- 0.5°C at 25°C.

- Soil Moisture Sensor - VH400 [61]

  To collect soil moisture data, VH400 Soil sensor, a low-cost sensor probe, is used. It measures volumetric water content (vwc) and outputs a voltage

proportional to the moisture level. Since the probe measures the dielectric constant of soil using a transmission line, it is insensitive to water salinity and will not corrode over time. The sensor can be used in irrigation and sprinkler systems, moisture monitoring, water conservation, etc.

The VH 400 takes supply voltages 3.5 V to 20 V DC and outputs related moisture content from 0 to 3 V. When activated, the senor consumes less than 7mA current, which is perfect for a low-power, embedded system. To convert the ADC reading value to vwc, we used the formula above, from the sensor datasheet. The curve can be approximated with four linear segments:

$$SoilMoisture(vwc) = \begin{cases} 10 * V - 1, & \text{if } 0 < Voltage \leq 1.1 \\ 25 * V - 17.5, & \text{if } 1.1 < Voltage \leq 1.3 \\ 48.08 * V - 47.5, & \text{if } 1.3 < Voltage \leq 1.82 \\ 26.32 * V - 7.89, & \text{if } 1.82 < Voltage \leq 2.2 \end{cases}$$

- Soil Temperature Sensor Probe - THERM200 [62]

  To obtain the soil temperature, a THERM200 soil temperature sensor probe is used. THERM200 is a soil temperature probe that has a temperature span from -40°C to 85°C. It outputs a voltage linearly proportional to the temperature to calculate the temperature from voltage. It is highly accurate with a 0.125°C resolution. THERM200 is compatible with data loggers and other wireless sensor applications.

  The sensor is powered from 3.6 V to 20 V DC and outputs a voltage of 0 to 3 V, where 0 represents -40°C and 3 V represents 85°C. To convert the ADC reading value to soil temperature, the following formula is used from the datasheet with unit of °C:

$$SoilTemperature = 41.67 * V_{out} - 40$$

5.1.3 Hardware Configuration

Figure 5.1 illustrates the hardware configuration of a single sensor node in which the MicaZ mote is mounted on top of the MDA300CA data acquisition board. Up to 7 sensors can be connected to the analog ports of MDA300CA. To enable the data bus of I2C, two 10 kΩ resistors are needed to connect the pin DATA with VCC and pin CLK with VCC, as shown in the figure.

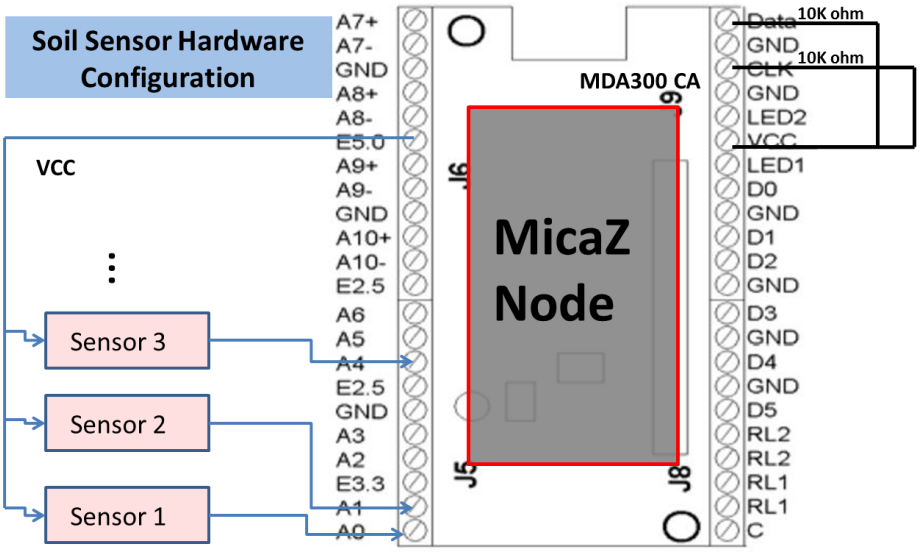


Figure 5.1: Hardware configuration of soil moisture sensor node

5.2 Software System

5.2.1 System Environment

The application is written in NesC under the TinyOS 2.x environment. TinyOS and its programming language NesC are specifically designed for embedded systems such as wireless sensor networks. They support event-driven concurrency mode and use modular, inference-driven design. The modular or component design provides a mechanism for structuring, naming, and linking software components into a robust embedded system. Each component provides and uses interfaces that are the access points of that component. The bidirectional interface declares a set of functions that can be categorized

into two types: *commands* and *events*. A *command* is a type of a function that an interface provider must implement, and an *event* is a function that the user must implement. The modular components provide a higher level of abstraction of hardware management and serve as basic building blocks for specific applications. This open-source feature gives the applications more flexibility and lets the developers create more complex designs.

5.2.2 Software Implementation

From a high-level view, each module used in our application can be categorized as being one of four types: Sensing, Wireless Communication, Control Unit, or Logging Unit. Figure 5.2 illustrates components and their interface functions used in each category.

SensorMDA300 and LogStorage components are used for sensing and logging purposes. SensorMDA300-implemented functions read the analog and digital readings at specific ports and return readings in hex format. Since the sensors draw power directly from the sensor node battery, the on/off state of the 5 V supply power line is controlled by the software via the power commands interface as well. LogStorage abstraction supports reliable (atomic) logging of events so as to survive the hardware power cycle. In our application, circular logging is used; that is, when storage is full, the least recently written data can be overwritten by new data.

Control Units are responsible for overall internal logic between components, such as scheduling the tasks and controlling the interrupt timing. At boot-up, the system will first initialize the scheduler, then it will initiate various other components, including timing and sensing units. A signal with error code is sent to the scheduler once the system has booted up. For the wireless communication part, AMsendC is used to send packets, and ActiveMessage is used for the acknowledgment mechanism and split control.

5.3 Evaluation

In this section, the experimental results of our sensor node are presented from both the hardware functionality and the effectiveness of reactive sensing aspects. A real experiment on the soil was performed to test the functionality

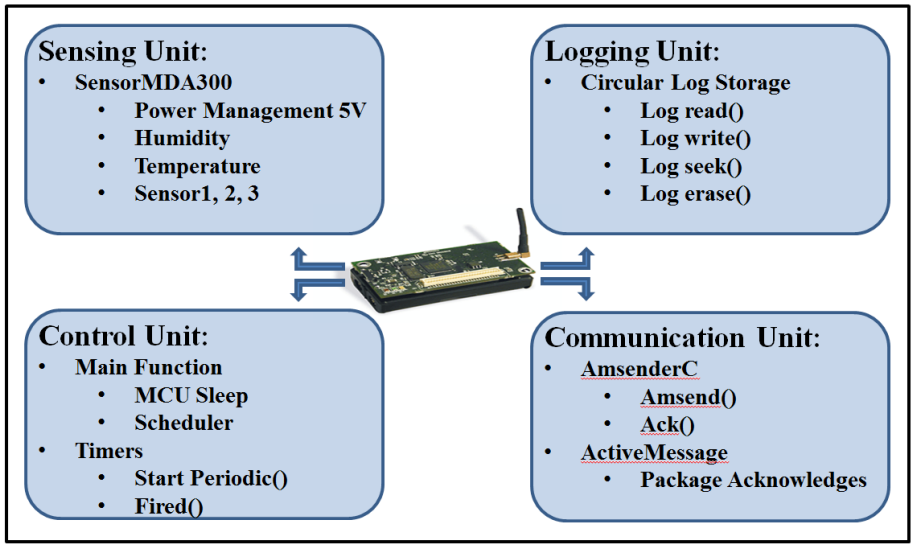


Figure 5.2: Software configuration of soil moisture sensor node

of the sensor node and responsiveness of soil moisture sensors. As expected, the sensor node is able to collect data at different soil depths (5 cm and 10 cm), and the soil moisture sensors are responsive to changes in soil moisture values. A second experiment was conducted on an ICN network dataset to evaluate the effectiveness of the sensing algorithm. Three years of hourly soil moisture data (around 24*365*3=26,280) from the DeKalb station were applied to show that our algorithm can improve the energy efficiency and still capture the dynamics of soil moisture in a timely manner.

5.3.1 Testing in the Soil

An experiment was conducted on the field soil to test the responsiveness of our sensor and functionality of the sensor network. In the experiment, two VH400 soil moisture sensors were attached to the sensor node, as shown in figure 5.3, and buried completely under the ground surface at 5 cm and 10 cm depths. The soil sample was taken from an open corn field and transferred to a bucket container. The hardware was configured such that the *maximal sample interval* was 40 seconds and the *granularity level* was 0.08 vwc. The adjustment of the sampling interval was based on the reading at the 5 cm depth. At each iteration, the data were read once instead of three times before being sent to the base node. This was done to show that errors can

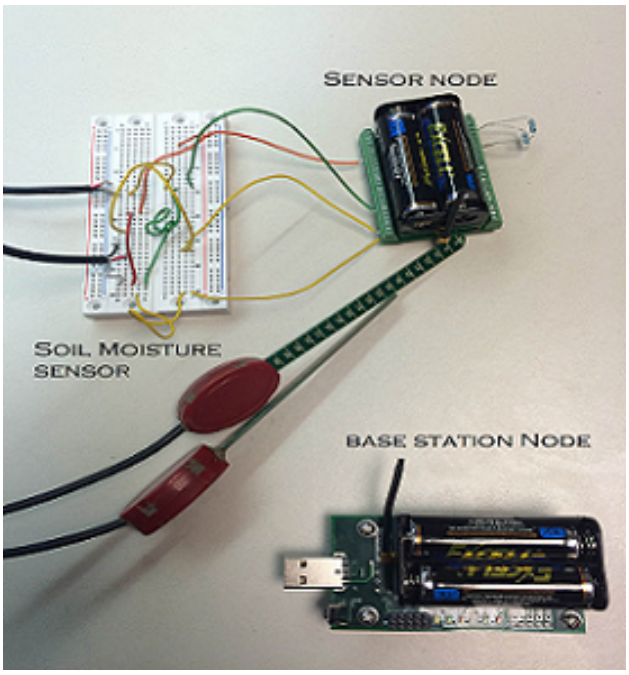


Figure 5.3: Physical appearance of the sensor node

occur during sensor power-on stage. A Java program was running on a PC concurrently to receive raw data from the base node. The job of the Java program was to take hex ADC data, parse the hex value, convert to the proper units and write to a log file in real time. The experiment lasted an hour in a lab setting, and results are shown in figure 5.4.

The red line (star) represents the soil moisture readings at 10 cm, and the blue line (circle) represents the readings at 5 cm. The first readings from both sensors are errors due to sensor warm-up stages. Such errors should not occur once the sensor takes three conservative readings before sending data to the base station. At point 30 (indicted by magenta diamond), a small volume of water was poured into the bucket. At point 60 (indicated by black upward-pointing triangle), a large volume of water was poured into the bucket.

