

IoT, big data and HPC based Smart Flood Management Framework

Sandeep K. Sood¹, Rajinder Sandhu^{1,2}, Karan Singla¹, Victor Chang³

san1198@gmail.com, rajsandhu1989@gmail.com, karan_singla82@yahoo.com and ic.victor.chang@gmail.com

1. Department of Computer Science and Engineering,

Guru Nanak Dev University, Regional Campus Gurdaspur
Punjab, India

2. School of Computing and Information Systems, University of Melbourne, Melbourne, Australia

3. International Business School Suzhou, Xi'an Jiaotong Liverpool University, Suzhou, China

Abstract:

The disastrous effect of flood has shown its influence in the past, and as a result, millions of dollars infrastructure have been shattered. Even after so much research, still there is no global ubiquitous system that can collect, store and analyze big data and generate the flood prediction results. In this paper, a social collaborative Internet of Things (IoT) based smart flood monitoring and forecasting architecture is proposed with the convergence between big data and HPC. It classifies geographical areas into a web of hexagonal for effective installation of energy efficient IoT devices. All relevant flood causing and flood preventing attributes are sensed using these IoT devices and computed by big data and HPC processing. Singular Value Decomposition (SVD) is used for attributes reduction. The K-mean clustering algorithm is used to predict the current state of flood and flood rating in any location, whereas Holt-Winter's forecasting method is used to forecast the flood. Experimental evaluation is being done on meteorological data collected by the Indian government and results indicated the effectiveness of the proposed architecture.

Keywords: Internet of Things (IoT); Cloud Computing; big data and HPC convergence; Single value Decomposition (SVD); K-mean Clustering; Holt-Winter's Forecasting.

1. Introduction

Industrial Internet of Things (IIoT) plays essential roles in the development of information technology services, as it provides a platform to serve for the community and people, as a result of convergence between big data and high performance computing (HPC). The concepts of IIoT has been begun since the Year of 1999, in which Ashton explained the concept of IoT in his presentation for the very first time related to future of sensors [1]. In 2005, International Telecommunication Unit furthered the definition of IoT in its report, although cloud computing and IoT technology was in the early stage of its development. With the development, establishment and adoption of cloud computing, IoT has the wider integrates multiple technologies such as sensor networks, big data analytics and cloud computing [3]. IoT integrates services, technology and people together, since it has the connection of digital devices (smart devices or IoT objects) socially collaborated to each other, in order to sense environment, formulate results and communicate effectively to humans or to other digital devices [3-5]. The maturity of IoT has been pervasive to industry such as healthcare services, manufacturing, urban planning, finance, education, emergency services and many more [5-7]. To further establish all IoT services for industry, IIoT can be used, which includes buying services through smart phones using digital scanning to purchase products and receive items in the following days and the connections to all buyers, sellers and online communities through IIoT. It can also include tele-medicine to allow patients to make their appointments, check medical records and payment through the smart devices. It can be adopted to emergency services that survivors and victims of natural disasters can report their safety to their families and friends, and also connect those who have survived, and inform all the victims to places for shelter and protection. This has motivated us to focus on IIoT services related to alarming and predicting effects caused by natural disasters such as flood.

Due to continuous change in earth's climate, natural disasters are increasing in numbers. Flood is the most frequent and destructive disaster. It causes infrastructure loss, threat to lives, and destructions in multiple public utilities. Though the threat to lives by the flood is considerably less than some of the other disasters, but the numbers of lives affected are much higher. According to a prediction in the United Nation survey report, around two billion people will be affected by flood till 2050 and most affected parts will be of Asia. According to Indian national disaster management authority, the average annual damage due to flood during 1996-2005 was 765 million dollars which was double as compared to the corresponding average of last 53 years [8]. In 2013 Uttarakhand flood disaster, the loss of lives reported were more than five thousands, the numbers of peoples affected were 1.6 million and the loss of property was estimated to be 484 million dollars [8]. In 2014, Jammu & Kashmir flood recorded as the worst Indian flood over the last sixty years claiming 250 lives and economic loss of 806 million dollars [8]. The history of floods over the world shows that the most affected country due to floods is China which has lost around 4 million lives in center China flood in 1931 [9].

Perceiving the gravity of the socio-economic consequences of floods, at present there is no effective and efficient global system for their forecast. Researchers have applied their major efforts in the field of spatial proximity like Geographic Information System (GIS) which provides only the real time sensing of the flood. The imagery models like Synthetic Aperture and Radar (SAR), Digital Elevation Model (DEM) have vital scope but not to the extent of early warning system. Modeling, managing and monitoring the flood effectively is a critical task for scientists and researchers. The analysis of hydro-meteorological sensory data techniques have faced pitfall because the ubiquitous accessing, storing and analyzing such a big pool of sensory data is a complex and resource consuming process. The earth consists of many different geographical structures; but still catchments or the area where the water from any source (rainfall, drainage, hailstorm and others) happens to accumulate (lake, pond, river) have certain resemblance among themselves. Therefore, a ubiquitous system can be defined for global flood monitoring because catchments with similar characteristics are likely to behave with similar flood generation responses so most of the research and analysis can be reused. The researches till date face problems of classifying and installing devices at proper central locations hence causing loss of capita and energy. Effective data gathering and storing is another challenge for prediction of floods. Though flood monitoring had been a topic of research for so long, still there is no global unit for defining or scale for measuring the intensity of flood. Above mentioned reasons motivated to design a long term as well as real time flood forecasting system using new technologies such as social collaborative IoT, big data and cloud computing. The recent proliferation of technologies like cloud computing, IoT, and big data has brought new vitality in effective flood monitoring. IIoT is responsible for collecting real time data from physical world, big data analytics is used to effectively analyze the data in real time and cloud computing provides the required IT infrastructure, as a result of convergence between big data and HPC. IIoT is not just a concept but has been transforming into reality; the Gartner IT hype cycle of 2016 shows that IoT for industry has a great potential in the near future [10]. Socially collaborative IIoT with big data and HPC convergence can be effective in predicting and monitoring flood activities but many challenges such as appropriate installation, energy consumption, and data collection corresponding to IoT devices still exist in its successful implementation.

To overcome these challenges in forecasting floods using IIoT, three-layer architecture is proposed in this paper. It classifies the geographical locations into variable size hexagonal geometrical structures for identifying appropriate locations to install IIoT devices. It not only retrieves the data efficiently from the physical world but also deployed in energy saving mode by changing the state of devices to partial active/sleeping states when sensing of data is not required. Its main aim is to collect information using social collaborative IoT devices and make it available for decision making processes. The enormous amount of data generated by social collaborative IoT devices is stored, mined and processed efficiently using dynamic resources of cloud computing. In the middle layer of three-layer architecture, K-mean clustering algorithm is used to categorize the flood rating for any particular catchment. Holt-Winter's forecasting method is used to forecast current as well as long term flood rating using previous and current sensor data. All predictions generated by proposed architecture are effectively communicated to all users and government departments using a mobile application. Major novel contributions of proposed architecture are:

- Representation of geographical locations into hexagonal structures to efficiently install IoT devices.
- Use of Social Network Analysis for energy saving and address resolution in IoT devices.
- Dimensionality reduction of big data using SVD.
- Use of K-mean clustering to categorize the perception of flood in five different levels.
- Holt-Winter's method to predict current as well as future flood rating of each hexagonal.
- Communication of flood prediction results to government departments and all other concerned peoples.

