Hindawi Publishing Corporation
Journal of Sensors
Volume 2016, Article ID 4178079, 11 pages
http://dx.doi.org/10.1155/2016/4178079

Research Article

Raspberry Pi Based Intelligent Wireless Sensor Node for Localized Torrential Rain Monitoring

Zhaozhuo Xu,[1] Fangling Pu,[1,2] Xin Fang,[1] and Jing Fu[1]

[1] *Wireless Communication and Sensor Network Laboratory, School of Electronic Information, Wuhan University, Wuhan 430072, China*
[2] *Collaborative Innovation Center of Geospatial Technology, Wuhan University, Wuhan 430072, China*

Correspondence should be addressed to Fangling Pu; flpu@whu.edu.cn

Received 5 July 2016; Revised 16 September 2016; Accepted 17 October 2016

Academic Editor: Salvatore Sorce

Wireless sensor networks are proved to be effective in long-time localized torrential rain monitoring. However, the existing widely used architecture of wireless sensor networks for rain monitoring relies on network transportation and back-end calculation, which causes delay in response to heavy rain in localized areas. Our work improves the architecture by applying logistic regression and support vector machine classification to an intelligent wireless sensor node which is created by Raspberry Pi. The sensor nodes in front-end not only obtain data from sensors, but also can analyze the probabilities of upcoming heavy rain independently and give early warnings to local clients in time. When the sensor nodes send the probability to back-end server, the burdens of network transport are released. We demonstrate by simulation results that our sensor system architecture has potentiality to increase the local response to heavy rain. The monitoring capacity is also raised.

1. Introduction

The development of Internet of Things (IoT) signals a shift in the resources of data. An increasing proportion of data collected today is generated by sensors. From this point of view, the public's urge for accurate environmental information may be sated by large-scale wireless sensor networks (WSN) [1] based on advanced Information Communication Technologies (ICT) infrastructures. Currently, contextualized and location-aware environmental sensor networks (ESN) [2] are the mainstream in this area. ESN is promising mainly due to the inexpensive embedded and system-on-chip hardware, convenient access to communication networks, and decreased cost for data storage.

In most existing ESN systems, the flood of sensor-generated data pours into the back-end server without processing, which shoves heavy load onto network transmission. In harsh environments such as torrential rain and typhoon, short time network failure leads to serious paralysis in monitoring system. The database server may receive missing or corrupted data for long period of time. In that case, ESN loses the capabilities of recording and forecasting environmental changes, which are the original intentions of ESN. Therefore, it is desirable to learn the environmental information in the front-end, so that the ESN system can respond to different environmental situations with less help from the back-end. This new data processing system, which we call *intelligent wireless sensor node*, is the core of future ESN systems in front-end. With increasing intelligence in the embedded system that collects and learns from the sensor data, ESN's performance in predicting and capturing sudden extreme environmental changes will be improved. Also, the life-long training enables sensor node to reach better accuracy.

Case Study: Localized Torrential Rain (LTR) Monitoring. Torrential rain is a phenomenon that can significantly affect residents in the immediate surrounding of localized areas. Torrential rain can cause flooding and road closures, which significantly affects day-to-day travels in the flooded areas. Hence, early warning is desired for adjusting travel plans in advance. Localized torrential rain (LTR) is usually preceded by a series of changes in weather phenomena. Therefore, it is possible to predict LTR using the past and current meteorological information. When early warnings are

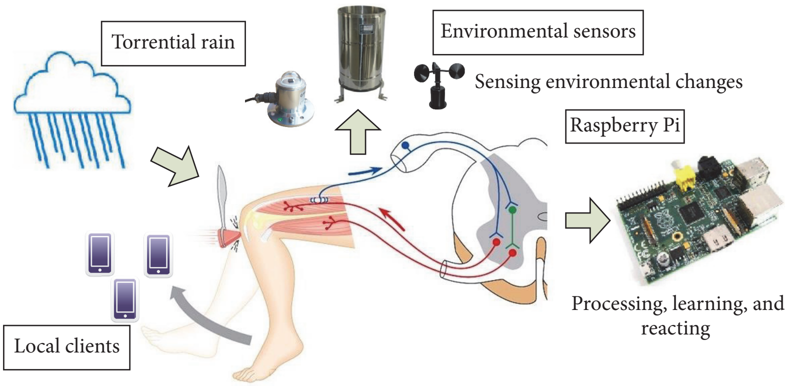

Torrential rain
Environmental sensors
Sensing environmental changes
Raspberry Pi
Processing, learning, and reacting
Local clients

Figure 1: Biology inspired intelligent wireless sensor architecture.

published [3, 4], local residents are able to make smart travel choices.

LTR has been found especially challenging in environmental monitoring. In principle, if the movement of cumulus clouds is captured by satellite or radar, then the precipitation in a certain area can be estimated by analyzing the acquired images. However, this approach is inapplicable because current spatial and temporal resolutions of radar or satellite image are far below the request. As ESN has gradually replaced remote sensing approaches in applications such as air [5] and water [6] quality monitoring, it may outperform existing approaches for localized torrential rain monitoring [7–9]. In fact, ESN has become the substitute of surface climate station for its high resolution and low cost. If we compare a city's ICT infrastructure to a person [10], there is no doubt that wireless sensor networks are the hidden neurons inside the body. Weather information obtained by sensors is usually transmitted from distributed wireless sensor nodes to back-end server, the clouds, where correlation analysis is conducted among all the received data [11]. Some positive results have been achieved by ESN in collecting localized meteorological information. However, ESN lacks an intelligent solution for monitoring LTR in a timely, efficient, and automatic manner.

Raspberry Pi Based Intelligent Wireless Sensor Node for LTR Monitoring. To offer an intelligent solution for timely, efficient, and automatic LTR monitoring, we introduce Raspberry Pi to be the wireless sensor node. Raspberry Pi is a well-known type of single-board computers (SBC), which has quickly occupied the embedded system market for its comprehensive abilities and low cost. Moreover, it gradually becomes the main products of environmental sensor information systems. However, it seems to be a waste of resources if we use SBC only for collecting and packaging sensor data; advanced data analysis can be expected to be done within SBC.

Inspired by the knee-jerk reflex, a well-known biological phenomenon, we developed an intelligent wireless sensor node for distributed heavy rain monitoring. The main idea is demonstrated in Figure 1. Multiple sensors stand out and sensory nerve, information such as temperature, humidity, solar radiation, and rainfall, is collected and transmitted to Raspberry Pi (the sensor neuron and spinal cord). Statistical and machine learning methods are implemented on Raspberry Pi to evaluate the connection between rainfall and environmental conditions.

Then, the probability of upcoming localized torrential rain is estimated and sent to back-end database and server. Also, clients in local area network will receive early warning of heavy rain.

Our major contributions in this paper are as follows:

(i) Environmental information acquisition with high spatial and temporal resolution: the area of study is smaller than the minimum observation area that satellite can tell.

(ii) Techniques for reducing ESN's failure time: our system works in times when harsh environments influence the communication quality and paralyze current ESN. Precious weather conditions that other might miss are observed by our new sensor node.