The sensor node was able to collect data at different depths and reactively adjust the sampling interval. When the small volume was poured into the bucket, only the soil at 5 cm depth was saturated with water, and soil moisture at 10 cm stayed at the same level. When a large volume was applied, the soil at both levels got saturated. On the 5 cm depth line, the frequency of sampling increases at times 5 and 35, due to our reactive algorithm.

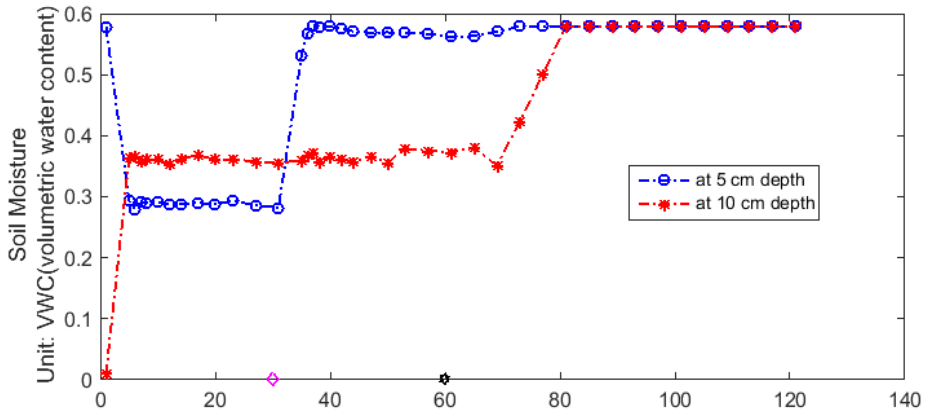


Figure 5.4: Sensor node performance on soil bucket

5.3.2 Evaluation at Historical Data Set

Historical data from the DeKalb station were used for evaluating the effectiveness of the reactive sampling algorithm in open-field soil moisture collection. The data were real soil moisture data at depth 5 cm from 2009 to 2011, containing 17,520 data entries at a per-hour granularity. The results show that the reactive sampling algorithm is able to effectively collect soil moisture dynamics.

Two static sample intervals and our reactive sampling algorithm were applied and compared on the same dataset to check their performance on capturing soil moisture changes. In method 1, a reactive sampling algorithm was used with the *maximal sample interval* set to 12 hours, and the *level of granularity* set to 0.03 vwc. In methods 2 and 3, the samples were taken every 12 hours and 4 hours, respectively. For each new data sample, we computed the difference between the new sample value and its previous value and then compared it with the *level of granularity* value. For each run of a specific method, the number at the respective *Granularity_Distance* field was increased by one on each sample iteration. *Granularity_Distance* is defined in the following formula. For example, if the difference between two consecutive readings is within the *level of granularity*, the *Granularity_Distance* is 0.

$$Granularity\_Distance = \lfloor \frac{abs(new\_sample\_value - previous\_value)}{level\_of\_granularity} \rfloor$$

In general, a good sampling method should produce sample points such

that two consecutive sample points do not have a very large variation. Ideally, all the difference values should have *Granularity_Distance* equal to 0. In this way, data is increased or decreased gradually, and fine granularity of data is preserved. It should be noted that two similar consecutive readings do not guarantee that any intermediate values are similar. But due to the characteristics of soil in an open field, we assume that this is unlikely to happen.

Table 5.1: Distribution of *Granularity_Distance* for different sampling methods

Gran_Dist*	**Method#1****	**Method#2**	**Method#3**
0	**1497**	**1302**	**4234**
1	98	103	88
2	17	25	18
3	13	10	14
4	8	9	10
Total	1633	1449	4364

* Gran_Dist stands for *Granularity_Distance*.
** Method#1 is reactive sampling at 12 hours, Method#2 is static sampling at 12 hours, Method#3 is static sampling at 4 hours

It is confirmed that method 1 is better at capturing dynamics readings than static methods in terms of aggregated *Granularity_Distance* number at each row from Table 5.1. The number at *Granularity_Distance* = 0 increases by 195 from 1302 to 1497, and the number at other *Granularity_Distance* fields decreases. The reactive algorithm only takes an additional 184 samples (1633-1449=184) in total, which confirms our assumption that soil moisture does not vary greatly in a 12-hour window. The result shows that due to the reactive algorithm, more variations are captured with fine granularity when soil moisture fluctuates. Method 1 is able to take about one fourth of the total sample operations of method 3 (from 16633 down to 4364) with little granularity loss. The majority of sample readings (90%) fall into the first row, where the difference is within the *Granularity_Distance* value. This indicates that the reactive sampling algorithm was not frequently triggered, and the sensor node sampled data at maximum sample intervals most of the time.

Figure 5.5 is taken from one subset of the data where soil moisture fluc-

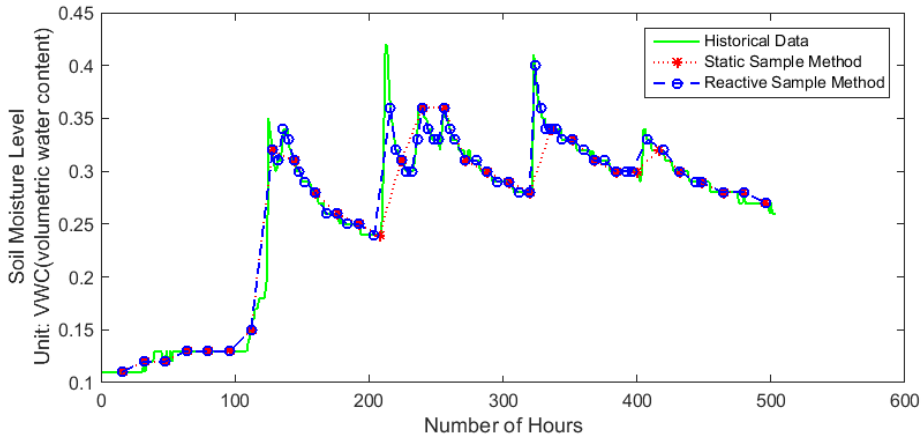


Figure 5.5: Algorithm performance on a subset of the data

tuates. This part of the data documents the hourly soil moisture variations within 500 hours (about 20 days). At the beginning and end of the graph, the reactive sampling method samples at the same rate as the static sampling method. In the middle of the graph, when there are large fluctuations of soil moisture level, reactive sampling is able to capture two peaks in the soil moisture level by increasing the sampling frequency, while the static method misses both peaks.

Chapter 6

SOIL MOISTURE PREDICTION SYSTEM

In this chapter, our soil moisture prediction system is introduced, followed by a detailed description of the features used in our machine learning algorithms. Then, two data-driven algorithms (SVM and RVM) are presented with a discussion of each algorithm's advantages and disadvantages. The experimental results on data from different sites are discussed in the next chapter.

6.1 Prediction System Overview

The objective of the system is to predict soil moisture at the root zone (from 5 cm to 50 cm) using data-driven modeling tools (support vector machine and relevance vector machine) based on meteorological data. As discussed in Chapter 2, root zone soil moisture level at 5 cm to 50 cm plays a vital role in crop growth, since most of the water is extracted within this range. The experimental data are parsed from the Illinois Climate Network database and fed to the models for training and validating purposes. The dataset consists of data from 9 independent sites. The trained model takes meteorological parameters including temperature, humidity, wind speed, solar radiation, precipitation, and soil temperature, together with the previous day's soil moisture values, as inputs. The output is the soil moisture value for the current day. The input and output of the machine learning algorithms are illustrated in figure 6.1.

A time-series framework with a feedback loop is built on top of the models to predict soil moisture in a longer time window. The model itself outputs soil moisture prediction in a 1 or 2 day time window, which is not sufficiently useful to farmers. Running the model n times using output from previous iterations as soil moisture input at iteration n, the predicted value at day n

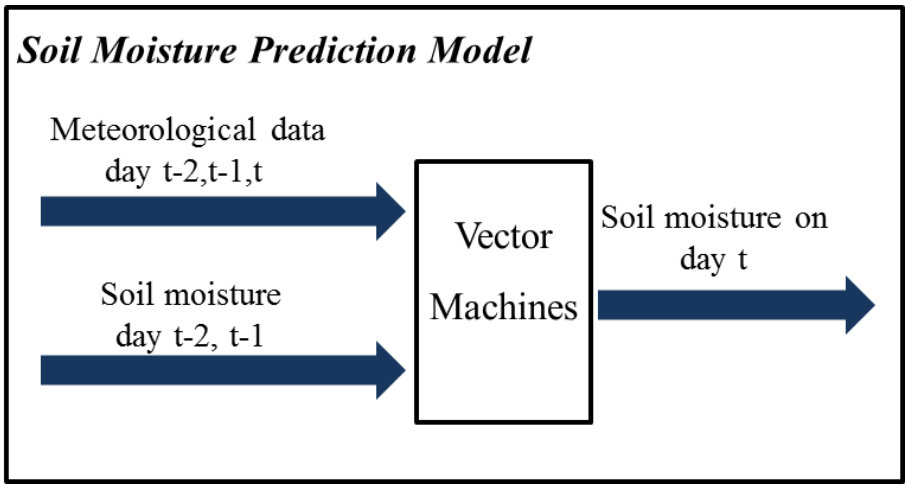


Figure 6.1: Vector machines input and output

can be obtained. More details about this framework are covered in section 6.1.2.

The system is specifically designed from the Precision Agriculture perspective, with site-specific modeling and the ability to integrate external data at the granularity of a day.

- Site-specificity is necessary in order to accurately model the soil moisture variations. Our time series plots from figure 6.2 show that sometimes even close sites can have quite different soil moisture patterns. Site-specific modeling allows the most accurate model to be obtained.

- The feature of including data from other sources makes the system more robust by allowing other reliable data to be considered when generating output. The model is flexible, as users can control the inputs at the granularity of a day. Unlike soil moisture as studied in hydrology, where variations in soil attributes come mainly from environmental changes, cropland is regularly maintained by people. The variations can come from both meteorological changes and human actions. This requires the system to be flexible in terms of variables. Also, because it is farmland, there are probably more similar types of data collected via other reliable measurement methods that are available for use. For example, the external data in our work can come from weather forecasting for meteorological parameters. For soil moisture, the data can come from other sources, such as sensor-based devices or remote sensing images.