Rest of the paper is structured as follows. Section 2 describes the related work of social collaborative IoT devices and use of cloud computing for digital earth. Section 3 proposes the architecture for effective flood monitoring. Section 4 explains our experimental setup and performance analysis, whereas Section 5 provides the conclusion.

2. Related Work

Successful development of cloud computing nourished many new technologies whose IT resource demands are large and variable. Data generated by IoT devices of no use if it is not analyzed in real time. Cloud provides required IT resources to deal with the large amount of generated data. Many researchers across the globe are interested and studying the combination of these technologies. This section studies related work of proposed architecture in two subsections: social collaborative IoT with big data and HPC convergence based smart environments and cloud computing for providing digital earth services.

2.1 Social Collaborative IoT based Smart Environments

Correlation and dependency between cloud computing and IoT have been studied by many researchers. In summary, IoT platform needs cloud computing for data collection and processing. The fusion between IoT with big data and HPC convergence serves for the purpose of IIoT, which includes examples and case studies for water risk management, physical human security, smart chemical industry, smart environment monitoring, land resource supervision, smart chemical industry, physical human security, smart environment monitoring, smart parking, smart traffic management, smart home, smart manufacturing system, assembly modeling system, smart healthcare, smart agriculture [5-7][11-13]. All these examples have demonstrated how to integrate both cloud computing and IoT together to jointly deliver services and achieve research contributions. Thus, fusion between IoT and cloud computing can be effective for IIoT development, with no exceptions for collaborative research for experts of different domains of specialization.

Integration frameworks between IoT and cloud were discussed with their benefits positively highlighted as follows [11-12; 14-15]. First, Botta et al. [14] surveyed different architectural and application scenarios of cloud computing and IoT and proposed a new name CloudIoT for IoT platforms using cloud computing. Second, Jiang et al. [13] examined the issues of scalability and flexibility of IoT devices in any smart environment. They used near field communication for communication and cloud computing for computation. Third, Renner et al. [16] discussed the challenge of transferring IoT data to cloud computing for processing. They proposed to use resource power of smart devices to pre-process the data before sending it to cloud. Forth, Dai et al. [17] provided a security framework for cloud computing in IoT environments and created a trusted execution environment for cloud end security while computing data of IoT devices. Fifth, Shaoling et al. [18] developed an ant colony based energy consumption model for cloud computing environment using IoT devices. Sixth, Villari et al. [19] recommended that lifecycle management of sensors essential for IoT based smart environments and performed a self-identification process which securely auto-configure sensors with cloud. Seventh, Amato et al. [20] analyzed the need and significance of big data in IoT environments using cloud computing. They argued that IoT environments will generate huge amount of data which need to store, manage and analyze for which big data analytics are necessary. Eighth, Galache et al. [21] presented a CloudT project which uses IoT sensors to help resident of cities of Japan about different available resources. Last, Jiang et al. [22] proposed a storage framework for IoT systems developed on cloud computing. Their system was capable to handle structured and unstructured data using Hadoop file system and initial assessment proved its effectiveness.

2.2 HPC Cloud Computing in Digital Earth

In 2004, Hossain et al. [23] acknowledged that any occurrence of error in remote sensing of rainfall by satellite can lead to high uncertainties in the flood prediction. In 2005, Merz and Blöschl [24] furnished that most of the research in flood monitoring is using spatial proximity like geospatial approach or catchment attributes like index flood approach or the combination of these two. They argued that combination of spatial proximity and catchment attributes yield the best predictive performance. In 2005, Nayak et al. [25] demonstrated that fuzzy models can obtain benefits of their ability to create the unidentified relationships among a set of significant hydrological data such as river flow and rain fall. In 2013, Gourbesville et al. [26] proposed that ubiquitous computing in relation to flood warning systems resulted in better supervision of warning systems. In 2013, Nguyen et al. [27] used adaptive neuro-fuzzy inference system to forecast water levels at three stations. The flow conditions at different locations along the mainstream can differ in terms of the contributions from upstream stations and contributions from rainfall and lateral flows from sub-basins. When upstream water levels in the mainstream were used in the input, it was found that forecasts improved only when the water level from one or at most two upstream stations was included. This is because when there are significant contributions of flow from the tributaries, the correlation between the water levels in the upstream stations and stations of interest decrease, limiting the effectiveness in achieving better forecasts by including the water levels from additional upstream stations as inputs. In 2013, Quiroga and Popescu [28] concluded that cluster computing and cloud computing is an efficient and effective technology for flood monitoring system in setting up a practical prospect even for complex scenarios. In 2014, Shih et al. [29] suggested that hydro-meteorological system was good to warn against potentially harmful floods in the near future, but the system was not good enough to forecast the magnitude and time of the peaks. In 2014, Lohani et al. [30] found that it was very important in flood monitoring to know the efficiency of flow monitoring model in order to predict higher magnitude flow. In 2014, He and Xu [31] envisioned that the IoT technology is fueled by the revolutionary wireless technologies such as embedded sensors, actuator nodes, RFID tags etc. in transforming internet into future integrated internet technology. In 2015, Fang et al. [32] explained that the integrated work based on IoT, cloud computing, GPS, and GIS would be the basis for environmental resources and management in near future.

3. Proposed Architecture for IIoT

An efficient flood monitoring and forecasting has been dependent on multiple factors performed in a sequence: correct data gathering, effective data analysis and timely result notifications. To cover all these factors, a three-layer architecture is proposed in this paper as shown in Figure 1. These factors depend on outputs of previous factors which encouraged to propose an architecture of layered structure. Three layers of proposed architecture are: lower most layer known as IIoT Layer (IL), middle layer known as Data Analysis Layer (DAL) and upper most layer known as Presentation Layer (PL). Each layer performs certain set of activities independent of other layers and provides outputs to higher layers. IL is responsible for installation of IIoT devices and data collection from installed IIoT devices. DAL is responsible for analysis of collected data to effectively forecast the flood in any geographical area. PL is responsible for timely distribution of information generated from DAL in the form of RSS feeds, emails, messages, alerts and notifications. Each layer is explained in detail ahead.

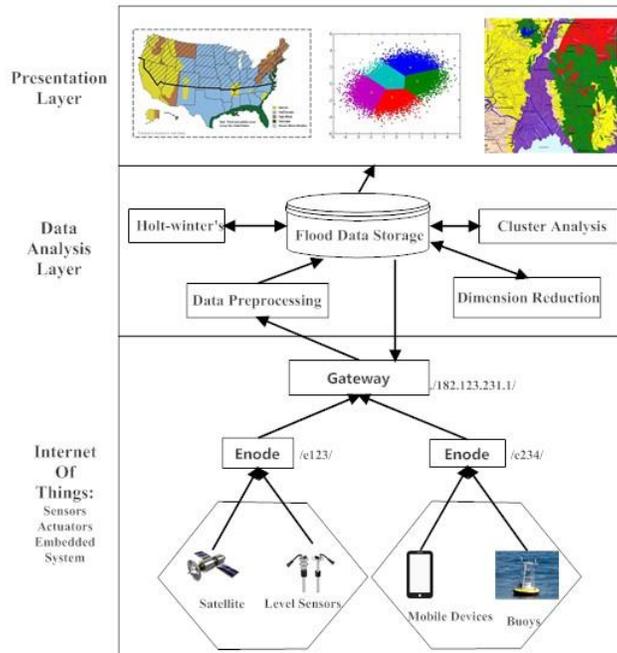


Figure 1 Three Layered Architecture

3.1 IIoT Layer with big data and HPC convergence

The functionalities of this layer are flood attributes selection, geographical classification in form of hexagonal structures, social collaboration of IIoT devices, and energy saving of IIoT devices. All these functionalities are explained in detail ahead.