(iii) Life-long learning and timely reaction with full use of prior knowledge: our system learns the labeled meteorological data using powerful unsupervised learning algorithms such as support vector machines (SVM) and predicts the probability of upcoming LTR with accuracy.

The reminder of this paper proceeds as follows. Section 2 presents the related works, which include WSN for localized heavy rain detection, Raspberry Pi for environmental monitoring, and probability estimation model for meteorological applications. Section 3 describes Raspberry Pi based intelligent torrential rain sensing system. The system architecture, multiple low-cost sensors, local computing kernel based on Raspberry Pi, and information processing software are introduced in this section. Section 4 provides the state-of-the-art information theoretical outlier detection method we utilized in our system. Section 5 presents probability estimation models for heavy rain, which involves data normalization, localized torrential rain correlation model, logistics regression, and multikernel support vector machines. Section 6 discusses the experiments and the results. Conclusion is given in Section 7.

2. Related Works

2.1. Architecture: WSN for Localized Heavy Rain Detection. Surface Meteorological Monitoring Networks (SMMNs) were first introduced to detect LHR. Researchers developed SMMN system to measure the rainfall and wind speed in numerous places. Calibration scheme and early predictions are made to LHR when SMMNs are installed in environments. On the other hand, studies in radio signals [12] show that attenuation of Wi-Fi or microwave signals may have connections with heavy rain in nearby environment. Models like compressive sensing [13] are used in finding the signals related to heavy rain.

Admittedly, SMMN fills the gaps between the minimum pixels of radar and satellite image, which enables us to identify rainfall in limited areas. However, the foundation of SMMN is stable and fast internet is required for its operation. All sensor information must be gathered and processed in background, which leaves uncertainty to the normal operation of sensors. Once the internet is disturbed by weather, sensor information may be lost.

2.2. Device: Raspberry Pi for Environmental Monitoring. Since Raspberry Pi's appearance in the markets, plenty of attention is paid to its use in environmental issues. At first, Raspberry Pi is considered to be one of the alternative plans towards wireless sensor node in system design areas [14]. Thanks to the assistance from open-source platforms and software, sensor system can be constructed by Raspberry Pi and Arduino. It is widely acknowledged that this combination is low-cost and available to all sorts of environmental sensors.

Also, Raspberry Pi plays an essential role in information processing on environmental relevant data. By extracting effective information from rainfall warning calls to authorities and weather posts online [15], Raspberry Pi provides useful information for background analysis. Experiments have shown that this method made full use of information ignored by most meteorologists and brought novel ideas to localized rain warning.

Nevertheless, existed applications of Raspberry Pi do not make full use of its competence in high performance computing. Especially when MATLAB offers its open source in Raspberry Pi, scientific computing algorithms can perfectly operate in embedded platforms. There is no doubt that complex algorithms' realization in open-source systems will receive more and more attention in the future.

2.3. Processing: Outlier Detection on Sensor Data. Due to the flood of machine-generated data received in database systems, hidden outliers and anomalous value must be effectively detected in advanced database systems. According to [16], outlier detection methods have three major categories: supervised, unsupervised, and semisupervised approaches. The difference among these three categories lies in the availability of labels in training datasets [17]. For example, supervised anomaly detection approach is close to supervised classification, which requires training data labeled as normal or abnormal. Popular supervised models like support vector machines (SVM) [18, 19] consider training data as a point in a multidimensional space. Then, they select a half-space that contains most of the points prelabeled as normal. Any test data that falls outside the area is determined as outlier. Sometimes we just want either to model normality or under few circumstances to model abnormality [20, 21]. In that case, only normal class is taught but the abnormality can be recognized. These approaches are named as semisupervised outlier detection, which learns a model from a given normal dataset and calculates the likelihood of test objects. Both supervised and semisupervised detection methods are widely applicable when provided with large volume of training objects. However, the general current situation that at present data scientists face is the shortage of historical data. Advanced algorithms, which are named as unsupervised approaches, are required to determine the abnormality without prior knowledge. Unsupervised outlier detection becomes the constant challenges in recent years [22]. Existing researches focus more on refining clustering algorithms such as K-neighbors [23], K-means [24], and K-methods [25] in order to meet the requirements for unsupervised detection. But most of those approaches are limited to numerical values. Only few outlier detection techniques are designed to process categorical information such as name, gender, and address. Also, owing to high dimensionality of datasets, complex statistical tests, and unnecessary approximation, the efficiency of these algorithms will suffer. Thus, new, general, efficient, and unsupervised outlier detection algorithms are required for big data analysis. Previous improvements have been made via introducing information theory concepts such as entropy, mutual information, and conditional mutual information to denote the outliers. These techniques expand the detection objects to both numerical and categorical data [26, 27]. Because the existence of outliers increases the overall entropy of certain data attributes, outlier can be modeled as constraints that impede datasets to reach their optimal entropy. Information theoretic techniques work well in quick outliers determination.

2.4. Learning: Rainfall Probability Estimation. Several probability estimation methods have been proposed in the literature. Rainfall probability is the most critical part of meteorological prediction. Probability and frequency analysis of rainfall data derive the expected rainfall occurrence and thus help in better understanding of spatial-temporal rainfall pattern [28]. Current rainfall prediction methods can be grouped into two categories: distribution fitting and classification approaches. The distribution fitting approaches select the best fit distribution models for annual, seasonal, and monthly rainfall time series based on values of statistical test [29].

However, rainfall probability is so changeable that it cannot be covered by certain distribution. Instead they are the result of combined factors. Therefore, logistic regression and SVM algorithms become major stream in rainfall probability estimation. Logistic regression was chosen and implemented on GIS system for quantitative prediction of rainfall and landslide in the study area [30]. SVM have been proved to impact positively the prediction of the Indian summer monsoon rainfall [31]. Also, hybrid models combining random forest

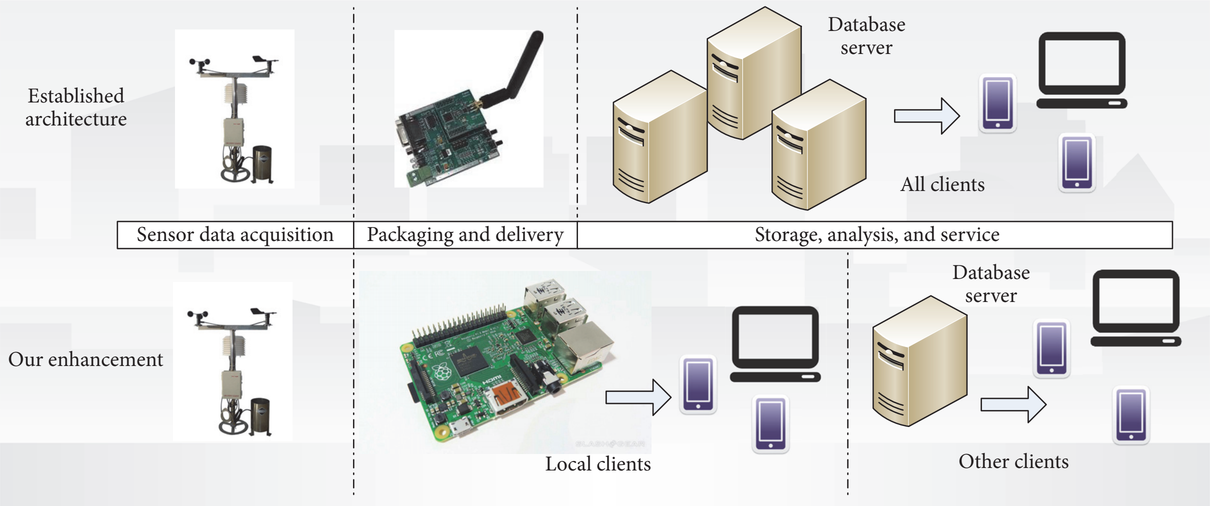


Figure 2: Enhancement in WSN architecture for localized torrential rain monitoring.