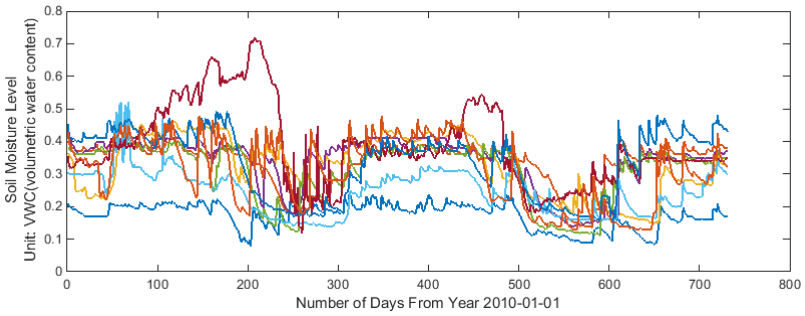
As a result, when designing models, one must take these factors into consideration to improve overall model accuracy.

We cover more details regarding these two features in the next two sections.

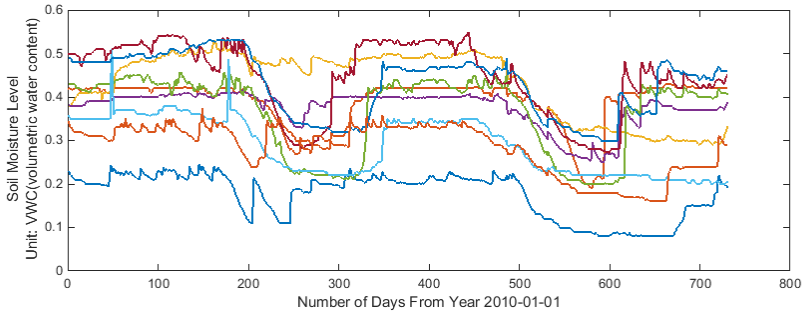
6.1.1 Site-Specific Modeling

Variations from both the soil's physical attributes and environment factors serve as major challenges to reaching a generic soil moisture model. In the past, various physically based models [63, 64] were proposed and evaluated. However, as pointed out in [14], the major impediment to obtaining a physically based model is that gathering physical parameters, such as soil chemicals and pH, can be difficult. On the other hand, environmental information has been collected and studied for decades and is relatively easy to gather. Soil moisture varies both in space and time because of spatial and temporal variations in the environment. Strategic site modeling is necessary so that the soil's physical attributes are similar across model locations. As a result, the algorithm can produce models that focus on meteorological parameters that affect soil moisture in making spatio-temporal predictions.

Furthermore, analysis of the ICN dataset shows that even with similar soil attributes, generating a generic soil model can be very difficult. Figure 6.2 shows comparisons of soil moisture at different depths at 9 sites in a time series. The data were gathered in bare soil for all the sites. The dataset contains soil moisture data from 2011 to 2012 at depths of 20 cm and 50 cm. It is shown that even though the meteorological parameters, such as temperature and precipitation, and physical parameters, are very similar at all locations, the soil moisture can still vary significantly between some of the sites. In figure 6.2a, while most sites share similar patterns throughout the year, the lowest curve and highest curve show significantly different patterns from the rest. At a 50 cm level in figure 6.2b, a pattern may exist among sites, as the trends are similar. But there is always a certain level of variation (offset). Based on our observation, the greater the soil depth, the more variations may exist between two sites. As a result, we believe that it is impossible to find a "one size fits all" soil moisture model; and the model should be site-specific.



(a) Time series plot of 9 sites at depth 20 cm



(b) Time series plot of 9 sites at depth 50 cm

Figure 6.2: Time series plot of 9 sites

6.1.2 Inclusion Data from Other Sources

A unique time-series model with a feedback loop can be structured for the purpose of predicting several days ahead, as shown in figure 6.3. To estimate the soil moisture value on the current day is straightforward: one runs the prediction model once. To retrieve the prediction of soil moisture n days ahead, one needs to run n iterations of the model with the soil moisture input at iteration k being the output at iteration k-1 and k-2.

The introduction of this time-series model creates the opportunity for users to manipulate inputs at any iteration. At each iteration, the input of environmental parameters can come from forecasting values or user-provided values. The input of soil moisture can be either data from other soil moisture retrieval techniques or previous predicted values. For example, a common practice in agriculture is to predict the future soil moisture value if the drought situation stays n more days. By setting the precipitation value to zero, our model is able to make a prediction under the assumption that the drought condition continues

A more reasonable prediction can be made by leveraging knowledge from

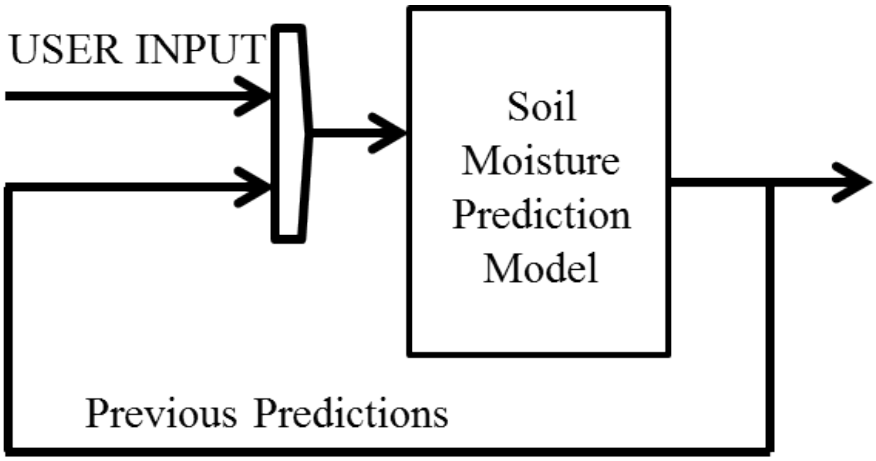


Figure 6.3: Feedback time series prediction

other sources at input fields. In previous work [14, 15] on soil moisture prediction, the model is fixed and only can predict values k days ahead, where k is a fixed number. The prediction at day $t+k$ (k days ahead) is based on soil moisture values and meteorological data at day t, day t-1, day t-2, and so on. The prediction result is unreliable as it disregards the meteorological data between day t and day $t+k$. A heavy precipitation between day t and day $t+k$ can make the result irrelevant and useless. With advances in weather forecasting technology, the forecasting of meteorological data has become more accurate and fine-grained, and it is available to the public. Our system is able to include meteorological data between day t and day $t+k$ from weather forecasting. On the soil moisture input side, readings from other soil moisture measurement methods can be integrated to further improve accuracy. Currently, soil moisture is measured by farmers at intervals of about 15 to 30 days. Physical measurement offers great accuracy in the cost of resources and human labor. The near absolute correct value from sensor measurement can be used as the input to correct the model, as the model result tends to "drift-away" from the ground truth after several runs. The evaluation of the above two methods for predicting soil moisture is presented in Chapter 7.

6.2 Features Selection

In this study, the data are obtained from 19 stations from June 2004 to the end of the 2012. The following parameters are selected as features in the machine learning process:

- Air temperature
- Relative humidity
- Wind speed
- Solar radiation
- Precipitation
- Soil temperature at depth 10 cm and 20 cm
- Soil moisture at depth 5 cm, 10 cm 20 cm, 50 cm

The data are parsed from two separate data sources after features are selected. The first data source contains weather parameters and soil temperature data and the second one contains soil moisture data. For the first data source, there are approximately 5,600 data points for each site. As a result, the first data source contains approximately 106,400 (5600 x 19) data points for weather information with granularity of one day. The second data source (soil moisture) has one hour granularity. There are roughly 100,000 data points, and thus approximately 1,900,000 (100,000 x 19) total data points for soil moisture information were parsed.

For each parameter included in the two data sources, average value and maximum and minimum values are given for each data category. If the data are missing or contain errors, an "error/missing" flag is set to positive. Where appropriate, some data are estimated from an adjacent station's data or interpolated from an adjacent time at the same site with an "estimated" flag set.

6.3 Preprocessing

To train the model, raw data from ICN needs to be preprocessed and put into machine learning format. A set of code scripts are written to operate on the raw data. The flow shown in figure 6.4 is applied to all 17 site data from ICN. The main tasks of the data preprocessing stage are:

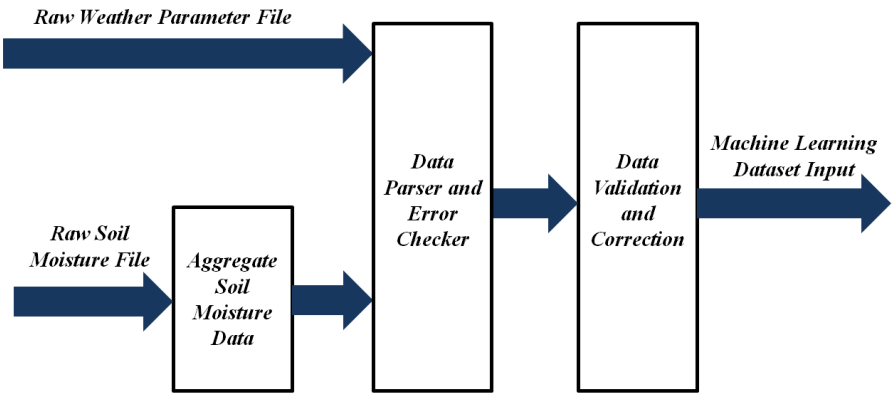


Figure 6.4: Raw data preprocessing stages

- Aggregate raw soil moisture hourly data into day granularity by averaging hourly values.
- Parse features data from two data sources with matching dates.
- Detect various errors and missing value problems from feature dataset.
- Validate and correct detected errors.

6.3.1 Data Parsing and Error Checking

The job of our data parser is to parse selected features from both weather files and aggregated soil moisture files. For weather data from 17 sites, the parser parses average values of wind speed, air temperature, relative humidity, soil temperature at 10 cm and 20 cm, total solar radiation, precipitation, and date entry. The values are set to a negative number if the data are estimated or are missing based on error flags.

One of the problems that the parser deals with is data format inconsistency. Three types of data formats appear across different locations in the raw data set. Since the date of the soil moisture data needs to match with the weather data of the same day, formatting the date entry into a unified format is necessary.

An error checking script is responsible for checking possible errors in the files and possibly correcting them. It is common to find different types of "error" or "missing" columns in the raw dataset. An error log file is generated for each site, which indicates the line locations and types of errors for manual

inspection. Although errors are only a small fraction of overall data, they can distort the format of raw data, making parsing more difficult. Some of most common errors found in the two datasets are:

- Missing column entry
- Missing data of several consecutive days
- Duplicated data entries at same site location

6.3.2 Data Correction

With the help of the error log, generated output files are manually checked at the line locations indicated in the error log. If the error is "duplicated data entry," we simply delete the duplicate. If the error is about a missing column or missing data, linear interpolation is applied to generate the missing data.

Out of 17 sites, 9 low-error-rate sites are selected as source inputs for the machine learning. Although almost every site's data contains different types of errors, some missing data entries hold for a substantially long period of time (more than a month) and thus are labeled as bad inputs and filtered out. For instance, in the soil moisture data from Monmouth data from 2009-Apr-29 to 2009-June-3 are absent. This can make the predication models inaccurate and should be eliminated. Among all 9 selected sites' data, the average missing period is around 1-3 days for each "missing" error.