3.1.1 Flood Attributes

Flood attributes can be classified in two main categories which are Flood Causing Attributes (FCAs) and Flood Preventing Attributes (FPAs).

Table 1 Flood Attributes

Category	Attributes	Increasing Order of occurrence →				
Flood Causing Attributes	Mean Precipitation	Very Less	Less	Moderate	High	Extreme
	Velocity of Catchment	Still	Less Flow	Moderate Flow	High Flow	Extreme Flow
	Stream Order	1	2	3	4	>4~5
	Density of Forests per Acre	Dense Forest (1)	Thick Forest (2)	Little Forest (3)	Cropland (4)	Barren Land (5)

Flood Preventing Attributes		>1000	800-1000	500-800	100-500	<100
	Current Season	Winter (1)	Spring (2)	Autumn (3)	Summer (4)	Monsoon (5)
	Drainage system	Best (1)	Good (2)	Average (3)	Bad (4)	Worst (5)
	Soil type	Sandy (1)	Silt (2)	Organic (3)	Clay (4)	Lummy (5)

FCAs are dynamic in nature and changes often with time but also provide current state of flood. FPAs help in preventing the flood spread rate and these are static for longer duration of time. Both these types of attributes are used to monitor and forecast any flood for geographical location. From large number of flood parameters, Table 1 shows list of important flood parameters classified in FCAs and FPAs which are used in proposed architecture. To remove the sudden fluctuations in flood parameters, they are also sorted in five different clusters as listed in Table 1.

Density of forests is measured as a factor called Stand Density Index (SDI) which can be calculated as $\log_{10} SDI = \log_{10} N + 1.65 \log_{10} D - 1.605$, where N is the number of trees per acre and D is the quadratic mean diameter calculated as $D = \sqrt{\frac{area}{N \cdot 0.05454}}$, velocity of catchment can be obtained from Manning's formula which is $V =$

$\frac{1}{n} R_h^{\frac{2}{3}} S^{\frac{1}{2}}$, where V is the cross-sectional average velocity, n is manning's constant, the value of n varies from 0.02 to 0.08 depending upon soil type of the catchment, S is the slope of the catchment and R_h is the hydraulic radius.

To collect flood parameters, appropriate installation of IIoT devices is significant for research, so that it can collect data of different sources, sizes, quantities and varieties for big data. It can connect to HPC backend so that processing and analysis of data can be completed at a faster rate. Over-installed IIoT devices will increase capital cost, maintenance cost and will generate redundant data. Under-installed IIoT devices will not able to collect all information about flood parameters for any geographical location which will decrease the overall effectiveness of forecasting process.

3.1.2 Hexagonal Classification per Catchment

Main challenge of flood monitoring using IoT devices is the effective distribution of geographical areas such that sensors data will not overlap. It will prevent the wastage of capital investment by providing appropriate number of required IoT devices as well as it will gather required data for effective flood monitoring. To tackle this challenge, variable size hexagonal structure based web is used in the proposed architecture. The advantage of using hexagonal shapes is that no area will be left uncovered and each hexagonal center is equidistant to another hexagonal center. An example of the proposed structure is shown in Figure 2, which clearly shows an origin (0,0) and the axis identifying each hexagon uniquely.

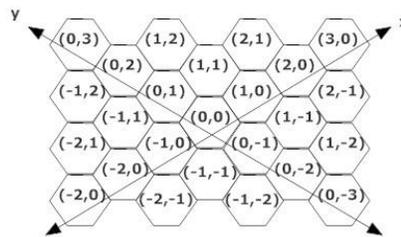


Figure 2 Hexagonal Classification

The proposed geographical classification is two dimensional system i.e. the abscissa and the ordinate where the abscissa represents the land inclination and the ordinate represents the width of the area (y). The hexagons are mapped on the geographical areas in such a way that the declination of the area is from negative x-axis towards positive x-axis. This is in context that the level of flood in hexagonal at lower elevation point will always be derived from a hexagonal at a higher elevation point. Proposed technique can be used to predict flash floods and current stage of any flood. Size of hexagonal is also variable depending on the probability of getting floods in any area. Size of hexagonal will be large in areas which are less prone to flood whereas size of hexagonal will be small in areas with high probability of flood. This will further help to reduce the cost required to install less number of devices in areas where chances of flood are very rare.

The size of the hexagon and the number of IoT to be installed are interdependent as different IoT devices have defined range of excellency. IoT devices range will be decided on the bases of number of hexagonal it require to cover. Here in Algorithm 1, we assumed that each IoT device covers one hexagonal region to maximize the collection of data at the optimum rate, as well as improving the rate of data transfer for HPC processing at the backend. Proposed hexagonal based classification makes geographical classification more adaptive and reliable. Algorithm 1 provides the method of installing IoT devices in hexagonal based on their respective range.

Algorithm 1: Hexagonal Division of Geographical Area

Step 1: Select the most sensitive area of the catchment as the origin.

Step 2: Set the directions of the co-ordinate axes:

x-axis: from highest elevation point (-x) towards lowest elevation point (+x).

y-axis: width of the catchment.

Step 3: Set the radius of each hexagon as half the range of the devices (R/2)
(for each hexagon centers are equidistant).

Step 4: Install each IoT device at center of hexagonal to cover the area of radius (R/2) and connect to gateway.

Step 5: Record the latitude and longitude of the origin for Geo-position referencing.

3.1.3 Social Collaboration of IIoT Devices

Effective connection and communication management of IIoT devices with each other is also very important aspect of any IIoT based smart environment. Proposed architecture uses a social network analysis dependent hierarchical approach for sending and receiving the data. As shown in Figure 1, IoT devices are connected to their enode which is connected to the gateway, a passage way to the cloud, so that a hierarchical structure is in place. All IIoT devices, enodes and gateway are installed and their connection to each other is represented as a social network graph. This deployment is similar to the HPC that nodes should be in hierarchical structure, so that the outputs of the analysis can be enhanced.

Correct identification of each device at the cloud layer is another difficulty as the Ipv4 or Ipv6 addresses are very limited as compared to number of devices to be installed. Therefore, while setting up IoT devices instead of exhausting limited Ipv4 or Ipv6 addresses on IoT devices, a method of Uniform Resource Name (URN) is used [3]. It will remove the need of assigning IP address to each IoT device. Each IIoT device as explained above has been classified into a unique hexagonal structure. Single enode has multiple hexagonal structures connected for receiving and sending the data. An illustration “./172.31.15.16/e123/32/43/M56g/dfy.jpeg” makes up a unique URN for an IIoT device which generates images of the geographical area. URN provides following information: gateway IP is 172.31.15.16, enode number is e123, abscissa is 32, ordinate is 43, device id is M56g and image file is dfy.jpeg. URN name method resolves the issue of uniquely identifying each IIoT device among large number of deployed IIoT devices.