(RF) and SVM have been used to predict amount of rainfall in rainfall occurring days [32]. Accuracy achieved by SVM in this application shows its potential in rainfall probability estimation. Researchers may get better performance using SVM, especially with large-scale historical sensor dataset.

3. Raspberry Pi Based Localized Torrential Rain Sensing System

3.1. System Architecture. As Figure 2 implies, our localized torrential rain WSN differs from existing ESN in the roles that wireless sensor nodes play. Our sensor nodes transform the centralized storage and computing to distributed alternatives, which share part of database's tasks. With swap space created under Raspbian (the Raspberry Pi Linux operating systems), storage spaces like SD cards and RAMs in MCUs are managed by unified virtual memory. Thus, sensor information can be saved orderly in Raspberry Pi's storage units. Additionally, Samba software is deployed on Raspberry Pi, which generates sensor files for windows back-end server using Common Internet File System (CIFS)/Server Message Block (SMB) networking protocol. Raspberry Pi supersedes database server in quick sensor data storage and local warning to registered clients. After running algorithms for localized heavy rain estimation, sensor nodes publish heavy rain probabilities directly to the local clients who registered in back-end previously. Client lists will be refreshed periodically according to the interaction between the back-end database and the sensor nodes.

We now provide details of the three main parts of our ESN system, which are multiple low-cost sensors, local computing platform based on Raspberry Pi, and information processing software. Both fundamental hardware and local service software are introduced. All the energy our system needed is supported by green solar power or electrical power.

3.2. Multiple Low-Cost Sensors. In order to capture the rapid climate changes, there is no doubt that an integrated sensor system must contain comprehensive sensors so that we can know the exact state in a localized area. The sensors need to be able to measure temperature, humidity, solar radiation, air pressure, wind speed, and rain gauge. Moreover, the sensors should be designed for long-time utilization and accurate in an acceptable range. As Table 1 shows, we choose a series of low-cost [33] and low-power meteorological sensors. Also our sensors share the same error ratio, so that influence brought by differences in error ranges is minimized.

Table 1: Meteorological sensors produced by Fuyuan Technology Feike Electronic Company in Wuhan, China.

Sensor type	Sensed physical quantity	Range (min–max)	Cost $
FY-P	Air pressure	500–1100 hPa	33
FY-T1	Temperature	50–70°C	10
FY-RH	Humidity	0–100%	16
FY-FS	Wind speed	0–100 m/s	33
FY-ZF	Solar radiation	0–2000 W/m^2	25

3.3. Local Computing Platforms Based on Raspberry Pi. Raspberry Pi board is one of the prevailing single-board computers specially designed for open-source development. We selected Raspberry 2 model B to construct sensor node. Raspberry 2 model B typically consumes power less than 2 W when conducting scientific computing. With a Broadcom BCM2836 system-on-chip (SoC), which consists of a 900 MHz quad-core ARM Cortex-A7 CPU and 1 GB RAM, Raspberry Pi is capable of complex operation. What is more, Raspberry Pi has 4 USB ports, 40 GPIO pins, full HDMI port, ethernet port, display interface (DSI), micro SD card slot, and 3D graphics core, which provide more possibilities for its application. Recently, MATLAB launched its package and open-source toolboxes on Raspbian. Therefore, MATLAB

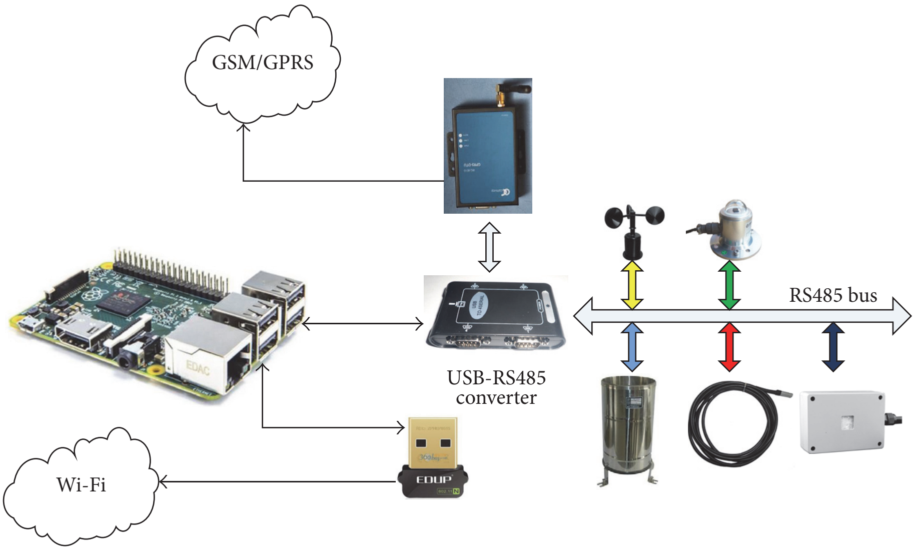


Figure 3: Integrated sensor system based on Raspberry Pi.