Another major problem that leads to site data being filtered out is data structure inconsistency. Within one file, the numbers of columns can be different and the date format can be different. These kinds of problems increase the difficulty of writing parser code and detecting errors. They also decrease confidence in the data integrity generated by the parser, as the data can belong to a neighboring column field. To ensure the data used for machine learning is correct in format and structure, we select 9 sites whose data have few or none of the above problems.

6.4 Learning Algorithms

Support vector machine (SVM) and relevance vector machine (RVM) techniques are used to build mathematical models to predict soil moisture content

based on previous years' collected data. This section provides some background on the two vector machine learning techniques and their respected regression models.

6.4.1 Support Vector Machine

A support vector machine constructs a hyperplane in a kernel space that has the largest functional margin between two separable classes [9]. In the case of a binary classification problem, given a set of training data $[x_1, x_2, ..., x_l]$ where $x_i \in R^k$ and k is the number of features, the output is prediction value y_i where $y_i \in \pm 1$. The idea of SVM was first introduced by Vladimir Vapnik and his co-workers in 1979, and the main paper [65] was published in the mid-1990s. Since then, there have been many variations of SVM techniques developed, and SVM has been widely adopted as a statistical learning tool. The algorithm is statistically sound, as the objective of SVM is to find a dependency function $f(x)$ that can maximize the margin between examples. The margin is defined as the closest distance between a positive case and a negative case, which is also called the *Euclidean distance* and derived to be $\frac{2}{|w|}$. The objective is then to minimize the following equation if the class can be perfectly separated:

$$\begin{aligned} \min \quad & \frac{1}{2} w^T w \\ s.t. \quad & y_i(w^T(x_i) + b) \geq 1 \end{aligned}$$

where w is the normal vector to the hyperplane.

However, it often occurs that there are some outliers in the dataset that make the data inseparable. In order to use SVM, the error tolerance parameter ξ is introduced to manipulate the errors so that they become tolerable to the algorithm. The objective function is transferred to the following equation, where ξ denotes the maximum errors tolerance of the algorithm:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \\ s.t. \quad & y_i(w^T(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \ldots, l. \end{aligned}$$

The ξ value makes the solution sparse by ignoring any error less than ξ. As a result, the solution is called ξ-*insensitive*. The quantity C represents the trade-off between the complexity of the function and the amount of error allowed. The larger C is, the less sensitive the algorithm is to error. After setting the objective function, the solution can be obtained by solving in dual form and using Lagrange multipliers.

The solution is sparse, since it contains only a subset of the overall training vector, called *support vectors*. The algorithm selects a set of data points to represent the boundary, and the prediction value is based on the following formula:

$$f(x) = \sum_{i=1}^{l} w_i K(x_i, x) + b$$

where w_i is the weight vector, K is the kernel function, and b is the bias.

Support vector regression is similar to support vector classification, but the output will be numeric instead of nominal. The underlying principle of regression mode is similar to the SVM classification, which is to minimize the errors by maximizing the margin. There are two types of SVR: epsilon-SVR and v-SVR. The difference between the two is in how the problem is parameterized. While both use the same loss function, v-SVR gives a more meaningful interpretation of error bound and number of support vectors to users. But given appropriate parameters, the same problem is solved in both regression algorithms.

6.4.2 Relevance Vector Machine

A relevance vector machine is another vector machine learning algorithm that shares similar modeling with SVM. The model adopts a probabilistic framework by incorporating a Bayesian treatment. The motivation is that

in a real-world data collection process, noise will appear in the dataset, and the prediction solution tends to be *overfitting*. To overcome this problem, we assume in the probabilistic formulation that the targets are sampled from the model with additive noise:

$$t_n = f(x_n; w) + \epsilon_n$$

where $f(x_n; w)$ is the same function for SVM; ϵ_n are independent from samples and normally distributed with mean-zero and variance σ^2.

In this probabilistic framework, a prior is imposed over the model weights that are governed by a set of hyperparameters, one associated with weight. The RVM process is an iterative one that repeatedly re-estimates the hyperparameters. The RVM model is able to produce a generalized model while utilizing fewer support vectors by applying parameter approximation. In practice, most of the posterior distributions of weights are centered around zero with a sharp peak. By setting those weights to zero, the solution becomes sparse and those non-zero weights are named *relevance vectors*. More detailed description of the relevance vector machine algorithm can be found in the original paper [10].

6.4.3 Discussion of SVM and RVM

For both vector machines, the use of kernel functions increases the performance, as inseparable data in linear space can become separable or closer to separable in kernel or feature space. The data are mapped onto other dimensions or feature spaces instance-by-instance before being used in the vector machine algorithm. The common kernel functions used in a vector machine are linear, polynomial, radial basis function, and sigmoid.

The support vector machine has a solid theoretical basis when building the model from a dataset. The use of a kernel allows one to map data into a higher dimensional space when drawing margins. Thus, SVM can be more flexible in finding an approximately perfect separate line. Unlike a neural network, SVM delivers a unique solution by solving a global optimization problem, which ensures that the solution is not trapped in local minima [15]. The use of parameter C can further allow one to control the trade-off between algorithm complexity and error tolerance. However, the disadvantage of SVM

is that the training phase is slow when the dataset gets large. The best result is obtained through trial and error on different combinations of C, epsilon, and other parameters, since SVM lacks results transparency.

Compared with SVM, RVM is able to produce outputs using fewer vectors (data points) from the training dataset. The number of support vectors generated from SVM usually grows linearly with the size of the training set. When applied to a large dataset for modeling, RVM solutions can greatly reduce computational complexity. The kernel function must satisfy Mercer's condition in an SVM, but not in an RVM. Some types of kernel function used in RVM can be Gaussian kernel, Cauchy' kernel, cube distance, and distance kernel.

However, since inferring the function parameters in the training phase requires performing an inverse operation on a covariance matrix, the computational complexity is $O(N^3)$. Compared to SVM, RVM requires longer training time, and the math model is more complex than that of SVM.

Chapter 7

PREDICTION MODEL EVALUATION

In this chapter, the performance of the soil moisture prediction model is presented. The mathematical models obtained through SVM and RVM algorithms are evaluated under two applications. The first application forecasts soil moisture several days ahead using the data from 2012. The second application estimates the soil moisture consecutively, and the predicted values are corrected every 45 days. Both of the applications show a strong prediction result with high correlation factor and low error rate.

7.1 Methodology

The method of obtaining models is described in the previous chapter where the meteorological data and soil moisture data are parsed separately from files and preprocessed to form a valid input set for the machine learning model. After the preprocessing stage, we are able to retrieve data from nine independent sites across Illinois from 2004-01-01 to 2012-12-31. For each site, there are about 3288 data points, each representing average values of meteorological parameters for the day.

The data are normalized between 0 and 1 and split into testing, training, and validating data, in order to properly train the model. The following experiments are conducted on testing data that have not been used or seen by the model. The testing data include a total of 365 data entries for 2012. The rest of the data are split into training data and validation data. The ratio of training to validating data is about 80:20, a common ratio in machine learning. Based on observations, a fairly good model can be reached with year-round data points, and the model becomes more accurate with each increment in training data size. A radial basis kernel is used for both vector machines, and the rest of the parameters are selected based on trial and error.

The experiment scripts are developed on top of the vector machine package provided in LIBSVM [66] and SparseBayes [67].

The metrics used for evaluating models are mean squared error (MSE), mean absolute error (MAE), and correlation coefficient (R^2). The MSE computes the average of the squares of the errors of model prediction from the actual value. MAE represents the mean value of the absolute error of the predicted value from the target value. Correlation coefficient R^2 measures the linear relationship between predicted value and actual value. R^2 can range from 0 to 1, with R^2 =1 being a perfect match and 0 meaning zero correlation between the two. Equations for obtaining MSE and MAE are as follows:

$$MSE = \frac{\sum_{i=1}^{n}(t_i - p_i)^2}{n}$$

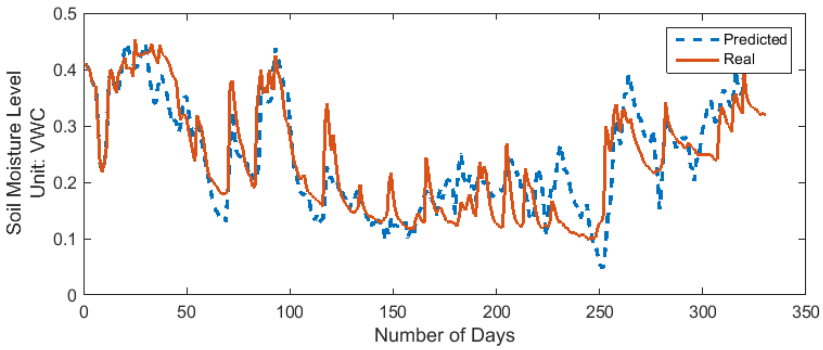
$$MAE = \frac{\sum_{i=1}^{n}|t_i - p_i|}{n}$$

where t_i and p_i are target value and predicted value, respectively, and n is the number of testing data points.

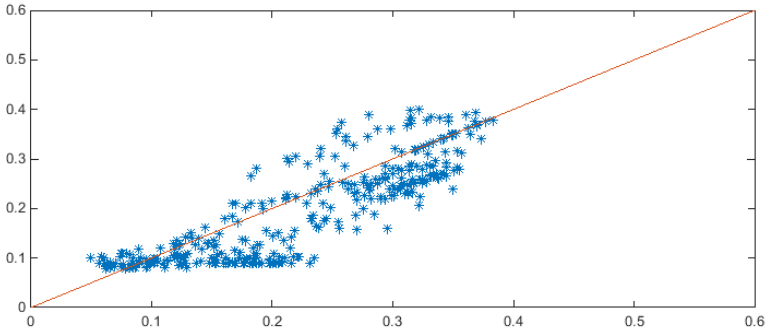
7.2 Soil Moisture Forecasting

One of the common practices in agricultural planting is to forecast the soil moisture values. Forecasting is often done at the seeding stage, where the ideal seeding time needs to be determined. With an accurate forecasting system, warnings or alerts can also be generated advising farmers to take proper actions and prepare several days ahead. In this experiment, we evaluate the performance of model forecasting by predicting soil moisture content at i+15 days, where t is the current date. The testing dataset is applied on the properly trained SVM and RVM model obtained by the methods described in the methodology section. Each forecasting result is produced by running trained models 15 times, given that the forecasting weather information is perfectly accurate. Table 7.1 provides the statistics of the forecasting result compared with actual values across nine different locations at depth 5 cm. The results at 10 cm and 20 cm show similar trends for forecasting up to around 20 days.