3.1.4 Energy saving at IIoT

The large numbers of devices installed are continuously monitoring the catchment and sending the data for analysis hence exhausting huge amount of energy per second. When there is less probability of flood in any geographical location, sensing of data by large numbers of devices is merely wastage of energy. In order to preserve this energy, a systematic shutdown and activation of devices is proposed in the architecture. The classification of catchment in five different colors based on probability of flood is as shown in Figure 3. We provide appropriate solution by putting some of the IIoT devices in sleep mode where chances of flood are negligible. There are maximum five flood levels defined for a system and the current flood level of the catchment is say 1 (very low), then all the devices that lies in that hexagonal structure can be shut down according to Algorithm 2. The steps involved in this shutdown and activation are shown in Algorithm 2 ahead.

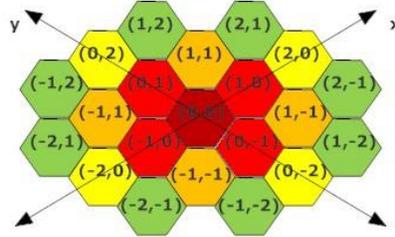


Figure 3 Device shutdown coloring

The RMS distance of different hexagons is shown in different colors in Figure 3. The hexagons in maroon colors are at RMS distance of one, the orange hexagons are at RMS distance of square root two and so on. Additionally, some more energy saving can be achieved by merely decreasing the frequency of the data transmission when there is a very low possibility of flood in the catchment and vice-versa. Conclusively, the overall energy consumption of the system can be reduced widely. Thus, the IIoT layer of proposed architecture classified the catchments successfully, saved energy and delivers the data to be processed by the cloud layer, and the input data can be used by HPC resources for further analysis.

Algorithm 2: Energy saving

N: number of levels of flood

T: maximum time interval between two data transmissions

H(x, y): IoT device co-ordinates of any hexagon

Step 1: Obtain the current flood level of the hexagon at origin (n).

Step 2: calculate S as: $S = (N - n)$

Step 3: for (i = -x to +x)

 for (j = -y to +y)

 Calculate the RMS distance $s = \sqrt{x^2 + y^2}$

 If ($s \geq S$)

 Activate H (i,j)

 Else

 Shutdown H (i,j)

 End for

End for

Step 5: Calculate new time interval between two transmissions (t) as: $t = \frac{T}{\sqrt{n}}$

3.2 Data Analysis Layer (DAL)

Data Analysis Layer (DAL) is a core part of any big data based smart system. It is responsible for receiving, storing and analyzing data for obtaining results. This section explains the algorithms used to monitor as well as forecast the floods. FCAs data entries are dynamically changing with time whereas FPAs data entries are constant. In proposed architecture, Singular Value Decomposition (SVD) is used to convert these multiple values to two attributes (abscissa and ordinate) so that more accurate prediction can be made. Thus, by finding those attributes that repeatedly represent the data, processing time as well as sudden data fluctuations can be removed. SVD is a multifunctional approach that has shown its potential in various applications. It is a method that finds the order of dimension along which the data exhibits most variation. It is also used to show better relationships in uncorrelated data than in correlated data. The basic idea of using SVD for dimensional reduction is to bring out the substructure of the original data in the decreasing order of variation. Also it is of interest that at least the threshold relationships among data are still maintained. SVD is a mathematical theorem that explains that a matrix can be decomposed into the product of three matrices, the left singular matrix, the diagonal matrix and the right singular matrix. The Algorithm 3 below signifies step by step procedure about how to obtain the reduced matrix $A_{m(n-\tau)}$ with fewer dimension but with no reduced importance.

Algorithm 3: SVD

A: new data matrix of FPAs before reduction.

m: number of tuples or all the data entries received in certain time intervals t.

n: number of FPAs attributes or the dimensions.

τ : threshold value representing maximum number of dimension that are to be reduced.

Step1: factorize the matrix A in the form: $A_{mn} = U_{mm} S_{mn} V_{nn}^t$

Step 2: find U_{mm} such that $U^t U = I$

Columns of U are orthogonal Eigen vectors of AA^t .

Step 3: find S_{mn} such that S is singular and is diagonal.

such that the values are the square root of Eigen values from U or V in decreasing order.

Also find $S_{m(n-\tau)}^1$ by deleting last τ rows and last τ column of S.

Step 4: find V_{nn} such that $V^t V = I$

Columns of V are orthogonal Eigen vectors of $A^t A$.

Also find $V_{(n-\tau)(n-\tau)}^1$ by deleting last τ columns of V.

Step 5: find $A_{m(n-\tau)} = U_{mm} S_{m(n-\tau)}^1 (V_{(n-\tau)(n-\tau)}^1)^t$.

$A_{m(n-\tau)}$ is the new matrix with τ number of reduced dimensions. The SVD is triggered by selecting some random time (say t) such that it reduces all the new set of values to fewer dimensions and update the database with just a small computation. Now, that the data is obtained and preprocessed, it must now be analyzed in order to define its flood level intensity based on the data obtained.

3.2.1 Flood rating for hexagonal

The flood is generally measured by the height of catchment and no general unit has been employed. There is a need to define a unit for floods similar to Richter scale for earthquake. Proposed unit should indicate the severity of the flood and it should also represent the flood rating for any geographical area. Data clustering approach has been used in the architecture for flood rating. There are various clustering methods but K-mean is sensitive to number of clusters and initial data input. It could be of more use by defining the number of clusters and the predate for maximum efficiency. The value of K is going to define the intensity levels of flood i.e. 0 level means no chance of flood and K level means high alert for the area under impact. Since K-mean clustering is sensitive to initial data cluster therefore the system must be trained by defining some initial values of the different cluster centers. The Algorithm 4 ahead explains how to define the predate and also the clustering method. Now, every time a new data set arrives, it not only defines its cluster center but also upgrades itself by minimizing the cluster distance. All this functioning ultimately defines each hexagonal with a cluster number representing the flood level of the hexagonal.

Algorithm 4: K-mean

K: number of cluster center

Q_i: Quartiles in a data range of a data set D

D: the data matrix of order n*d

d: no of attributes in D

n: number of tuples or all the data entries received in certain time intervals t

k: point at which it reaches the saturation point

μ_j : jth cluster center (j=1,2,...,K)

$(\mu_1, \mu_2 \dots \mu_k) \in D$

Step 1: Divide the range of possible values in d into k Quartiles and assign to all the clusters centers

For (i = 1 to K)

$$\mu_i = \|Q_i\|;$$

Step 2: Assign all the data points (x_i) to cluster centers (μ_j) whose distance from μ_j is minimum out of all μ_j .

$$a[i][j] = \begin{cases} 1 & \text{if } j = \arg \min_j \|x_i - \mu_j\|^2 \\ 0 & \text{else} \end{cases}$$

Step 3: Redefine the new cluster center

$$\mu_j = \frac{1}{n_j} \sum_{i=1}^n a_i x_i$$

Step 4: Recalculate the distance between each x_i and the new μ_j .