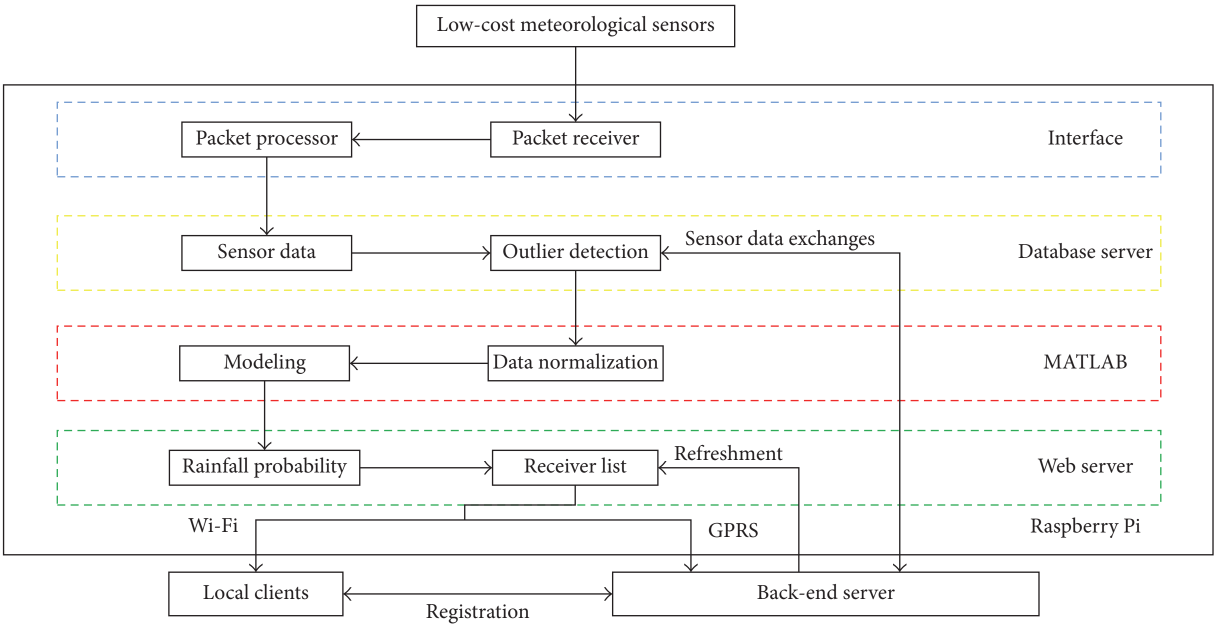


Figure 4: Software architecture.

based intelligent algorithms can be implemented on Raspberry Pi, which bring the idea of our paper.

As Figure 3 shows, all sensors are connected to a RS485 bus. MODBUS protocol is applied on Raspberry Pi to start sensor and get data through a USB-RS485 converter. Raspberry Pi stores the received data and then conducts correlation analysis aiming at offering probabilities of upcoming heavy rain. After calculation, Raspberry Pi sends results to local registered clients through Wi-Fi (wireless fidelity) communication module and database server will receive data package through GPRS (General Packet Radio Service) module.

3.4. Information Processing Software. The layers of our software are shown in Figure 4. From top to down, the obtained sensor data are managed by the data management module. It stores the sensor data in MySQL database on board and controls the sensor data exchanges between Raspberry Pi and the back-end server. The sensor data are interpreted by MATLAB database toolbox that is installed on Raspberry Pi [34]. The data processing algorithm is the core of the whole software. Both logistic regression and support vector machine classifiers are available for MATLAB programs to train and test, which will be discussed in later part. Our programs compare the test results of several popular algorithms in period and select the best fitting model according to the accuracy in LTR prediction and efficiency. All following localized torrential rain estimation will be based on the model we chose. With the receiver list given by database server, Raspberry Pi sends probabilities of localized heavy rain to nearby clients through local Wi-Fi. The receiver list will be updated by back-end periodically according to clients' registration on our website. Also back-end server database can get sensor data stored on Raspberry Pi through its interaction with Raspberry Pi's data management system. Once the command is sent by back-end server, sensor data in certain period will be packed and transmitted from Raspberry Pi to back-end server through GPRS/protocols. [35, 36].

4. Default Outlier Detection

In this section we present the outlier detection method that filters the original sensor data for following probability estimation on rainfall probability. We introduce and refine information theoretical outlier function developed by Wu and Wang [26] and apply the new function in anomaly identification on environmental sensor data.

4.1. Entropy and Mutual Information. Consider a dataset D containing k objects $\{x_1, x_2, \ldots, x_k\}$, each object x_i $(1 \leq i \leq k)$ being m attributes categorical vector. To make it clear, we use matrix $\{x_{i,1}, x_{i,2}, \ldots, x_{i,m}\}$ to represent x_i. And $\{A_1, A_2, \ldots, A_m\}$ is used to represent D as a form of attributes. The entropy of a certain attribute A_j $(1 \leq j \leq m)$ is defined as

$$H\left(A_j\right) = -\sum_{v\in \mathrm{domain}(A_j)} P\left(A_j = v\right) \log P\left(A_j = v\right), \quad (1)$$

where the set of possible values in A_j is called its domain and written in domain (A_j), which is $\{v_1, v_2, \ldots\}$.

The mutual information is developed as follows to evaluate the correlation of heterogeneous information:

$$I\left(A_i, A_j\right) = H\left(A_i\right) + H\left(A_j\right) - H\left(A_i, A_j\right), \quad 1 \leq i, j \leq m \ (i \neq j), \quad (2)$$

where $H(A_i, A_j)$ is the joint entropy of A_i, A_j which uses joint probability of A_i, A_j to calculate entropy.

4.2. Information Theoretical Outlier Detection. Frequency based algorithms cannot handle the massive informative datasets because the value of those attributes has too much variance. Therefore, deep research on the essence of outlier was conducted previously by Wu and Wang [26]. In that paper, they derive the function of outliers using the differential holoentropy which takes both entropy and total correlation of data attributes into account. However, their work mainly focuses on detecting abnormal objects other than values, so they sum up data from different attributes and add weights to each value to make a combination factor for an object. After comparing outlier factors with threshold, outlier objects can be detected. This method shuts down the future opportunity for us to derive connections among outliers from various attributes. Therefore, in this section, we refine his model and focus on outlier detection within one attribute.

Definition 1 (the refined approximate differential holoentropy). Generally, for a dataset $Y = \{y_1, y_2, \ldots, y_m\}^T$ with m attributes and n objects $\{x_1, x_2, \ldots, x_n\}$, where $x_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$, the approximate differential holoentropy of x_0 is written as

$$\widehat{h}\left(x_0\right) = \sum_{i=1}^{m} W_X\left(y_i\right)\left(\log a - \frac{a}{b}\log b\right) - aW_X\left(Y\right) + a\sum_{i=1}^{m} \mathrm{OF}\left(x_{o,i}\right), \quad (3)$$

where

$$\begin{gathered} W_X\left(y_i\right) = 2\left(1 - \frac{1}{1 + \exp\left(-H_x\left(y_i\right)\right)}\right) \\ W_X\left(Y\right) = \sum_{i=1}^{m} W_X\left(y_i\right) H_x\left(y_i\right) \\ \mathrm{OF}\left(x_{o,i}\right) = \begin{cases} 0, & \text{if } n\left(x_{o,i}\right) = 1 \\ \delta\left[n\left(x_{o,i}\right)\right], & \text{otherwise} \end{cases} \\ \delta\left(x\right) = \left(x - 1\right)\log\left(x - 1\right) - x\log\left(x\right). \end{gathered} \quad (4)$$

$H_x(Y)$ represents the joint entropy of attributes in Y. b and a are reciprocal values of the cardinality of original attributes and attributes without $x_{o,i}$, and $n(x_{o,i})$ denotes the times $x_{o,i}$ appears in the ith attribute.

In our work, environmental data is fluctuated and highly correlated. The weights of different attributes are almost the same. Moreover, we want to preserve the normal value in certain objects instead of deleting the whole objects. So we refine the function by giving uniform weights and then focus on certain attributes rather than objects. The refined function is as follows:

$$\widehat{h_r}\left(x_0\right) = \sum_{i=1}^{m}\left(\log a - \frac{a}{b}\log b\right) - aH_x\left(Y\right) + a\sum_{i=1}^{m} \mathrm{OF}\left(x_{o,i}\right). \quad (5)$$

Now we have derived the function of outliers in one attribute. In the next step, this function is applied to an outlier detection algorithm as in Algorithm 1.