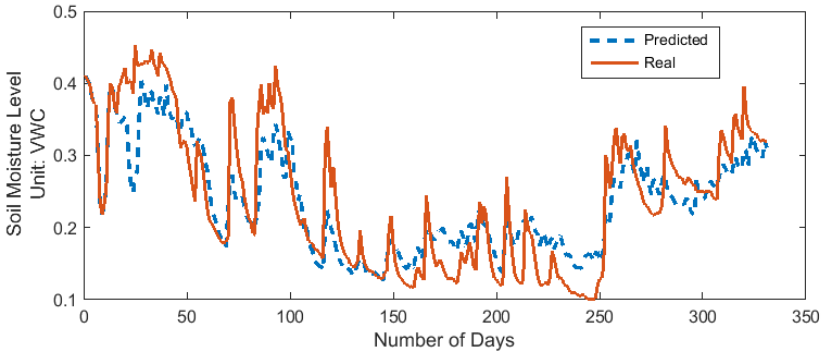
As shown in Table 7.1, both algorithms are able to produce a site-specific



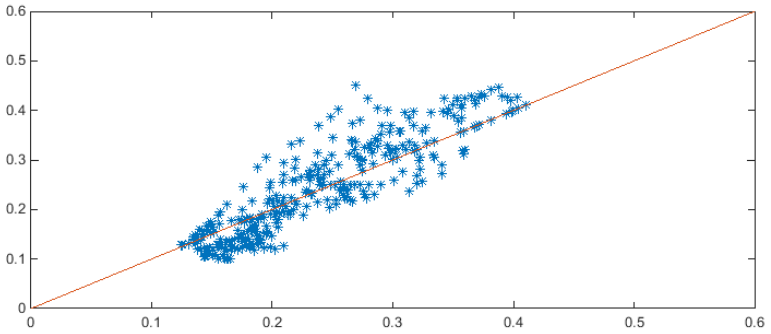
(a) RVM forecasting in time series plot



(b) 45 degree line plot of RVM



(c) SVM forecasting in time series plot



(d) 45 degree line plot of SVM

Figure 7.1: RVM and SVM forecasting performance

Table 7.1: Soil moisture forecasting result

	MSE*(x100)		**MAE*(x100)**		R^2(%)	
	RVM	SVM	RVM	SVM	RVM	SVM
Site 1:	0.20	0.06	3.30	1.75	82.7	85.0
Site 2:	0.18	0.11	3.27	2.63	96.1	97.2
Site 3:	0.26	0.21	3.91	3.64	93.0	94.7
Site 4:	0.49	0.12	4.15	2.42	89.3	96.9
Site 5:	0.35	0.19	4.51	3.64	94.9	98.5
Site 6:	0.18	0.22	3.30	3.25	95.5	94.5
Site 7:	0.28	0.13	4.27	2.76	93.9	97.3
Site 8:	0.33	0.09	4.15	2.13	92.6	97.4
Site 9:	0.32	0.13	4.54	2.85	92.6	97.3
Average:	0.29	0.14	**3.93**	**2.78**	**92.3**	**95.4**

*Note: The statistics of MSE and MAE values are actual values times 100 for display purposes

model that captures the underlying relationship between inputs and outputs. The mean absolute errors across nine sites are around 0.039 and 0.028 for RVM and SVM, which means the error rates are less than 15%. From the obtained average R^2 values, it is shown that there exist strong linear correlations (above 92%) between the predicted values and the actual values. Figure 7.1a and figure 7.1c plot the performance of RVM and SVM algorithms in time-series from one of the sites. Figure 7.1b and figure 7.1d show prediction values with respect to testing values for the same site on 45-degree lines for RVM and SVM. The X and Y axes in the figures represent the predicted values and real values, respectively. As demonstrated in the figures, the predicted values are consistent in terms of error margins and are tightly centered around the 45-degree line with only small error offsets. Compared with SVM, the RVM algorithm requires fewer support vectors to produce the result. The support vectors or relevance vectors used in RVM are less than 10% of total input vectors, while SVM needs about 50% of input vectors. As a result, the RVM solution is sparser than the SVM one. Even though the performance of SVM is slightly better than RVM, the difference is insufficient to consider it a better algorithm.

To assess the performance of adding forecasting information to previous

approaches, a comparison experiment was also conducted on the same dataset using the approach of previous work [14, 15], where the model predicts values k days ahead based on current meteorological data without forecasting information. The mean absolute error aggregated for nine sites is 0.06 with a low correlation coefficient (81%). Compared to their approach, our models are able to lower the error rate by one-half and obtain a high correlation coefficient. It should be noted that here we are assuming the weather data is perfectly accurate, but in actual practice the forecasting values may not be so accurate. Applying real noise-included weather forecasting data to models can be a subject of future work.

7.3 Soil Moisture Estimation in Time-Series

Aside from making predictions, another experiment is conducted on our system to evaluate its performance in making consecutive estimations of soil moisture. Since the model can take inputs from its previous estimation result, the estimated values probably tend to "drift away" from the ground truth. The purpose of this experiment is to measure how the estimation may vary from the actual values over long time periods. Further, unlike the subjects of hydrology study, agricultural lands are regularly managed by people. In real practice, modeling data play supporting roles in decision-making and probably are not the only source for monitoring soil moisture. Point measurements from sensor devices or other methods offer accuracy but are costly in terms of labor and resources. Our system is able to correct its errors by integrating those accurate measurements as inputs. In this way, farmers can still preserve fine-grained soil moisture data without frequently measuring soil moisture in the field.

Depending on the specific application requirements for soil moisture accuracy, one may set various time intervals for correcting estimation values. Table 7.2 lists the experimental results of our models with data corrected every 45 days at depth 20 cm using SVM. Two strategies of estimation are tested using the same trained models in which method 1 (No CRT) makes consecutive estimations based on previously estimated data without correction, and method 2 (CRT) corrects data every 45 days. Results of running RVM at other depths from 5 cm to 50 cm show similar trends in performance.

Table 7.2: Comparison of time series estimation with or without correction

	MSE*(x100)		**MAE*(x100)**		R^2(%)	
	No CRT**	CRT	No CRT	CRT	No CRT	CRT
Site 1:	0.06	0.05	1.99	1.74	91.7	92.6
Site 2:	0.32	0.25	3.73	2.60	88.1	90.8
Site 3:	0.16	0.11	3.05	2.58	96.9	96.8
Site 4:	0.76	0.12	6.39	2.34	**78.8**	**96.2**
Site 5:	0.44	0.14	4.99	2.38	88.8	96.1
Site 6:	0.12	0.05	2.64	1.78	93.0	96.1
Site 7:	0.88	0.77	6.79	6.33	**79.0**	**93.7**
Site 8:	0.39	0.16	5.21	2.99	89.0	95.6
Site 9:	0.23	0.20	3.90	3.36	89.0	95.0
Average:	0.37	0.21	4.30	2.90	**89.0**	**94.8**

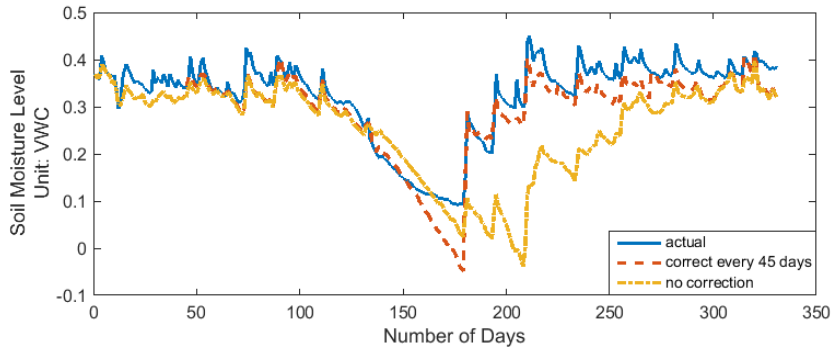
*Note: The statistics of MSE and MAE values are actual values times 100 for displaying purpose.

**Note: No CRT means the data are not corrected during the experiment. In CRT, predicted values are corrected every 45 days.

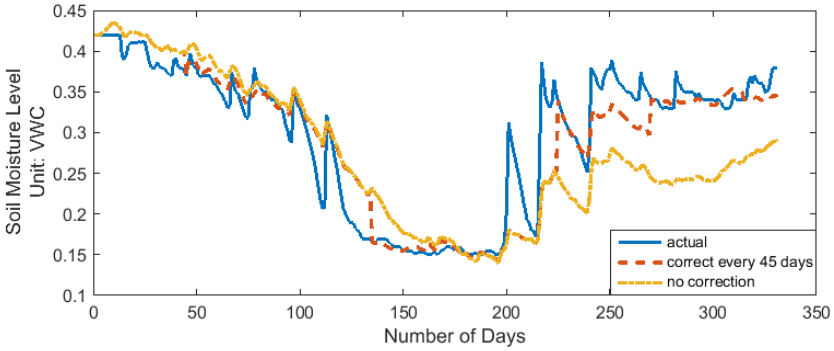
As shown in Table 7.2, even without correction, our models can still be responsive to the environment changes and remain fairly accurate over the course of a year. The overall average MAE is 0.043 (vwc) for method 1, and average correlation coefficient is 89%. Compared with method 1, method 2 shows improvements in terms of average correlation coefficient and MAE. However, it should be noted that in sites 4 and 7, the improvement gains are significant compared with other sites. Using method 2, the MAE can be reduced from 0.07 to 0.02, while R^2 increases from 78% to 95%. Based on our observations, the model can accurately follow the slope of soil moisture decay when there is no precipitation or when precipitation is small. However, a large error gap between predicted value and real value occurs when there is a sudden increase in soil moisture. This indicates that the models perform relatively poorly on days of heavy precipitation but are able to follow the decay trends of soil moisture fairly well. In practice, a dynamic correction strategy may be adopted where the models are corrected only when heavy precipitation occurs.

Figure 7.2 gives visual comparisons of the performance of method 1 and method 2 at depths 5 cm, 10 cm, 20 cm, and 50 cm. The solid lines represent

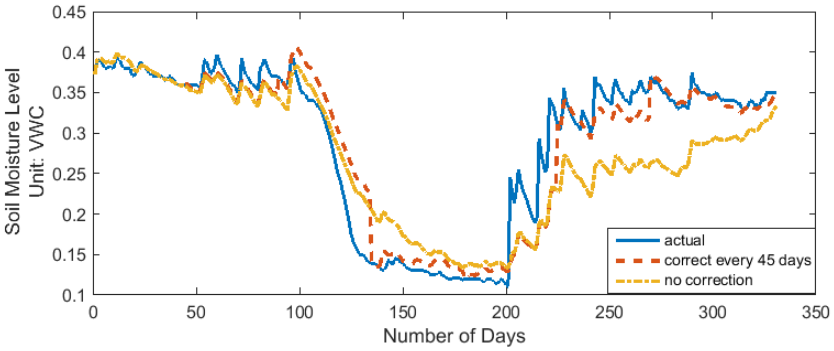
real values in 2012 from one of the ICN sites, the dash-dot lines show the performance of the No-CRT method, and dashed lines show the performance of the CRT method. The models perform well initially, when the trend of soil moisture is decreasing. Large error gaps between predicted and actual values occur when soil moisture content has a sudden large increase in the middle of the plot, particularly in 7.2a and 7.2b. Using the correction method, the predicted value is brought back to a reasonable value in a short period of time, while the large error gaps remain until the end in the no-correction method.