Step 5: If there is no change in the assignment of data point then stop

Else goto step 2.

Step 6: if (n > k)

goto step 1.

After defining the cluster value for each hexagonal, the picture of current status of flood will be clear. This current value of the level of flood and all its previous dataset could be helpful in predicting the flood not only in real time but also in long term.

3.2.2 Holt-Winter's forecasting

Forecasting a flood is the real time interpretation of various flood causing attributes. Holt-Winter's method of forecasting is a well proven and popular method of forecasting used worldwide by many industries for forecasting demand. Holt-Winter's method also called as multiplicative exponentially weighted moving average is a way of smoothing unsystematic fluctuations by putting declining weight on older data. This method not only smoothes current unsystematic fluctuations but also revises continuously the level, seasonal and trend adjustments. These are then extrapolated into the future for flood forecasts. In application to forecasting the flood, the five levels of flood values are evaluated from the database by this method and objective is presumed. Since the fluctuations are changing proportionally, the method applied would be the multiplicative seasonality instead of additive seasonality i.e. forecasting a linear trend with multiplicative seasonality. Now as soon as the system receives a new data point, the K-mean define its new cluster center and Holt's method forecasts the values, there is a need to adjust the values using the hexagonal pre-node and post-node values. If the new cluster center is more than certain threshold cluster centers say α , then it triggers all the post nodes. The K-mean defines the new cluster center for the post nodes using the value of cluster center of the pre-node and the new data points.

Algorithm 5: Holt-Winter's Method

C_t : the change rate adjustment,

T_t : the trend adjustment,

S_t : the seasonal adjustment,

m : period after which the values start repeating itself,

n : time period for a season,

t : current time period,

α, β, γ : level, trend, seasonal smoothing factor that determines how fast the exponential value decreases over time, their values belong to range $[0,1]$,

\bar{y}_1 : the average of all the actual values of y for m periods,

(x, y) : co-ordinates of any enode

Step 1: initialize the level, trend and season at $t = 0$;

$$\begin{aligned} (\text{level at time } 0)C_o &= \bar{y}_1 - \frac{n}{2}T_o \\ (\text{trend at } t = 0)T_o &= \frac{y_n - y_1}{(n-1)m} \\ (\text{season at } t = 0)S_o &= \frac{\bar{y}_t}{\bar{y}_i - \left(\frac{m-1}{2} - j\right)T_o} \end{aligned}$$

\bar{y}_t is the average for the season equivalent to the t , and j is the position of the period t within the season, the equation derives first m values of the seasonal component S .

Step 2: compute the new value for the level adjustment

$$C_t = \alpha \frac{y_t}{S_{t-m}} + (1 - \alpha)(C_{t-1} + T_{t-1})$$

Step 3: compute the new value for the trend adjustment

$$T_t = \beta(C_t - C_{t-1}) + (1 - \beta)T_{t-1}$$

Step 4: compute the new value for the season adjustment

$$S_t = (\gamma) \frac{y_t}{C_{t-1}} + (1 - \gamma)S_{t-m}$$

Step 5: forecast the value for $t = t + f$ as follows: $F_{t+f} = (C_t + fT_t)S_{t-m+f}$

Step 6: readjust the post-node values

for ($i = -y$ to y)

for ($j = -x$ to x)

if ($F_{t+f}(\text{enode}(i,j)) - F_{t+f}(\text{enode}(i+1, j)) > x$)

$F_{t+f}(\text{enode}(i+1, j)) = F_{t+f} - x$;

End for;

// i=1 signifies the root node, x is some threshold difference, enode_i is the pre-node of enode_j

The different functions of the DAL layer are detailed above, the DAL analyze the data and produces the results. The functions performed by the DAL in the system would be no use if they are not delivered in detail and meaningfully. Thus, the role of the third layer, the presentation layer is also crucial.

3.3 Presentation layer

The presentation layer is the user's view of the system. The results that are attained after processing all the information are accessible at this layer. These results are in the form of different graphs or tables or even warnings that represent the current situation or the forecast situation. These graphs are processed on the cloud and can be accessed from anywhere round the world. Based on the results, the safety measurement can be triggered either manually or automatic as per requirement. Certain ways like voronoi diagrams can be used, in which the results in detail are generated. These results are accessed based on the user end application that can be designed for the purpose of User Interface (UI), an example of such interface can be android application or simple website.

4. Experimental Setup

The experimental setup is classified into three stages which are data collection, data analysis and user interface with the big data and HPC convergence, as shown in Figure 4. These stages are discussed in detail ahead.

4.1 The Data Collection for big data

The layered architecture has different functions at each layer. The lowermost layer aims at physical classification of the catchment and installation of large number of IIoT devices. Even though one sets-up such costly system for the experimental purposes, one cannot test the system until it actually faces a real flood situation at that location. Therefore, to test the efficiency of the system instead of actually setting up such cost adhering system, it is emulated. The most important part of any system being emulated is its data. For the emulation of this system though the data cannot be obtained directly from the environment but can be obtained from the various official data sources. A large quantity of good data was taken for experimentation from official website <http://www.indiawaterportal.org> [33]. It is an agency that coordinates with all kinds of disaster including flood. The historical weather data is available on their official website, the data available is with the frequency of one month.

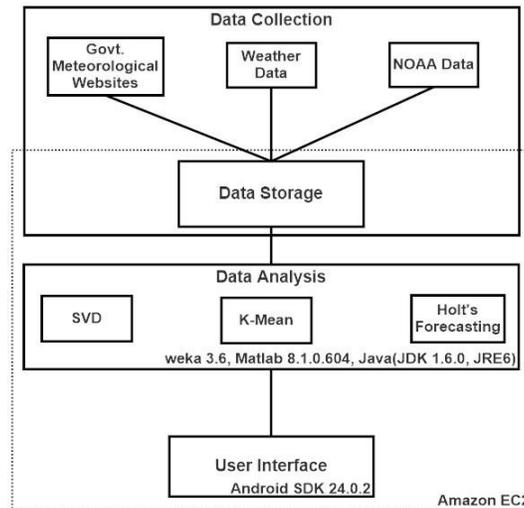


Figure 4 Experimental Testbed

Weather forecasting widgets can be directly integrated to Amazon EC2 cloud. The data can also be retrieved from various nation or state government websites [34-36]. The data considered for analysis is from the Utrakhand, India for the area which has faced flood frequently in recent times and many villages are affected by the same [33]. The raw data from various sources is stored at the cloud storage, the data is then analyzed to produce results. The FCAs discussed in Section 3.1.1 has two main attributes as mean precipitation and velocity of the catchment. The variation in mean precipitation can be seen in Figure 5 below. The variations show mainly the four peak points of mean

precipitation, corresponding to those flood points in the data. These floods are in the month of July 2011, July 2012, July 2013 and July to August 2015. That is the floods are occurring in the season of monsoon, so special care for flood protection is required during monsoon season. The variation in the velocity of catchment is proportional to the variation in mean precipitation. That is why in Figure 6, all peaks correspond to the peaks of Figure 5.