5. Probability Estimation Model for LTR

In this part we discuss localized torrential rain estimation models utilized in our information processing software. After methods of default outlier detection, the meteorological data attributes must be normalized so that they can be processed by logistic regression and SVM model. Because meteorological data attributes are correlated according to past research [3], SVM kernels that model the data vectors by their lengths might reach a high accuracy level. Therefore, we introduce two kernels in order to get better performance.

```
**Input**: Informative and correlated subsets $L = \{A_1, A_2, \dots, A_l\}$ of $D$
**Output**: Label matrix label of $L$
$L \leftarrow \emptyset$
Compute and store the Refined Approximate Differential Holoentropy $\widehat{h_r}(x_0)$ and $\mathrm{OF}(x_{o,i})$ for all $x_{o,i}$
**For** all $x_{o,i} \in D$
**If** $\widehat{h_r}(x_0) > 0$ or $\mathrm{OF}(x_{o,i})$ is the minimum $\mathrm{label}_{i,j} = 1$
**Else** $\mathrm{label}_{i,j} = 0$
**Return** label
```

Algorithm 1: Outlier detection on informative dataset.

5.1. Data Normalization. Because of the disunity of dimension, all the collected data must be standardized before it can be used for analysis. There are two major methods of normalization below:

$$D_s = \frac{D - D_{\min}}{D_{\max} - D_{\min}}$$
$$D_s = \frac{D - D_{\mathrm{mean}}}{\sigma_d}. \tag{6}$$

Here D_s represents the results of standardization. $D_{\min}$, $D_{\max}$, D_{mean}, and σ_d are the minimum, maximum, average, and variance of all received data, respectively. Because the variance calculation of data requires more computing time and the mean of rainfall data is less valuable in our discussion, we choose the first method as our standardization process.

5.2. Estimated LTR Probability

Definition 2 (estimated heavy rain probability). Generally, for a dataset $S = [S_t, S_h, S_p, S_w, S_r]$ with attributes as normalized temperature, humidity, air pressure, wind speed, and solar radiation, the estimated probability of heavy rain is as follows:

$$y_{\mathrm{rain}} = f\left(W^T \cdot S + C\right), \tag{7}$$

where $W = [w_t, w_h, w_p, w_w, w_r]$ represents he weight for each attribute.

As for training samples,

$$y_{\mathrm{rain}} = \begin{cases} 1, & \text{rainfall more than 50 mm per day or 30 mm per 12 hours} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

5.3. Logistic Regression. Logistic regression, which is used to estimate the probability of the binary response based on one or more variables, is considered to be one of the prevailing models in various economic and medical applications. When logistic function is established to measure the relationship between categorical dependent variable and independent variables, accuracy in categorization is promoted. In our paper, the logistic function is given by

$$P_{\mathrm{rain}} = P\left(y_{\mathrm{rain}} = 1 \mid S\right) = \frac{\exp\left(W^T \cdot S + C\right)}{1 + \exp\left(W^T \cdot S + C\right)}$$
$$\mathrm{logit}\ \left(P_{\mathrm{rain}}\right) = \log\left(\frac{P_{\mathrm{rain}}}{1 - P_{\mathrm{rain}}}\right) = W^T \cdot S + C. \tag{9}$$

For later evaluation,

$$y_{\mathrm{rain}} = \begin{cases} 1, & \mathrm{logit}\left(P_{\mathrm{rain}}\right) > 0.5 \\ 0, & \mathrm{logit}\left(P_{\mathrm{rain}}\right) < 0.5. \end{cases} \tag{10}$$

Therefore, the heavy rain probability estimation problem has been transferred to linear models. From this point of view, our statistical hypothesis is defined as

$$H_0^{(i)} : w_i = 0 \quad (i = t, w, h, p, r). \tag{11}$$

Wald test is also utilized for estimating parameters. We assign the Wald test model described as below:

$$w = \frac{{B_w}^2}{{\mathrm{SE}_w}^2}, \tag{12}$$

where B_w represents the predicted weights w_i $(i = t, w, h, p, r)$ of vector S and SE_w stands for its standard error. Given the level of significance α, hypothesis $H_0^{(i)}$ is denied by

$$p = P\left\{w \geq w_\alpha\right\}. \tag{13}$$

After vector W is fully estimated with affordable Wald test scores, logistic model for rainfall probability is accomplished.

5.4. Multikernel Support Vector Machines. Logistic regression does not make full use of prior knowledge gained by historical

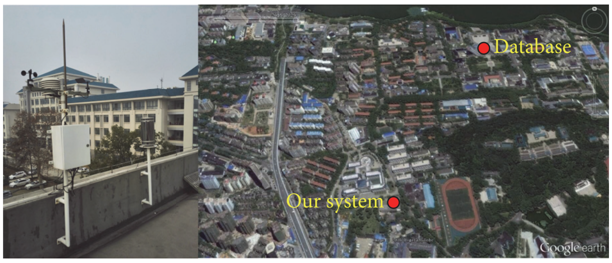


Figure 5: Picture for system installation (map from Google Earth).

data. Support vector machines (SVM) [31] outperform logistic regression in binary classification, especially with growing training dataset [32]. When variables are mapped to higher dimensional space through divergent SVM kernels, nonlinear classification can be achieved by identifying maximum margin hyper plane between two sides [18, 19]. Because weather information like temperature and humidity is highly correlated, kernels that summarize the characteristics of different data transactions may perform better. We introduce two length based SVM kernels to estimate the probability of heavy rain.

Definition 3 (length based SVM kernels). Given datasets S_i, S_j, the length based SVM kernels are defined as

$$k\left(S_i, S_j\right) = f\left(\left\|S_i\right\|, \left\|S_j\right\|\right). \tag{14}$$

Kernel 1:

$$k\left(S_i, S_j\right) = \left\|S_i\right\| \left\|S_j\right\|. \tag{15}$$

Kernel 2:

$$k\left(S_i, S_j\right) = \frac{S_i \cdot S_j}{\left\|S_i\right\| \cdot \left\|S_j\right\|}. \tag{16}$$

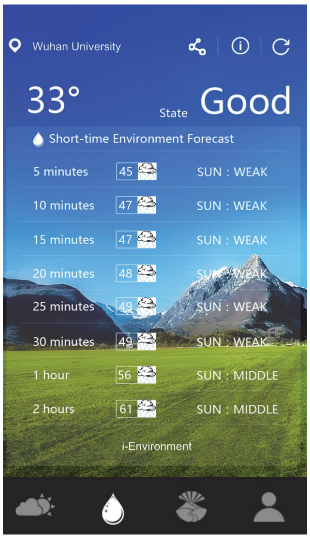
Also we introduce the classic polynomial kernel for comparison.