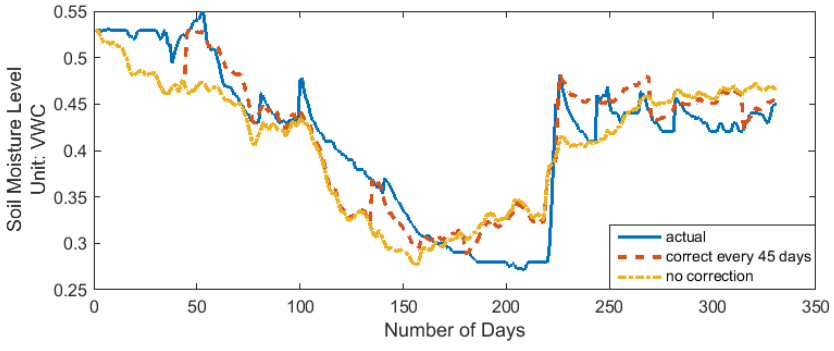
(a) No-CRT and CRT performance comparison at depth 5 cm



(b) No-CRT and CRT performance comparison at depth 10 cm



(c) No-CRT and CRT performance comparison at depth 20 cm



(d) No-CRT and CRT performance comparison at depth 50 cm

Figure 7.2: No-CRT and CRT performance comparison at all depths

Chapter 8

CONCLUSION AND FUTURE WORK

This chapter concludes the thesis work introduced in the previous chapters, followed by a discussion of future work.

8.1 Conclusion

In this thesis, a soil moisture collection and prediction system is designed specifically from a Precision Agriculture perspective using a data-driven approach. The solutions address two of the most important aspects of applying a data-driven approach: gathered data quality (i.e., the effectiveness of data collection process) and data analysis tools (i.e., the effectiveness of data interpretation). Precision Agriculture requires site-specific system that can incorporate data from other sources as well as from expert knowledge. In our system, a framework is presented on both the collection and prediction sides that allows the system to be made site-specific by incorporating user-specified inputs. Evaluations are done using years of historical soil moisture data from the Illinois Climate Network (ICN).

On the collection end, a wireless sensor network node is prototyped on an open-source platform TinyOS. The wireless sensor node is capable of collecting soil moisture data, soil temperature, temperature, and humidity. It can be further expanded to collect other types of information by attaching new sensors. The sensor node offers two user-defined variables to regulate the level of data granularity and sample intervals based on deployment requirements. An effective reactive sampling algorithm is proposed based on patterns retrieved from open field soil moisture dynamics. Compared with the static sampling approach, the reactive sampling algorithm shows its capability in capturing soil moisture dynamics while sampling at low frequency to save energy.

On the prediction end, machine learning techniques (SVM and RVM) are used for building prediction models based on types of data collected in the WSN system. On top of the machine learning model, a time series feedback loop is proposed that allows users to manipulate the model input fields at a granularity of one day. Experimental results measuring the model's ability to predict consecutively are presented in Chapter 7. The results show that our model is able to produce a prediction 15 days ahead with high accuracy (less than 15% error rate).

8.2 Future Work

In this section, we briefly mention other areas in agriculture that are worth investigating using data-driven approaches, followed by future work that can be done to improve our system.

More data-driven techniques can be explored on other attributes in the crop production cycle. In terms of physical parameters, factors such as fertilizer (nitrogen profiles and pH), pesticides, and herbicides all contribute greatly to the final cost of the crop. Site-specific methods to manage the use of fertilizer, pesticides, and herbicides can make farms more sustainable and lower the cost of crops. In terms of market information, data-driven approaches can be used to close the gap between farmers and consumers or to model future market prices for certain crops. Farmers can be more informed when making market decisions and realize better profits. Furthermore, data-driven approaches can also be used to give farmers recommendations on activities, such as when to plant seeds and when to apply fertilizers. Agriculture is a relatively old industry in which lots of human knowledge has accumulated. Data-driven techniques represent an advance in which activity is determined by data, not by human knowledge based on experience. Currently, farmers mostly rely on human experience to make decisions, and the opportunity cost of relying on data driven results is large for the agriculture industry. As a result, when designing a system to recommend farming activities, one should consider integrating the knowledge from human experience with data driven results to improve effectiveness of the recommendations.

Future work on our soil moisture collection and prediction system should focus on testing on a large scale and on security. Most of the work in this

thesis focuses on developing a reactive sensor node; the performance of the wireless sensor network when deployed at large scale is not tested. Future work can measure the performance metrics of different wireless network protocols and develop a network protocol suitable for our application characteristics. On the security side, sensor nodes are deployed in the open fields and share communication media. Since many automated systems rely on sensor data to make decisions, one needs to ensure that the data are reliable and accurate. Data can be tampered with at the sensor data collection phase and database storage phase. False information from the soil moisture sensors may lead to unnecessary irrigation operations which may drown the crops. A security layer needs to be added on top of the sensor messaging layer to prevent unauthorized sensor data from being inserted into the database. In real deployment, database security techniques should also be considered to protect databases against compromises. On the prediction side, in our prediction experiment, we assume the forecasting data is error-free. Applying real, noise-included weather forecasting data onto models can be done in future work. Lastly, more advanced techniques in machine learning can be explored to reduce the amount of data required to achieve a good model. Currently, obtaining a fairly good model requires at least one year of data. Methods that include spatial factors into the modeling and achieve a fairly good model with less data are potential areas for future research. Using machine learning to study the temporal-spatial variations within one site is worth exploring as well, as it may result in more fine-grained models.

Appendix A

TOP-LEVEL CONFIGURATION FILE

```
#include "SoilRead.h"
#include "StorageVolumes.h"
configuration SoilAppC {}

implementation
{

components MainC, SoilP, LedsC;
components new TimerMilliC() as MyTimer;

/* Data Acquisition Board Component file */
components new SensorMDA300CA() as Mda300;

components ActiveMessageC;
//components HplSleepC;
components new AMSenderC(AM_ADC_MESSAGE);
components  new LogStorageC(VOLUME_LOGTEST, TRUE),

SoilP.Boot -> MainC.Boot;
SoilP.Timer0 -> MyTimer;
SoilP.Leds -> LedsC.Leds;

SoilP.Packet -> AMSenderC;
SoilP.AMPacket -> AMSenderC;
SoilP.AMSend -> AMSenderC;
SoilP.AMControl -> ActiveMessageC;
SoilP.PacketAcknowledgements -> ActiveMessageC.PacketAcknowledgements;
```

```
SoilP.Sensor_0_Read -> Mda300.ADC_0;
SoilP.Sensor_1_Read -> Mda300.ADC_1;
SoilP.Hum -> Mda300.Humidity;
SoilP.Temp -> Mda300.Temperature;
SoilP.ex_5_V->Mda300.Excitacion_50;


//logger
SoilP.LogRead -> LogStorageC.LogRead;
   SoilP.LogWrite -> LogStorageC.LogWrite;
}
```