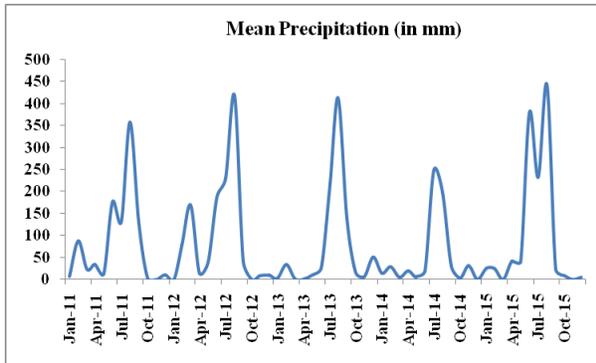


Figure 5 Mean Precipitation

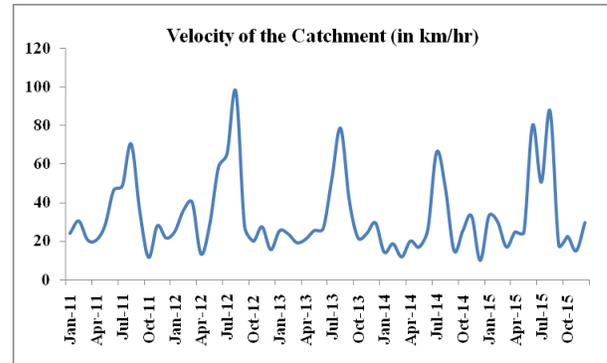


Figure 6 Velocity of the catchment

Similarly, the FPAs as stream order, density of forests, current stream, drainage system and soil type are collectively shown in Figure 7 ahead. Current season is according to the rain season and causes highest chances of flood in monsoon season. Alarming point is decrease in the density of forests which is the major cause of floods. Drainage system should be improved for proper water flow. Water absorbent soil can decrease the chances of flood. The FPAs can be improved over the period of time by taking above mentioned measures to control the floods. This scientific step has used big data techniques to collect and synthesize data.

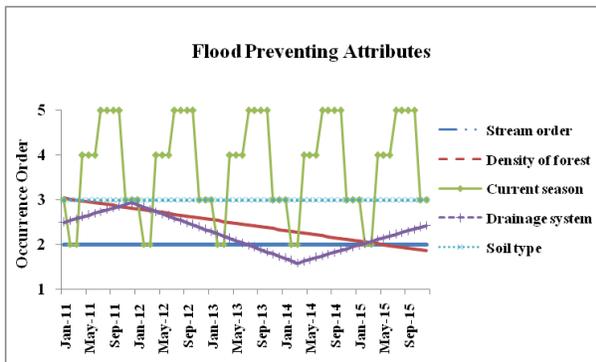


Figure 7 FPA attributes

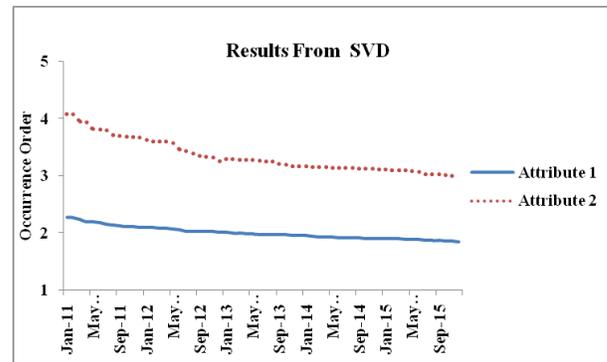


Figure 8 Results from SVD

4.2 Data Analysis

During data analysis, data gathered in the cloud is dimensionally reduced using SVD. The SVD reduces the FPAs as explained in the Section 3.2. The SVD algorithm is available in the Weka 3.6 tool and is applied directly on the cloud storage using cloud platform, and is also a algorithm that can be used by HPC, so that completion time can be reduced. Though the time compilation for PCA and SVD are same, the efficiency of SVD increases with numbers of datasets. The results generated by the SVD are shown in Figure 8. The five attributes are reduced to two attributes that are of utmost importance to the system. After applying SVD, the K-mean algorithm clusterifies the data. The K-mean algorithm is applied directly to the cloud storage and HPC processing. As a useful method for HPC, the K-mean assigned the current flood level to each node and thus signifies the current situation. The results generated by K-mean show the number of times a node or area in Uttarakhand that was under its impact from level 1 to level 5, as shown in Figure 9. This method based on HPC K-clustering will allow scientists to know the behaviors of flood and weathers over a period of time, so that any abnormal observations can be obtained and recorded.

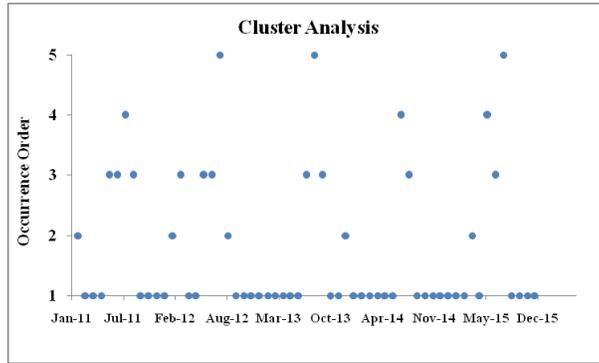


Figure 9 K-mean Results

4.3 Holt-Winter's Forecasting

The results generated by K-mean are used by the Holt-Winter's method. The Holt Winter's method has been implemented in java and is made to run on EC2 cloud center. The Holt-Winter's method not only depends upon results from K-mean but also depends on UI stage for certain input like time span for which predictions are to be made. For emergency purposes, the Holt-Winter's method is run in parallel to K-mean for recording some advance forecasting values say for example the forecasted values for time span of one month, three months, six months, one year etc. This pre-determined automatic data saves time during emergency services. All these results from SVD, K-mean and Holt-Winter's method are available at the UI stage in different forms. The results from Holt-Winter's method are shown in the form of graphs as shown in the following figures; Figure 10 represents forecasting for one month, Figure 11 for three months, Figure 12 for six months and Figure 13 for one year.