Polynomial kernel:

$$k\left(S_i, S_j\right) = \left(\frac{S_i \cdot S_j}{\dim\left(S\right)}\right)^2. \tag{17}$$

6. Experiments

6.1. System Installation and Study Area. Our system is installed at the top of the main building on our campus. The whole intelligent sensor node serves as a meteorological station that characterizes the weather of Wuhan University. As Figure 5 shows, multiple sensors are supplied by solar panels and excess power will be stored in specialized batteries for later uses. Also the sensor node is connected to our building's Wi-Fi resources so that local clients (the students and staffs in the laboratory) can receive probability of heavy rain every 5 minutes. All the predictions will be sent to our laboratory (another building that is about 2 miles away through GPRS channels). The sensor platform was installed on June 12, 2015, and our system has been working perfectly so far. All sensor data and results of calculations conducted in Raspberry Pi have enormous meteorological significance, especially in exploring local meteorological differences and early warning of heavy rain. Also we have developed APPs for local client with screenshot as in Figure 6.



Wuhan University

33° State Good

Short-time Environment Forecast

5 minutes	45	SUN : WEAK
10 minutes	47	SUN : WEAK
15 minutes	47	SUN : WEAK
20 minutes	48	SUN : WEAK
25 minutes	49	SUN : WEAK
30 minutes	49	SUN : WEAK
1 hour	56	SUN : MIDDLE
2 hours	61	SUN : MIDDLE

i-Environment

Figure 6: APP screenshot.

Table 2: Study area.

Rainfall forecast tools	Grid/degree	Time sampling rate	Location
ECMWF ERA-interim	0.75 × 0.75	4 times/day	N 30°32.25′ E 114°21.10′
Centralized WSN	0.75 × 0.75	268 times/day	N 30°32.25′ E 114°21.10′
Our ESN	0.75 × 0.75	268 times/day	N 30°32.25′ E 114°21.10′

For rainfall forecast in small grid, European Centre for Medium-Range Weather Forecasts (ECMWF) is broadly applied [37]. ECMWF ERA-interim provides 6-hour forecasts in any 0.75 × 0.75 (longitude and latitude coordinates) degrees' areas. In our experiment, our system and ECMWF predict rainfall in the same 0.75 × 0.75 degrees' region which centers around N 30°32.25′ E 114°21.10′. Also, we have centralized WSN, which treats sensor system as a pure collector and relies heavily on networks. The results are compared and discussed in Section 6.3.

6.2. LTR Dataset. To compare the efficiency of our ESN and ECMWF, we construct a LTR dataset for evaluation. The LTR dataset is a collection of environmental information in the study area, including rainfall, air pressure, temperature, humidity, and solar radiation, both from centralized WSN and our ESN (Table 2). Also, we have ECMWF forecast dataset that gives prediction to rain probability 4 times a day. These two datasets record the weather information and forecast from Sep 26, 2015, to June 15, 2016.

Table 3: LTR dataset.

Total number of transactions	Number of training transactions (our ESN)	Number of training transactions (centralized WSN)	Number of testing transactions
50000	25000	25000	25000

Table 4: ESN fault rate.

Rainfall forecast tools	Number of valid transactions (n)	Expected number of valid transactions ($m \times 268$)	Fault rate
Centralized WSN	25742	70484	36.53%
Our ESN	6198	70484	8.80%

In the LTR dataset, there are 25000 data objects that were randomly selected from Sep 26, 2015, to June 15, 2016. Each transaction contains rainfall, air pressure, temperature, humidity, and solar radiation data. They have been validated as correct. So we can use them as test dataset. Also as Table 3 shows there are 25000 training samples each for centralized WSN and our ESN.

6.3. Comparison and Discussion. In this section, we conduct test to evaluate the effectiveness and efficiency of our ESN, centralized WSN and ECMWF. To test the working time of our ESN and centralized WSN, we compare their fault rate. For the accuracy test, we plot the accuracy of ECMWF, SVM, and logistic regression versus the size of training samples. When we find out that only SVM with our ESN beats ECMWF in LTR estimation, we compare the different SVM kernels in both accuracy and efficiency.

6.3.1. Efficiency between Our ESN and Centralized WSN. Before the probability estimation, we must calculate the downtime when both our ESN and centralized WSN cannot transmit any information. Also default outlier detection method presented in the previous part will be applied on the original dataset. In our work, we use the fault rate as follows to represent the efficiency of the environmental monitoring system.

Definition 4 (ESN fault rate). Given a dataset S, after downtime detection and outlier detection, it still has n valid transactions $\{x_1, x_2, \ldots, x_n\}$, which present the environmental information from our ESN and centralized WSN in m days (Table 4). The fault rate is defined as

$$\mathrm{FR} = \frac{n}{m \times 268}. \tag{18}$$

As Table 3 shows, from Sep 26, 2015, to June 15, 2016, it is obvious that our system has lower fault rate, which implies that our ESN keeps working when traditional centralized WSN stops working. This improvement in efficiency enables our ESN to deal with more LTR situations with more prior knowledge.

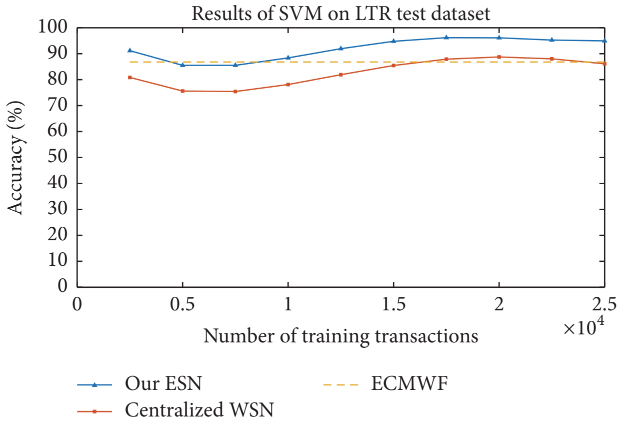


Figure 7: Results of SVM on LTR.

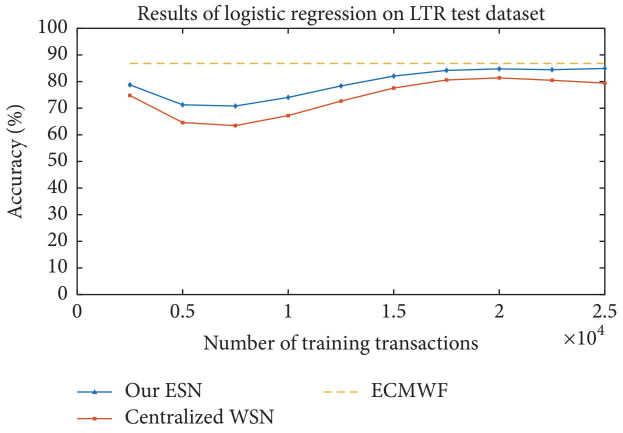


Figure 8: Results of logistic regression on LTR.

6.3.2. Performance of LTR Probability Estimation. After evaluating the fault rate between two WSN architectures, the performances of our ESN, centralized WSN, and ECMWF are tested by LTR dataset. The training data transactions from both centralized WSN and our ESN are learned by linear SVM and logistic regression. In Figures 7 and 8, we plot the accuracy of ECMWF, SVM, and logistic regression on LTR classification versus the size of training samples. As Table 5 and Figures 7 and 8 imply, our ESN has higher accuracy than centralized WSN in classification. And linear SVM have better results than logistic regression in this problem because only the result of SVM from our ESN outperforms ECMWF.