Appendix B

IMPLEMENTATION CODE

```
#include "SoilRead.h"

module SoilP {

uses {
interface Power as ex_5_V;
interface Read<uint16_t> as Sensor_0_Read;
interface Read<uint16_t> as Sensor_1_Read;
interface Read<uint16_t> as Hum;
interface Read<uint16_t> as Temp;
interface Packet;
interface PacketAcknowledgements;
interface AMPacket;
interface AMSend;
interface SplitControl as AMControl;
interface Boot;
interface Leds;
interface LogRead;
interface LogWrite;
interface Timer<TMilli> as Timer0;
//interface McuSleep as PowerManager;
}
}
implementation
{ //user supplied data
enum{
  MAX_INTRVAL=100000,// in mill sec
```

```
  MIN_INTRVAL= 20000,// in mill sec
 GRANULARITY = 0xC6,  // in hex unit
};
//data variables
uint16_t  SAMP_INTRVAL = MAX_INTRVAL;
uint16_t soil_mositure = 0;
uint16_t soil_temp = 0;
uint16_t air_hum = 0;
uint16_t air_temp = 0;
uint16_t counter=0;
//used for reactive algorithm
uint16_t prev_val=0;
uint16_t num_simi_read=0;
bool  reduce_intrval= FALSE;
bool  set_timer_flag= FALSE;

//logger buf
typedef nx_struct logentry_t {
    nx_uint8_t len;
    message_t msg;
   } logentry_t;
   logentry_t m_entry;
   bool m_busy=FALSE;


bool radio_busy = FALSE;
bool sensors_busy = FALSE;
adc_message_t pkt;
message_t package;


task void Send_soil_data();

event void Boot.booted()
{
call AMControl.start();
```

```
}


event void AMControl.startDone(error_t err) {

if(err == SUCCESS) {

//set the sample rate
call Timer0.startPeriodic(SAMP_INTRVAL);
call Leds.led2On();
call  ex_5_V.on();
}
else {
call AMControl.start();
}
}
event void ex_5_V.ExctDone(error_t err)
{}
event void AMControl.stopDone(error_t err) {
call Leds.led2On();
}

event void Timer0.fired() {
if(sensors_busy == FALSE) {
sensors_busy=TRUE;
call Hum.read();
}
}
event void Hum.readDone(error_t err, uint16_t val) {

if(err == SUCCESS)
{
//call Leds.led0On();
air_hum = val;
}
else
```

```
{
air_hum = 0xffff;
}
call Temp.read();
}
event void Temp.readDone(error_t err, uint16_t val) {

if(err == SUCCESS)
{
call Leds.led0On();
air_temp = val;

}
else
{
air_temp = 0xffff;
}
//post Send_soil_data();
call Sensor_0_Read.read();
}
event void Sensor_0_Read.readDone(error_t err, uint16_t val) {
if(err == SUCCESS) {
soil_mositure = val;
}
else
{
soil_mositure = 0xffff;
}
//post Send_soil_data();
call Sensor_1_Read.read();
}
event void Sensor_1_Read.readDone(error_t err, uint16_t val) {
if(err == SUCCESS) {
soil_temp = val;
}
else
```

```
{
soil_temp = 0xffff;
}
sensors_busy=FALSE;
post Send_soil_data();
}


task void Send_soil_data() {
counter=counter+SAMP_INTRVAL/10000;
sensors_busy=FALSE;

if(!sensors_busy&& !radio_busy) {

adc_message_t* soil_data = (adc_message_t*)
(call Packet.getPayload(&package, sizeof(adc_message_t)));
if(soil_data == NULL) {
return;
}
soil_data->counter = counter;
soil_data->nodeid = 0x0101;
soil_data->hum= air_hum;
soil_data->temp=air_temp;
soil_data->adcvalue0=soil_mositure;
soil_data->adcvalue1=soil_temp;
//set ack request
call PacketAcknowledgements.requestAck(&package);
if (call AMSend.send(1, &package, sizeof(adc_message_t)) == SUCCESS) {
radio_busy = TRUE;
}
//AM_BROADCAST_ADDR
sensors_busy = FALSE;
}


}
```

```
event void LogWrite.appendDone(void* buf, storage_len_t len,
                                bool recordsLost, error_t err) {
     m_busy = FALSE;
     call Leds.led2Off();
   }

event void AMSend.sendDone(message_t* bufPtr, error_t error) {
call Leds.led2Toggle();
if(call PacketAcknowledgements.wasAcked(bufPtr))
{
call Leds.led1Toggle();
}
else
{
//log event
if (!m_busy) {
      m_busy = TRUE;
m_entry.len = sizeof(bufPtr);
m_entry.msg = *bufPtr;
if (call LogWrite.append(&m_entry, sizeof(logentry_t)) != SUCCESS) {
m_busy = FALSE;}
}

}
if (&package == bufPtr) {
call Leds.led0Off();
radio_busy = FALSE;
}


//reactive algorithm
reduce_intrval= FALSE;
    set_timer_flag= FALSE;
if(soil_mositure> prev_val)
{
if(soil_mositure-prev_val >(2*GRANULARITY) )
```

```
{reduce_intrval=TRUE;}
if(soil_mositure-prev_val < GRANULARITY)
{num_simi_read=num_simi_read+1;}
}
else{
if(prev_val-soil_mositure>(2*GRANULARITY) )
{reduce_intrval=TRUE;}
if(prev_val-soil_mositure < GRANULARITY)
{num_simi_read=num_simi_read+1;}
}
if(reduce_intrval==TRUE)
{
if(SAMP_INTRVAL/8 >= MIN_INTRVAL )
{
SAMP_INTRVAL=SAMP_INTRVAL/8;
set_timer_flag=TRUE;
}
num_simi_read=0;
}
else if(num_simi_read >3)
{
if(SAMP_INTRVAL+ MIN_INTRVAL <=MAX_INTRVAL )
{
SAMP_INTRVAL=SAMP_INTRVAL+ MIN_INTRVAL;
set_timer_flag=TRUE;
}

num_simi_read=0;
}
else{}
if(set_timer_flag==TRUE)
{
call Timer0.startPeriodic(SAMP_INTRVAL);
}
//update previous
prev_val=soil_mositure;
```

```
}

event void LogRead.readDone(void* buf, storage_len_t len, error_t err) {

}

event void LogWrite.eraseDone(error_t err) {

   }

event void LogRead.seekDone(error_t err) {
}

event void LogWrite.syncDone(error_t err) {
}


}
```

Appendix C

BASE NODE LOGGING APP

```
import static java.lang.System.out;
import net.tinyos.message.*;
import net.tinyos.util.*;
import net.tinyos.packet.*;


import java.io.PrintWriter;
import java.io.*;
import java.text.DecimalFormat;
class Mda300Tester implements MessageListener{
private PhoenixSource phoenix;
private MoteIF mif;
private File f=null;
private PrintWriter logger;

public void writeToLog() throws FileNotFoundException
{

try
{    f = new File ("LOG_FILE.txt");
    logger = new PrintWriter(f);

}
catch (FileNotFoundException ex)
{
```

```
    // insert code to run when exception occurs
}



}

public Mda300Tester(final String source){
try {
writeToLog();

}
catch (FileNotFoundException ex)
{}
phoenix=BuildSource.makePhoenix(source, PrintStreamMessenger.err);
mif = new MoteIF(phoenix);
mif.registerListener(new SoilRead(),this);

}

public void messageReceived(int dest_addr,Message msg){
if(msg instanceof SoilRead){
SoilRead results = (SoilRead)msg;

double[] sensirionCalcData=null;
//out.println("The measurement will be printed here **//** ");
out.println();
out.println("Counter ID:"+results.get_counter());
out.println("Node ID :"+results.get_nodeid());

sensirionCalcData=calculateSensirion(results.get_temp()
,results.get_hum());

double sm_1=calculatedSoilMositure(results.get_adcvalue0());
double sm_2=calculatedSoilMositure(results.get_adcvalue1());
out.printf("Soil Moisture 1: %.3f\n",sm_1);
```

```
out.printf("soil Moisutre 2: %.3f\n",sm_2);
out.printf("Sensirion temperature: %.2f\n",sensirionCalcData[0]);
out.printf("Sensirion humidity: %.2f\n",sensirionCalcData[1]);
DecimalFormat df= new DecimalFormat("#.###");
//write to log
 logger.println(results.get_nodeid()+"\t"+results.get_counter() +
 "\t"+df.format(sm_1)+"\t"+df.format(sm_2)+"\t"
 +df.format(sensirionCalcData[0])+"\t"
 +df.format(sensirionCalcData[1]));
 logger.flush();
}

}

private double  calculatedSoilMositure(int sm){
double ret;
//out.printf("raw data %d\n",sm);
double vol= 2.5 * ((double)sm/4096);
if(vol< 1.1)
{
ret= vol*10-1;
}
else if (vol <1.3)
{
ret= vol*25-17.5;
}
else if (vol< 1.82)
{
ret= vol*48.08-47.5;
}
else
{
ret= vol*26.32-7.89;
}
return ret;
}
```

```
private double[] calculateSensirion(int Temperature,int Humidity){
double [] converted = new double[2];

converted[0]=-39.4+(0.01*(double)Temperature);
converted[1]=(-2.0468+0.0367*(double)Humidity
-0.0000015955*Math.pow((double)Humidity,(double )2))
+(converted[0]-25)*(0.01+0.00008*(double)Humidity);

return converted;
}



public static void main (String[] args) {
if ( args.length == 2 && args[0].equals("-comm") ) {

Mda300Tester hy = new Mda300Tester(args[1]);

} else {
System.err.println("usage: java Mda300Tester [-comm <source>]");
System.exit(1);
}

}

}
```

References

[1] "U.S. farms and farmers," Mar. 2015. [Online]. Available: http://www.agcensus.usda.gov/Publications/2012/Preliminary_Report/Highlights.pdf

[2] J. M. MacDonald, P. Korb, and R. A. Hoppe, "Farm size and the organization of U.S. crop farming," *Economic Research Report*, vol. ERR-152, August 2013. [Online]. Available: http://www.ers.usda.gov/media/1156726/err152.pdf

[3] "The global food crisis," Mar. 2015. [Online]. Available: http://ngm.nationalgeographic.com/2009/06/cheap-food/bourne-text/1

[4] "Scientists at MSU warn of global food crisis," Mar. 2015. [Online]. Available: http://www.bozemandailychronicle.com/news/education/scientists-at-msu-warn-of-global-food-crisis/article_365f69c1-df5c-52af-b084-cfe5a23f3f0b.html

[5] A. M. Adrian, S. H. Norwood, and P. L. Mask, "Producers' perceptions and attitudes toward precision agriculture technologies," *Computers and Electronics in Agriculture*, vol. 48, no. 3, pp. 256 – 271, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169905000852

[6] S. M. Pedersen, S. Fountas, B. S. Blackmore, M. Gylling, and J. L. Pedersen, "Adoption and perspectives of precision farming in Denmark," *Acta Agriculturae Scandinavica, Section B Soil & Plant Science*, vol. 54, no. 1, pp. 2–8, 2004. [Online]. Available: http://dx.doi.org/10.1080/09064710310019757

[7] E. Murakami, A. M. Saraiva, L. C. R. Junior, C. E. Cugnasca, A. R. Hirakawa, and P. L. Correa, "An infrastructure for the development of distributed service-oriented information systems for precision agriculture," *Computers and Electronics in Agriculture*, vol. 58, no. 1, pp. 37 – 48, 2007, precision Agriculture in Latin America. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169907000609

[8] J. W. Clay, *World Agriculture and the Environment: A Commodity-by-Commodity Guide to Impacts and Practices*. Island Press, 2004. [Online]. Available: http://islandpress.org/world-agriculture-and-environment

[9] "Support vector machine," Mar. 2015. [Online]. Available: http://en.wikipedia.org/wiki/Support_vector_machine

[10] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Sep. 2001. [Online]. Available: http://dx.doi.org/10.1162/15324430152748236

[11] B. Majone, F. Viani, E. Filippi, A. Bellin, A. Massa, G. Toller, F. Robol, and M. Salucci, "Wireless sensor network deployment for monitoring soil moisture dynamics at the field scale," *Procedia Environmental Sciences*, vol. 19, pp. 426 – 435, 2013, Four Decades of Progress in Monitoring and Modeling of Processes in the Soil-Plant-Atmosphere System: Applications and Challenges. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1878029613003198

[12] R. Cardell-Oliver, K. Smettem, M. Kranz, and K. Mayer, "Field testing a wireless sensor network for reactive environmental monitoring [soil moisture measurement]," in *Intelligent Sensors, Sensor Networks and Information Processing Conference. Proceedings of the 2004*, Dec 2004, pp. 7–12.

[13] X. Qiao, F. Yang, and X. Xu, "The prediction method of soil moisture content based on multiple regression and RBF neural network," in *Ground Penetrating Radar (GPR), 2014 15th International Conference on*, June 2014, pp. 140–143.

[14] B. Zaman and M. McKee, "Spatio-temporal prediction of root zone soil moisture using multivariate relevance vector machines," *Open Journal of Modern Hydrology*, vol. 4, pp. 80–90, 2014.

[15] M. Kashif Gill, M. W. Kemblowski, and M. McKee, "Soil moisture data assimilation using support vector machines and ensemble Kalman filter," *Journal of the American Water Resources Association*, vol. 43, no. 4, pp. 1004–1015, 2007. [Online]. Available: http://dx.doi.org/10.1111/j.1752-1688.2007.00082.x

[16] S. Gorthi, "Prediction models for estimation of soil moisture content," M.S. thesis, Utah State University, Logan, Utah, 2011.

[17] "Illinois climate network," Mar. 2015. [Online]. Available: http://www.sws.uiuc.edu/warm/soil/

[18] H. Finch, A. Samuel, and G. Lane, "10 - precision farming," in *Lockhart & Wisemans Crop Husbandry Including Grassland*, 9th ed., H. Finch, A. Samuel, and G. Lane, Eds. Woodhead Publishing, 2014, pp. 235 – 244. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9781782423713500102

[19] N. Zhang, M. Wang, and N. Wang, "Precision agriculture–a worldwide overview," *Computers and Electronics in Agriculture*, vol. 36, no. 23, pp. 113 – 132, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169902000960

[20] J. F. Power, P. L. Brown, T. J. Army, and M. G. Klages, "Phosphorus responses by dryland spring wheat as influenced by moisture supplies," *Agron. J*, vol. 53, pp. 106–108, 1961. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169902000960

[21] C. Stewart, A. McBratney, and J. Skerritt, "Site-specific durum wheat quality and its relationship to soil properties in a single field in northern new south wales," *Precision Agriculture*, vol. 3, no. 2, pp. 155–168, 2002. [Online]. Available: http://dx.doi.org/10.1023/A%3A1013871519665

[22] S. Machado, E. J. Bynum, T. Archer, R. Lascano, L. Wilson, J. Bordovsky, E. Segarra, K. Bronson, D. Nesmith, and W. Xu, "Spatial and temporal variability of corn grain yield: Site-specific relationships of biotic and abiotic factors," *Precision Agriculture*, vol. 2, no. 4, pp. 359–376, 2000. [Online]. Available: http://dx.doi.org/10.1023/A%3A1012352032031

[23] "Irrigation Management for Corn," May 2008. [Online]. Available: http://ianrpubs.unl.edu/live/g1850/build/

[24] "Maximizing root zone utilization: A key to precision irrigation management," 2014. [Online]. Available: http://cropmetrics.com/2014/04/maximizing-root-zone-utilization-a-key-to-precision-irrigation-management/

[25] "Measurement of soil moisture," 2014. [Online]. Available: http://www.ext.colostate.edu/drought/soilmoist.html

[26] J. F. Arya, Lalit M.and Paris, "A physicoempirical model to predict the soil moisture characteristic from particle-size distribution and bulk density data," *Soil Sci. Soc. Am. J*, vol. 45, p. 10231030, 1981. [Online]. Available: https://dl.sciencesocieties.org/publications/citation-manager/sssaj/45/6/SS0450061023

[27] X. Liang, E. F. Wood, and D. P. Lettenmaier, "Surface soil moisture parameterization of the vic-2l model: Evaluation and modification," *Global and Planetary Change*, vol. 13, no. 14, pp. 195–206, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0921818195000461

[28] J. Huang, H. M. van den Dool, and K. P. Georgarakos, "Analysis of model-calculated soil moisture over the United States (1931-1993) and applications to long-range temperature forecasts," *J. Climate*, vol. 9, pp. 1350–1362, 1996.

[29] K. Y. Vinnikov and I. B. Yeserkepova, "Soil moisture: Empirical data and model results," *J. Climate*, vol. 4, p. 6679, 1991.

[30] E. G. Njoku and D. Entekhabi, "Passive microwave remote sensing of soil moisture," *Journal of Hydrology*, vol. 184, no. 12, pp. 101 – 129, 1996, Soil Moisture Theories and Observations. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0022169495029702

[31] V. Lakshmi, "Remote sensing of soil moisture," *ISRN Soil Science*, vol. 4, p. 33, 2013.

[32] D. Entekhabi, E. Njoku, P. O'Neill, K. Kellogg, W. Crow, W. Edelstein, J. Entin, S. Goodman, T. Jackson, J. Johnson, J. Kimball, J. Piepmeier, R. Koster, N. Martin, K. McDonald, M. Moghaddam, S. Moran, R. Reichle, J.-C. Shi, M. Spencer, S. Thurman, L. Tsang, and J. Van Zyl, "The soil moisture active passive (SMAP) mission," *Proceedings of the IEEE*, vol. 98, no. 5, pp. 704–716, May 2010.

[33] U. Narayan and V. Lakshmi, "Characterizing subpixel variability of low resolution radiometer derived soil moisture using high resolution radar data," *Water Resources Research*, vol. 44, no. 6, 2008. [Online]. Available: http://dx.doi.org/10.1029/2006WR005817

[34] U. Narayan, V. Lakshmi, and T. Jackson, "High-resolution change estimation of soil moisture using l-band radiometer and radar observations made during the smex02 experiments," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 44, no. 6, pp. 1545–1554, June 2006.

[35] T. J. Jackson, J.Schmugge, and E. T. Egnman, "Remote sensing applications to hydrology: soil moisture," *Hydrological Sciences Journal*, vol. 41, no. 4, pp. 517–530, 1996. [Online]. Available: http://dx.doi.org/10.1080/02626669609491523

[36] M. Matin and M. M. Islam, "Wireless sensor networks - technology and protocols," 2012. [Online]. Available: http://www.intechopen.com/books/export/citation/BibTex/wireless-sensor-networks-technology-and-protocols/overview-of-wireless-sensor-network

[37] "Machine learning," Mar. 2015. [Online]. Available: http://en.wikipedia.org/wiki/Machine_learning

[38] P. Patil, H. Vidya, S. Patil, and U. Kulkarni, "Wireless sensor network for precision agriculture," in *Computational Intelligence and Communication Networks (CICN), 2011 International Conference on*, Oct. 2011, pp. 763–766.

[39] R. Beckwith, D. Teibel, and P. Bowen, "Report from the field: results from an agricultural wireless sensor network," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, Nov. 2004, pp. 471–478.

[40] I. F. Akyildiz and E. P. Stuntebeck, "Wireless underground sensor networks: Research challenges," *Ad Hoc Networks*, vol. 4, no. 6, pp. 669 – 686, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870506000230

[41] H. Liu, Z. Meng, and S. Cui, "A wireless sensor network prototype for environmental monitoring in greenhouses," in *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, Sept 2007, pp. 2344–2347.

[42] C. Wang, C. Zhao, X. Qiao, X. Zhang, and Y. Zhang, "The design of wireless sensor networks node for measuring the greenhouse's environment parameters," in *Computer And Computing Technologies In Agriculture, Volume II*, ser. The International Federation for Information Processing, D. Li, Ed. Springer US, 2008, vol. 259, pp. 1037–1046. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-77253-0_36

[43] H. Murase, N. Honami, and Y. Nishiura, "A neural network estimation technique for plant water status using the textural features of pictorial data of plant canopy," *Acta Hort*, no. 255-262, 1995. [Online]. Available: http://www.actahort.org/books/399/399_30.html

[44] S. Dimitriadis and C. Goumopoulos, "Applying machine learning to extract new knowledge in precision agriculture applications," in *Informatics, 2008. PCI '08. Panhellenic Conference on*, Aug 2008, pp. 100–104.

[45] "Wikipedia. Weka," Mar. 2015. [Online]. Available: http://en.wikipedia.org/wiki/Weka_%28machine_learning%29

[46] H. Auernhammer, M. Demmel, F. Maidl, U. Schmidhalter, T. Schneider, and P. Wagner, "An on-farm communication system for precision farming with nitrogen real-time application," *ASAE*, no. 991150, 1999.

[47] C. Goumopoulos, A. Kameas, and B. OFlynn, "Proactive agriculture: An integrated framework for developing distributed hybrid systems," in *Ubiquitous Intelligence and Computing*, ser. Lecture Notes in Computer Science, J. Indulska, J. Ma, L. Yang, T. Ungerer, and J. Cao, Eds. Springer Berlin Heidelberg, 2007, vol. 4611, pp. 214–224. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-73549-6_22

[48] N. Stojanovic, Y. Xu, B. S. Nissatech, and L. Stojanovic, "Mobile CEP architecture: From intelligent sensing to collaborative monitoring," in *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, ser. DEBS '14. New York, NY, USA: ACM, 2014. [Online]. Available: http://doi.acm.org/10.1145/2611286.2611322 pp. 350–353.

[49] "MicaZ - memsic," Mar. 2015. [Online]. Available: http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf

[50] "Tinyos," Mar. 2015. [Online]. Available: http://www.tinyos.net/

[51] "Soil climate analysis network," Mar. 2015. [Online]. Available: http://www.wcc.nrcs.usda.gov/scan/

[52] J. Paek, K. Chintalapudi, R. Govindan, J. Caffrey, and S. Masri, "A wireless sensor network for structural health monitoring: Performance and experience," in *Embedded Networked Sensors, 2005. EmNetS-II. The Second IEEE Workshop on*, May 2005, pp. 1–10.

[53] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, "Deploying a wireless sensor network on an active volcano," *Internet Computing, IEEE*, vol. 10, no. 2, pp. 18–25, March 2006.

[54] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, ser. WSNA '02. ACM, 2002. [Online]. Available: http://doi.acm.org/10.1145/570738.570751 pp. 88–97.

[55] H. Bogena, J. Huisman, C. Oberdrster, and H. Vereecken, "Evaluation of a low-cost soil water content sensor for wireless network applications," *Journal of Hydrology*, vol. 344, no. 12, pp. 32 – 42, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022169407003514

[56] M. Kramer and A. Geraldy, "Energy Measurements for MicaZ Node," University of Kaiserslautern, Germany, techinical report KrGe06, 2006.

[57] "Mts/mda sensor board users manual," Mar. 2015. [Online]. Available: http://eps2009.dj-inod.com/docs/09-05-16/MTS-MDA_Series_Users_Manual.pdf

[58] "Memsic, Inc - gateways mib520," Mar. 2015. [Online]. Available: http://www.memsic.com/wireless-sensor-networks/MIB520

[59] "Ultra-low-power msp430 microcontroller launchpad kits," Mar. 2015. [Online]. Available: http://www.ti.com/ww/en/launchpad/launchpads-msp430.html

[60] "Waspmote," Mar. 2015. [Online]. Available: http://www.libelium.com/products/waspmote/

[61] "Vh400 soil moisture sensor probes," Mar. 2015. [Online]. Available: http://vegetronix.com/Products/VH400/

[62] "Vegetronix soil temperature sensor probes," Mar. 2015. [Online]. Available: http://www.vegetronix.com/Products/THERM200/

[63] A. Rao and K. Saxton, "Analysis of soil water and water stress for pearl millet in an indian arid region using the SPAW model," *Journal of Arid Environments*, vol. 29, no. 2, pp. 155 – 167, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140196305800862

[64] V. K. Arora, C. Singh, and K. Singh, "Comparative assessment of soil water balance under wheat in a subtropical environment with simplified models," *The Journal of Agricultural Science*, vol. 128, pp. 461–468, 6 1997. [Online]. Available: http://journals.cambridge.org/article_S0021859697004358

[65] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. [Online]. Available: http://dx.doi.org/10.1007/BF00994018

[66] "Libsvm a library for support vector machines," Mar. 2015. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[67] "Sparsebayes software," Mar. 2015. [Online]. Available: http://www.miketipping.com/downloads.html