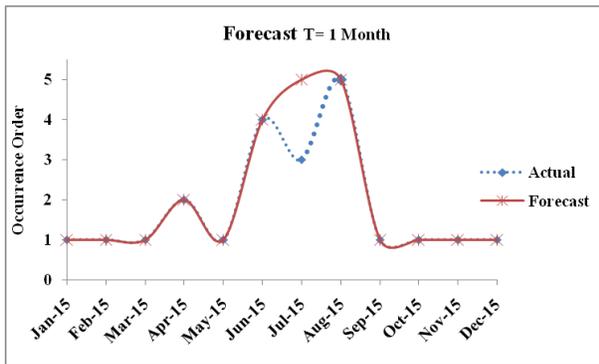


Figure 10 forecast for one month

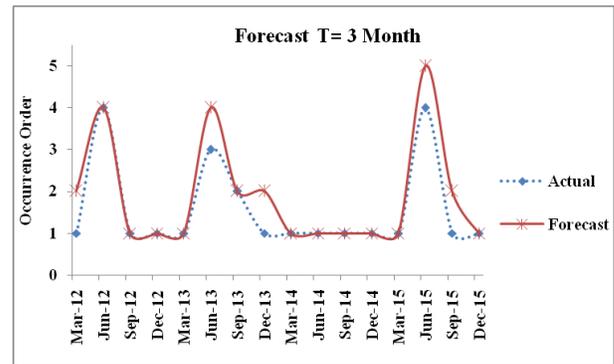


Figure 11 forecast for three month

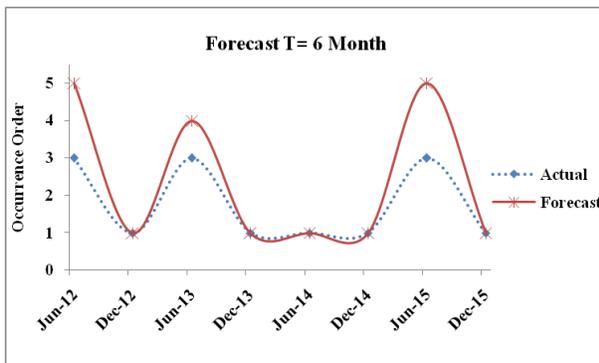


Figure 12 forecast for six month

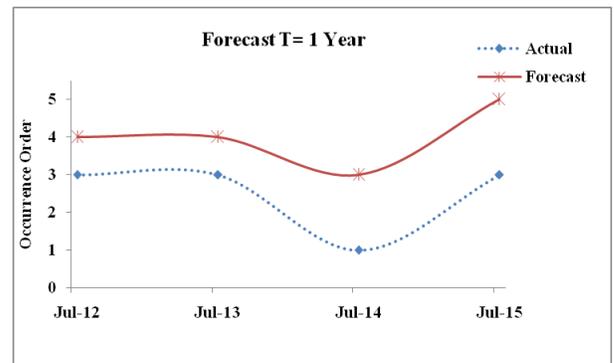


Figure 13 forecast for one year

4.4 Discussion of Results

The results obtained in Figure 10 to Figure 13 show variations between forecast and actual values. Various statistical parameters as MAE, MSE, MAPE and U-statistics computed by HPC resources, with results being shown in Table 2. These are the indicators to test accuracy and robustness of scientific proposed models [37]. The results reflected that the duration in which flood predicted actually observed during that time span. When data is analyzed month wise, forecasting and actual results were very close to each other as reflected in Figure 10 and statistical parameters in Table 2 varies its accuracy. When data is analyzed for three months, six months and one year, difference in forecasting and actual prediction increases and thus error rate also increases as shown in Table 2. It happens because frequency of observed data has been reduced. Our proposed methodology can sense data continuously using IIoT devices, stored on cloud, processed by HPC and synthesized and analyzed by big data analytics. Results in Table 2 support the robustness and accuracy of our IIoT flood management service for big data-HPC convergence thoroughly tested over a period of one year.

Table 2 Obtained Results

Time Period	One Month	Three Month	Six Month	One Year
Mean Absolute Error (MAE)	0.0416667	0.1041667	0.1041667	0.125
Mean Squared Error (MSE)	0.0833333	0.1041667	0.1875	0.2083333
Mean Absolute Percentage Error (MAPE)	1.3888889%	7.4652778%	3.4722222%	6.9444444%
U-Statistics	0.002062	0.0023219	0.004204	0.0048557

Big data has been collected and synthesized by our IoT services, whereby HPC is used to accelerate data processing and reduce complexity and completion time. In this way, the convergence between big data and HPC has been achieved for delivering IIoT service. Henceforth, flood forecasting efficiency can be enhanced. When the predicted results can be released earlier rather than later, it allows the scientists and public to be aware of the likely consequences as a result of flood, so that they have more time to prepare for emergency, escape and rescue actions. Examples and key lessons for the weather analysis and forecasting services in [38] will be further integrated to improve the capabilities and functionalities of our services.

In general, our work provides research contributions to sustainable computing with big data and HPC convergence. First, it provides quality of service since it can analyze data and make accurate predictions for flood management. Second, the architecture is an efficient runtime system to make the big data-HPC convergence useful and effective. Third, the IoT service has the heterogeneous architecture and the embedded system, with sensors to collect data and the HPC to process data. Forth, our IoT service has three layers which can be easily scalable and reconfigurable, and can be deployed in any suitable laboratories. Last, our work has demonstrated the data reduction for big data and HPC convergence since our service can continuously analyze data in the past one year to achieve good accuracies.

5. Conclusion and Future Work

Social collaborative IIoT along with big data-HPC convergence can be used to monitor and forecast floods more effectively and efficiently. Proposed system provides architecture to make this vision a reality. Main key point of proposed architecture is to represent geographical areas into different size hexagonal structures based on FCAs and FPAs and to correctly identify the locations and numbers of energy efficient IIoT devices. For data analysis, SVD is used for dimensionality reduction, K-mean clustering to predict the sensitivity level of flood in catchment and Holt-Winter's method is used for flood forecasting according to flood probability. Experimental results show that proposed system can be quite effective to predict the floods probability efficiently. Big data and HPC convergence has been successfully delivered to make this service to provide accurate and reliable predictions, so that flood management can be managed with better outcomes to serve for the scientists and the public.

In this paper, we present our case of big data-HPC convergent IIoT service for flood management and demonstrate the usefulness and effectiveness of IIoT, including full explanations for our architecture, algorithms, steps for conducting experiments, results and analysis. Our flood management service can contribute to IoT, big data and HPC convergence and can offer a more reliable forecasting and analysis of flood, such as using business intelligence approach [37], together with modern forecasting and data visualization techniques [38]. While more data can be collected, the management and security aspects have become more challenging, since it is highly essential to keep all results and data protected and safe against unauthorized access [39]. In the event of natural disasters that may cause damage, abilities to recover data and resume services without disruption become important [40]. Our future work will include improving accuracy, development of disaster recovery and security. Future work can include the development of mobile application for appropriate broadcasting of results so that the public can be alerted in due time.