6.3.3. Evaluation of Multiple Kernels on SVM Based Rainfall Probability Estimation. In this subsection, the accuracy and efficiency in rainfall probability estimation are tested for SVM with various kernels. Because SVM have been shown to achieve higher accuracy in rainfall probability estimation, it is meaningful to compare the performance of SVM with different kernels. Also in LTR dataset, meteorological data

Table 5: Best logistic regression models.

Rainfall forecast tools	Our ESN	Centralized WSN
Number of training samples	20000	20000
Accuracy	85.48%	80.48%
Weights/W = $[w_t, w_h, w_p, w_w, w_r]$	[0.947, 0.025, 0.363, 0.102, −0.013]	[0.969, 0.03, 0.37, 0.1, −0.012]
Constant/C	−401.87	−402.85

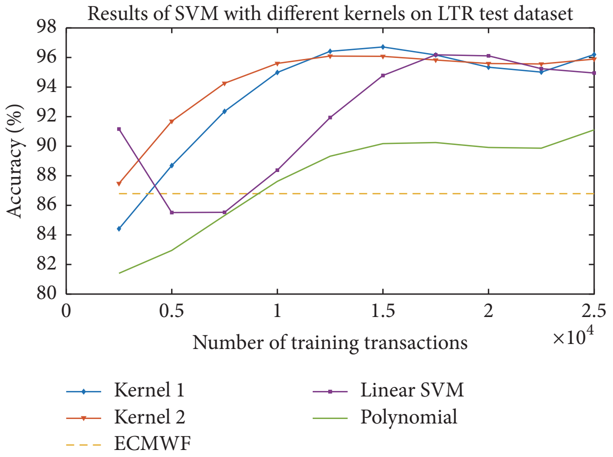


Figure 9: Results of SVM with different kernels on LTR.

attributes are highly correlated so that length based SVM kernels may have better performance.

In Figure 9, we plot the accuracy of various SVM kernels on rainfall probability estimation versus the size of training samples. It is easy to find out that length based kernels estimate rainfall probability better than traditional SVM kernels like linear SVM and polynomial SVM. However, we can see from Figure 10 that length based SVM kernels cost much more time than linear and polynomial SVM kernels. Obviously there is a tradeoff between accuracy and efficiency.

7. Conclusion

Accurate and in-time localized torrential rain monitoring with prior data is a core challenge in environmental sensor networks. Despite a series of wireless sensor networks developed for environmental monitoring, current architecture relies too much on network transportation, which causes information loss and response delay when harsh environment conditions paralyze network. In this paper, we developed a Raspberry Pi based intelligent wireless sensor node that can estimate the probability of LTR according to the self-collected environmental data and publish forecast in localized area using Wi-Fi. Our ESN outperforms centralized WSN in efficiency. Our ESN achieved higher efficiency and lower fault rate than centralized WSN. Moreover, SVM on our system enjoyed higher accuracy in LTR estimation. SVM on our system is tested to have higher accuracy in LTR estimation.

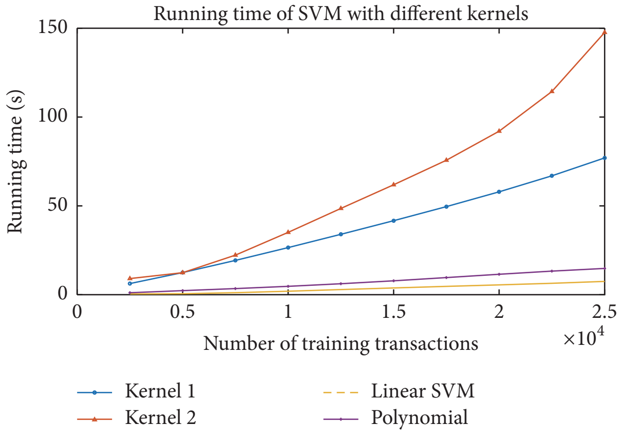


Figure 10: Running time of SVM with different kernels on LTR.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This wok was supported by Major Science and Technology Program for Water Pollution Control and Treatment (2013ZX07105-005). Gratitude is due to Professor Xu, Xin from Wuhan University, who helped the authors construct models and approaches to achieve their goals.

References

[1] M. F. Othman and K. Shazali, "Wireless sensor network applications: a study in environment monitoring system," *Procedia Engineering*, vol. 41, pp. 1204–1210, 2012.

[2] J. K. Hart and K. Martinez, "Environmental sensor networks: a revolution in the earth system science?" *Earth-Science Reviews*, vol. 78, no. 3-4, pp. 177–191, 2006.

[3] N. C. Chen, L. J. Zhou, and Z. Q. Chen, "A sharable and efficient metadata model for heterogeneous earth observation data retrieval in multi-scale flood mapping," *Remote Sensing*, vol. 7, no. 8, pp. 9610–9631, 2015.

[4] D. Liu and C. Q. Qin, "Influence on rainfall run-off due to urbanization in Wuhan city of China," in *Proceedings of the International Conference on Environmental Science and Information Application Technology (ESIAT '09)*, pp. 182–184, IEEE, Wuhan, China, July 2009.

[5] A. Kadri, E. Yaacoub, M. Mushtaha, and A. Abu-Dayya, "Wireless sensor network for real-time air pollution monitoring," in *Proceedings of the 1st International Conference on Communications, Signal Processing, and Their Applications (ICCSPA '13)*, pp. 1–5, IEEE, Sharjah, United Arab Emirates, February 2013.

[6] F. Adamo, F. Attivissimo, C. G. C. Carducci, and A. M. L. Lanzolla, "A smart sensor network for sea water quality monitoring," *IEEE Sensors Journal*, vol. 15, no. 5, pp. 2514–2522, 2015.

[7] K. Hiroi, Y. Seto, F. Matsumoto et al., "Accurate and early detection of localized heavy rain by integrating multivendor sensors in various installation environments," in *Proceedings of the IEEE SENSORS*, Baltimore, Md, USA, November 2013.

[8] L. Calderoni, D. Maio, and P. Palmieri, "Location-aware mobile services for a smart city: design, implementation and deployment," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 7, no. 3, pp. 74–87, 2012.

[9] N. G. S. Campos, D. G. Gomes, F. C. Delicato, A. J. V. Neto, L. Pirmez, and J. N. de Souza, "Autonomic context-aware wireless sensor networks," *Journal of Sensors*, vol. 2015, Article ID 621326, 14 pages, 2015.

[10] L. Calderoni, D. Maio, and S. Rovis, "Deploying a network of smart cameras for traffic monitoring on a 'city kernel'," *Expert Systems with Applications*, vol. 41, no. 2, pp. 502–507, 2014.

[11] J. M. Hernández-Muñoz, J. B. Vercher, L. Muñoz et al., "Smart cities at the forefront of the future internet," in *The Future Internet*, pp. 447–462, Springer, Berlin, Germany, 2011.

[12] J. L. F. Zamora, S. Kashihara, and S. Yamaguchi, "Radio signal-based measurements for localized heavy rain detection using smartphones," in *Proceedings of the 3rd IEEE Global Humanitarian Technology Conference (GHTC '13)*, pp. 173–178, IEEE, San Jose, Calif, USA, October 2013.