References

- [1] K. Ashton, "That 'Internet of Things' Thing," *RFID Journal*, 2009. [Online]. Available: <http://www.rfidjournal.com/articles/view?4986>. Last Accessed on 25th August 2016.
- [2] "ITU Internet Reports 2005: The Internet of Things." [Online]. Available: <https://www.itu.int/net/wsis/tunis/newsroom/stats/The-Internet-of-Things-2005.pdf>. Last Accessed on 25th January 2016.
- [3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [4] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.
- [5] B. Xu, L. Da Xu, H. Cai, C. Xie, J. Hu, and F. Bu, "Ubiquitous Data Accessing Method in IOT-based Information System for Emergency Medical Services," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1578–1586, 2014.
- [6] Zhuming Bi, Li Da Xu, and Chengen Wang, "Internet of Things for Enterprise Systems of Modern Manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1537–1546, 2014.
- [7] T. Sauter and M. Lobashov, "How to Access Factory Floor Information Using Internet Technologies and Gateways," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 699–712, 2011.
- [8] "Floods - National Disaster Management Authority" [Online]. Available: <http://www.ndma.gov.in/en/media-public-awareness/disaster/natural-disaster/floods.html>. Last Accessed on 28th August 2016.
- [9] D. Pietz, "Engineering the State: The Huai River and Reconstruction in Nationalist China 1927–1937," 2002.
- [10] "Top 10 Strategic Technology Trends | Gartner." [Online]. Available: <http://www.gartner.com/technology/research/top-10-technology-trends/>. Last Accessed on 30th June 2016.
- [11] C. Wang, Z. Bi, and L. Da Xu, "IoT and Cloud Computing in Automation of Assembly Modeling Systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1426–1434, 2014.
- [12] F. Tao, Y. Cheng, L. Da Xu, L. Zhang, and B. H. Li, "CCIoT-CMfg: Cloud Computing and Internet of Things-Based Cloud Manufacturing Service System," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1435–1442, 2014.
- [13] Z. X. Jiang, Y. L. Chen, J. J. Chen, and W. T. Wu, "An NFC-driven Home Automation Framework: An Integration of WSN, Social Networks and Cloud Computing," *Frontiers in Artificial Intelligence and Applications*, vol. 274, pp. 1489–1498, 2015.
- [14] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of Cloud Computing and Internet of Things: A Survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.
- [15] T. Shon, J. Cho, K. Han, and H. Choi, "Toward Advanced Mobile Cloud Computing for the Internet of Things: Current Issues and Future Direction," *Mobile Networks and Applications*, vol. 19, no. 3, pp. 404–413, 2014.
- [16] T. Renner, A. Kliem, and O. Kao, "The Device Cloud - Applying Cloud Computing Concepts to the Internet of Things," *IEEE 11th International Conference on and Autonomic and Trusted Computing in Ubiquitous Intelligence and Computing*, pp. 396–401, 2015.
- [17] W. Dai, H. Jin, D. Zou, S. Xu, W. Zheng, L. Shi, and L. T. Yang, "TEE: A Virtual DRTM Based Execution Environment for Secure Cloud-end Computing," *Future Generation Computer Systems*, vol. 49, pp. 47–57, 2015.
- [18] S. Shaoling, Z. Jing, C. Moliang, R. Hui, C. Yi, and F. Xiaodong, "Network Energy Consumption Analysis and Dormancy Mechanism Based on Ant Colony Algorithm in Cloud Computing Environment for IOT Service and Real-time Embedded Industrial Control System," *27th Chinese Control and Decision Conference*, pp. 5588–5592, 2015.
- [19] M. Villari, A. Celesti, M. Fazio, and A. Puliafito, "A Secure Self-Identification Mechanism for Enabling IoT Devices to Join Cloud Computing," *Internet of Things. IoT Infrastructures*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 151, pp. 306–311, 2015.
- [20] A. Amato, B. Di Martino, and S. Venticinque, "Big Data Processing for Pervasive Environment in Cloud Computing," *International Conference on Intelligent Networking and Collaborative Systems*, pp. 598–603, 2015.
- [21] J. A. Galache, T. Yonezawa, L. Gurgen, D. Pavia, M. Grella, and H. Maeomichi, "ClouT: Leveraging Cloud Computing Techniques for Improving Management of Massive IoT Data," *7th IEEE International Conference on Service-Oriented Computing and Applications*, pp. 324–327, 2014.

- [22] L. Jiang, L. Da Xu, H. Cai, Z. Jiang, F. Bu, and B. Xu, "An IoT-Oriented Data Storage Framework in Cloud Computing Platform," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1443–1451, 2014.
- [23] F. Hossain, E. N. Anagnostou, and T. Dinku, "Sensitivity Analyses of Satellite Rainfall Retrieval and Sampling Error on Flood Prediction Uncertainty," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 1, pp. 130–139, 2004.
- [24] R. Merz and G. Blöschl, "Flood Frequency Regionalisation - Spatial Proximity vs. Catchment Attributes," *Journal of Hydrology*, vol. 302, no. 1–4, pp. 283–306, 2005.
- [25] P. C. Nayak, K. P. Sudheer, and K. S. Ramasastri, "Fuzzy Computing Based Rainfall-runoff Model for Real Time Flood Forecasting," *Hydrological Processes*, vol. 19, no. 4, pp. 955–968, 2005.
- [26] P. Gourbesville, J. Batica, J. Y. Tigli, S. Lavirotte, G. Rey, and D. K. Raju, "Flood Warning Systems and Ubiquitous Computing," *La Houille Blanche*, no. 6, pp. 11–16, 2013.
- [27] P. K. T. Nguyen, L. H. C. Chua, and L. H. Son, "Flood Forecasting in Large Rivers with Data-driven Models," *Natural Hazards*, vol. 71, no. 1, pp. 767–784, 2013.
- [28] V. Quiroga and I. Popescu, "Cloud and Cluster Computing in Uncertainty Analysis of Integrated Flood Models," *Journal of Hydroinformatics*, vol. 15, no. 1, pp. 55–70, 2013.
- [29] D. S. Shih, C. H. Chen, and G. T. Yeh, "Improving Our Understanding of Flood Forecasting using Earlier Hydro-meteorological Intelligence," *Journal of Hydrology*, vol. 512, pp. 470–481, 2014.
- [30] A. K. Lohani, N. K. Goel, and K. K. S. Bhatia, "Improving Real Time Flood Forecasting using Fuzzy Inference System," *Journal of Hydrology*, vol. 509, pp. 25–41, 2014.
- [31] W. He and L. Da Xu, "Integration of Distributed Enterprise Applications: A Survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 35–42, 2014.
- [32] S. Fang, Y. Zhu, L. Xu, J. Zhang, P. Zhou, K. Luo, and J. Yang, "An Integrated System for Land Resources Supervision Based on the IoT and Cloud Computing," *Enterprise Information Systems*, pp. 1–17, 2015.
- [33] India Water Portal, "<http://www.indiawaterportal.org/articles/district-wise-monthly-rainfall-data-list-rain-stations-india-metrological-department>" Last Accessed on 28th August 2016.
- [34] "National Oceanic and Atmospheric Administration | U.S. Department of Commerce." [Online]. Available: <http://www.noaa.gov/>. Last Accessed on 28th August 2016.
- [35] "UttarakhandStat Provides Comprehensive Information on All The Major Sectors of The State Level Consolidated Research Statistical Data." [Online]. Available: <http://www.uttarakhandstat.com/default.aspx>. Last Accessed on 28th August 2016.
- [36] "FMIS." [Online]. Available: <http://fmis.bih.nic.in/Riverbasin.html>. Last Accessed on 31st August 2016.
- [37] V., Chang. "The business intelligence as a service in the cloud". *Future Generation Computer Systems*, 37, 512-534, 2014.
- [38] V., Chang (2017). "Towards data analysis for weather cloud computing". *Knowledge-Based Systems*, 127, 29-45.
- [39] V., Chang, & M., Ramachandran, M. "Towards achieving data security with the cloud computing adoption framework". *IEEE Transactions on Services Computing*, 9(1), 138-151.
- [40] V., Chang. (2015). "Towards a Big Data system disaster recovery in a Private Cloud". *Ad Hoc Networks*, 35, 65-82.