[13] G. D. Abrajano and M. Okada, "Compressed sensing based detection of localized heavy rain using microwave network attenuation," in *Proceedings of the 7th European Conference on Antennas and Propagation*, April 2013.

[14] S. Ferdoush and X. Li, "Wireless sensor network system design using raspberry pi andarduino for environmental monitoring applications," in *Proceedings of the 9th International Conference on Future Networks and Communications*, 2014.

[15] S. Gaitan, L. Calderoni, P. Palmieri, M.-C. ten Veldhuis, D. Maio, and M. B. Van Riemsdijk, "From sensing to action: quick and reliable access to information in cities vulnerable to heavy rain," *IEEE Sensors Journal*, vol. 14, no. 12, pp. 4175–4184, 2014.

[16] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[17] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Computing Surveys*, vol. 41, no. 3, article 15, pp. 1–58, 2009.

[18] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, Cambridge, UK, 2000.

[19] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[20] N. Japkowicz, C. Myers, and M. Gluck, "A novelty detection approach to classification," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, vol. 1, pp. 518–523, 1995.

[21] T. Fawcett and F. Provost, "Activity monitoring: noticing interesting changes in behavior," in *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 53–62, ACM, 1999.

[22] G. O. Campos, A. Zimek, J. Sander et al., "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 891–927, 2016.

[23] D. Barbará, C. Domeniconi, and J. P. Rogers, "Detecting outliers using transduction and statistical testing," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*, pp. 55–64, Philadelphia, Pa, USA, August 2006.

[24] P. Chauhan and M. Shukla, "A review on outlier detection techniques on data stream by using different approaches of K-Means algorithm," in *Proceedings of the 2nd International Conference on Advances in Computer Engineering and Applications (ICACEA '15)*, pp. 580–585, IEEE, Ghaziabad, India, March 2015.

[25] R. J. Bolton and D. J. Hand, "Unsupervised profiling methods for fraud detection," in *Proceedings of the Credit Scoring and Credit Control VII*, pp. 235–255, 2001.

[26] S. Wu and S. Wang, "Information-theoretic outlier detection for large-scale categorical data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 589–602, 2013.

[27] W. Lee and D. Xiang, "Information-theoretic measures for anomaly detection," in *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, Calif, USA, May 2000.

[28] S. Mandal and B. U. Choudhury, "Estimation and prediction of maximum daily rainfall at Sagar Island using best fit probability models," *Theoretical and Applied Climatology*, vol. 121, no. 1-2, pp. 87–97, 2015.

[29] N. S. Novella and W. M. Thiaw, "A seasonal rainfall performance probability tool for famine early warning systems," *Journal of Applied Meteorology and Climatology*, 2016.

[30] S. Lee, J.-S. Won, S. W. Jeon, I. Park, and M. J. Lee, "Spatial landslide hazard prediction using rainfall probability and a logistic regression model," *Mathematical Geosciences*, vol. 47, no. 5, pp. 565–589, 2015.

[31] M. Chattopadhyay and S. Chattopadhyay, "Elucidating the role of topological pattern discovery and support vector machine in generating predictive models for Indian summer monsoon rainfall," *Theoretical and Applied Climatology*, vol. 126, no. 1, pp. 93–104, 2016.

[32] S. H. Pour, S. Shahid, and E. S. Chung, "A hybrid model for statistical downscaling of daily rainfall," *Procedia Engineering*, vol. 154, pp. 1424–1430, 2016.

[33] C. L. Muller, L. Chapman, S. Johnston et al., "Crowdsourcing for climate and atmospheric sciences: current status and future potential," *International Journal of Climatology*, vol. 35, no. 11, pp. 3185–3203, 2015.

[34] S. Hussain, N. Schofield, and A. W. Matin, "Design of a web-based application forwireless sensor networks," in *Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA '06)*, pp. 319–326, Krakow, Poland, September 2006.

[35] S.-H. Lee, T. Yigitcanlar, J.-H. Han, and Y.-T. Leem, "Ubiquitous urban infrastructure: infrastructure planning and development in Korea," *Innovation: Management, Policy and Practice*, vol. 10, no. 2-3, pp. 282–292, 2008.

[36] J. W. Park, C. H. Yun, S. W. Rho, Y. W. Lee, and H. S. Jung, "Mobile cloud web-service for U-City," in *Proceedings of the IEEE 9th International Conference on Dependable, Autonomic and Secure Computing (DASC '11)*, pp. 1061–1065, December 2011.

[37] A. Manzato, A. Cicogna, and A. Pucillo, "6-Hour maximum rain in Friuli Venezia Giulia: climatology and ECMWF-based forecasts," *Atmospheric Research*, vol. 169, pp. 465–484, 2016.

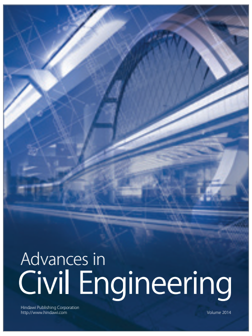
Journal of
Engineering

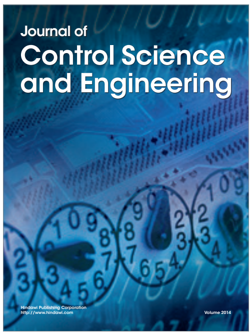
The Scientific
World Journal

International Journal of
Rotating
Machinery

Journal of
Sensors

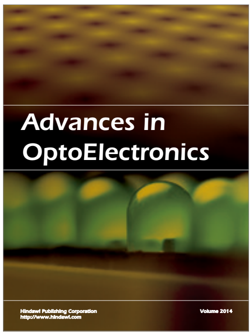
International Journal of
Distributed
Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

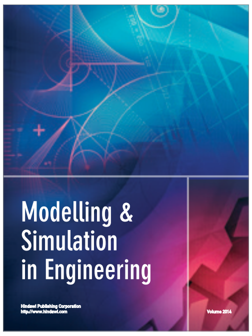
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

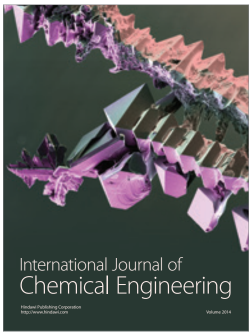

Advances in
OptoElectronics

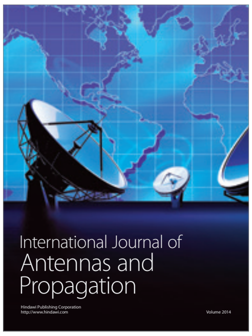
VLSI Design

International Journal of
Navigation and
Observation

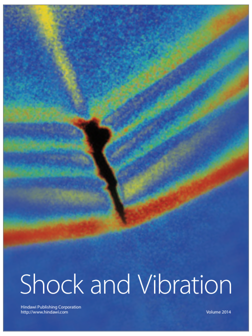

Modelling &
Simulation
in Engineering

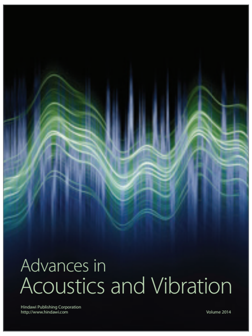
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration