S-image (Situation Image) a new technique for data aggregation in cloud server for IoT based Smart City

SK Alamgir Hossain[1], Md. Anisur Rahman[2], and M. Anwar Hossain[3]

[1] Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh
alamgir@cseku.ac.bd, https://alamgirhossain.com
[2] Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh
[3] Department of Software Engineering,CCIS, King Saud University, KSA

Abstract. The diversity and sheer expanding in the number of Internet of Things (IoT) devices in a smart city context has raised substantial problems about storage and processing. Different sensors use different data formats. A situation is formed by combining data obtained from different sensors. This combination process needs a unified representation of sensor data. However, processing this massive amount of data and combining it to represent appropriate situations is a difficult task. To overcome this challenge, a data aggregation mechanism that is both efficient and light-weight is required. In this research, we developed a new data aggregation technique in cloud servers, where the processed data is transformed into a two-dimensional image-like spatial representation called Situation Image (S-image). We also developed a prototype that realizes the aforementioned aggregation model. In our experiment, multiple data mining techniques were chosen for processing various datasets in order to meet a variety of application goals. The experimental findings proved the viability of our data aggregation method.

Keywords: Smart City, Internet of Things, Situation Image, Data Aggregation

1 Introduction

With the advancement of technology in society, new opportunities have evolved that have the potential to simplify our everyday lives and deliver more efficient services or industrial processes. The Smart City concept operates in a demanding metropolitan context where infrastructure, human behavior, technology, societal factors, and the economy are all complicated systems [3] [18]. A smart city allows for the intelligent management of technologies such as transportation, health, education, energy, residences, and environment [25] [15]. A smart city's construction is made up mostly of information and communication technologies (ICT). It assists the globe in developing, disseminating, and promoting sustainable development techniques in order to tackle the problems of rising urbanization [13].

The growing Internet of Things drives and enables a city's smartness technologically [14]. The concept of connecting everything is known as the internet of things, where everything is equipped with electronics, software, and sensors to the internet in order to gather and share data [27] [26]. The primary structure of IoT is based on data being detected by sensors or actuators and then transferred to the server via the gateway [12] [11]. Interaction in the IoT takes several forms, including device-to-device, thing-to-device, and thing-to-thing. This engagement leads us into an unparallel era of Big Data explosion. It has recently seen a drastic transition in data sources from mega-scale cloud data-centers to more common end devices, such as mobile devices and Internet of Things (IoT) devices. Big data was traditionally created and stored in massive data centers. The tendency is now reversing due to the rise of mobile computing and the Internet of Things. Clearly, the edge ecosystem will provide numerous innovative application possibilities for Smart Cities and drive the further growth of IoT by providing massive amounts of data.

In a smart city environment, the diversity and sheer rise in the quantity of IoT devices has created significant storage and processing issues. A situation [8] is a condition at a moment in a particular location. In other words, a situation is a set of things that happen and the conditions that exist at a particular time and place. In the IoT, a situation is created by mixing data from many sensors. This approach requires a uniform representation of sensor data. However, it is a challenging effort to interpret such a large amount of data and combine it to reflect acceptable scenarios. A data aggregation approach that is both efficient and light-weight is required to overcome this difficulty. In this paper, we establish a novel data aggregation approach on cloud servers, in which the processed data is transformed into a two-dimensional image-like view called the Situation Image. We also developed a prototype that implements the aggregation paradigm stated before. According to our findings, the proposed framework is feasible.

In this paper, we make three contributions. First, we presented an approach for a data aggregation mechanism using an innovative technique called Situation Image. We then go over the proposed system's architectural design. Finally, we assessed our system using several metrics and presented the findings of our developed prototype.

The rest of this work is arranged in the following manner. We go through various pertinent research findings on Smart Cities, the IoT, and Data Aggregation approaches in Section 2. The suggested Situation Image creation technique is depicted in Section 3. Section 4 discusses the implementation concerns, development hurdles, and the findings of our investigation. In Section 5, we give a conclusion and address the future study directions.

2 Related Works

We explored existing literature that focused on data aggregation issues and techniques in the IoT. This section goes through the basics of these approaches as well as the principles and architecture that govern them.

IoT applications demand the transmission of massive volumes of acceptable data from one network node to another [6]. Because sensor nodes have limited energy, it is wasteful for all nodes to transfer all collected data to the sink node instantly. Data from nearby sensor nodes is typically correlated and highly redundant. Furthermore, in large sensor networks, the volume of data collected is usually too high for the sink node to process. As a result, a technique for merging this redundant and related data into useful, high-quality information is necessary at the intermediate nodes. Data aggregation of this type has been demonstrated to reduce power consumption, bandwidth utilization, load balancing, network lifetime, and data accuracy [16]. It requires gathering data from several sensor nodes at intermediate nodes and sending it to the end node. A smart data aggregation strategy may save energy usage and increase network traffic density while also extending network life and enhancing data accuracy. Meaningful information extraction is an important step for data aggregation. Researchers have examined hidden patterns in traditional data in a variety of application systems, and data mining plays a key part in identifying this relevant information [9] [17]. However, IoT data mining studies that incorporate the diversity and variety of IoT characteristics are still scarce [21] [10].

For smart cities, several IoT applications are being developed. Smart parking, garbage management, and traffic congestion management are a few examples of those IoT applications. These applications make use of billions of sensors, resulting in a vast volume of data classified as "Big Data." There has to be an appropriate framework through which the needed sensors can be readily discovered. To make the most use of this data, IoT/Smart-city frameworks have been proposed. A detail study on this area can be found in [2]. Quality of services (QoS) in data aggregation is also an important indicator. Because of its significance, Shamim et. al. [28] provide detailed data aggregation processes on the IoT, which demonstrates the essential hurdles in context design concerns. They provided the data aggregation process in detail, but a complete picture of how the aggregation technique will work in an IoT-based smart city was missing. Among the several studies that make use of the combination of IoT and cloud, the authors of [1] focused on combining cloud computing techniques to create new location-aware services while reducing latency. However, the combination of aggregation in the IoT is missing where, little emphasis has been made by researchers on data aggregation for the internet of things, which might benefit from knowledge or dataset or feature characteristics aggregation. This prompted us to perform the current study by addressing the issue of real-time data aggregation of diverse IoT datasets.

3 Proposed Aggregation Technique

3.1 Overview of the System

The main idea of our proposed technique is the unified representation of heterogeneous sensor data by an image like 2-dimentional plane (*S-image*). In our technique, for every S-image representation, the system needs to select an appropriate data mining algorithm for classification. The suggested system architecture for the dynamic *S-image* based aggregation model is depicted in Figure 2.

The diagram includes a number of functional components: *dataPreprocessing*, *unifiedRepresentation*, *goalRepresentation*, *algorithmSelection*, and *aggregateData* which are described in the sections that follow. For the proposed system to work, a *S-image* creation technique has been developed (Algorithm 1) that uses G (Goal), D (Date), and A (Algorithm) as input for discovering the best algorithm for the data preparation task. Before the details of our proposed system, we are going to explain the situation image structure. A Situation image, or S-image, is an image like 2-dimensional spatial representation. It is generated from the unified representation of data.

There are primarily two factors for each S-image: *element* and *situation image set*. Every situation image component is a collection of key and value pairs, where the key is the name of the factor or characteristics (e.g., temperature, moisture, noise, etc.) and the value is the data set's cumulative value for the key. In a practical setting, in addition to the $key - values$ pairs, other data such as location, data size, and so on should also be stored. In a real-world setting, a continuous Situation Image will be created throughout the time. The situation image set, often known as the S-image set, is a constantly created finite collection of situation images. The construction of an S-image set is shown in Figure 1.

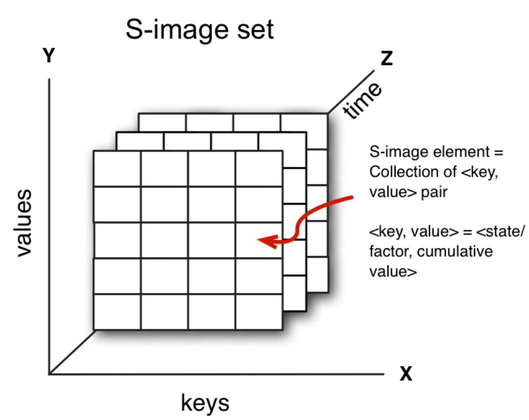

Fig. 1: S-image set.

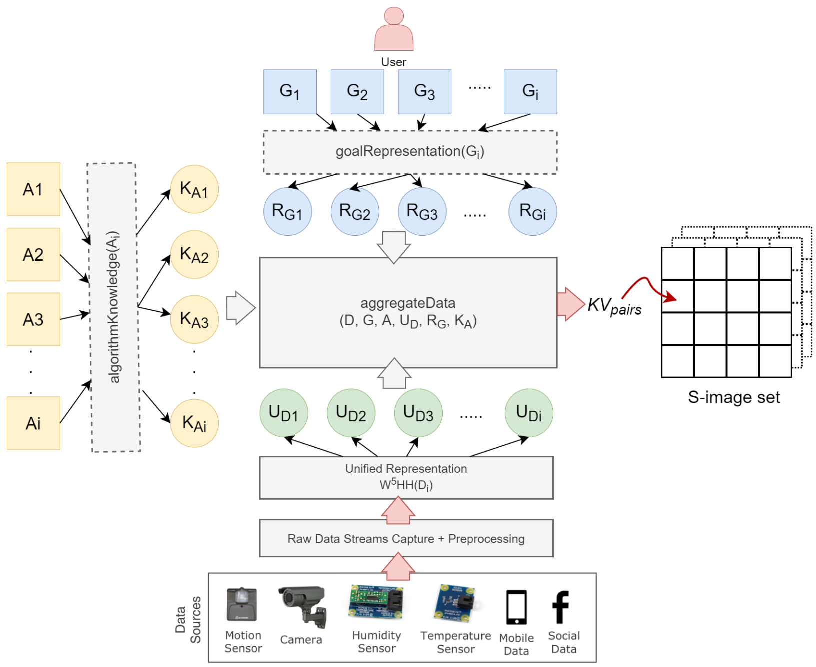

Fig. 2: Aggregation framework.

3.2 Data Preprocessing

As the initial stage of data aggregation, the raw IoT data is captured and processed. If the unprocessed data is delivered straight to the server for processing, the processing time will be enormous. As a result, several on-line and off-line mechanisms such as query modeling and data compression are used to decrease the cost of energy usage. In our suggested technique, data is labeled with its frequency of generation and location. We chose the ADMM method [5], which adequately meets our application of dividing and processing a huge collection of data into different processing blocks.

3.3 Knowledge Retrieval

This section discusses the steps for retrieving knowledge from the goals, datasets, and algorithms. Also, describes the process of creating S-images.

Unified Representation After the preprocessing, the IoT data needs to be represented in a uniform format before being sent to the model. This is essential because the results of processing IoT data from several sensors may overlap and

Algorithm 1: *S-image construction*

```
/* Based on the information gathered from the sensors, this
   algorithm creates an s-image. Individual sensors may be used, or
   they may be joined together to achieve a common goal.          */
```

Input: Collection of Mining Algorithms A, Goals G, and Datasets D
Output: A 2-D situation image
/* U_D is the Unified data, R_G is the goal represenation, K_A is the algorithm knowledge */
Initilize U_D, R_G and K_A; **while** *data available in edge server* **do**
  $D_i \leftarrow dataPreprocess(D_i)$;
foreach $D_i \in D$ **do**
  $U_{Di} \leftarrow W^5HH(D_i)$;
  /* W^5HH is the unifiedRepresentation functional unit */
  $U_D \leftarrow U_D \cap U_{Di}$;
foreach $G_i \in G$ **do**
  $R_{Gi} \leftarrow goalRepresentation(G_i)$;
  $R_G \leftarrow R_G \cap R_{Gi}$;
foreach $A_i \in A$ **do**
  $K_{Ai} \leftarrow algorithmKnowledge(A_i)$;
  $K_A \leftarrow K_A \cap K_{Ai}$;
$KV_{pairs} \leftarrow aggregateData(D, G, A, U_D, R_G, K_A)$ /* Where KV_{pairs} is as set of key value pair */
return $S_{Image}(KV_{pairs})$

provide a composite output. For example, to process the weather data for a certain place, we must aggregate the output of several sensors that produce data in a variety of patterns, formats, and other characteristics. So, before proceeding to the next stage, we need to give the data to a framework that converts it to a unified representation. We utilized the idea W^5HH [4] for the unified representation, which is actually a bunch of queries designed to extract acceptable answers from a problem. This method is widely used in software projects and process control.

Normally, in a smart city, different sectors like traffic control and healthcare management generate diverse forms of data that have many kinds of attributes [7]. So, mining information in a specific area becomes complex. This is complex not only because of the diversity of the data, but also because of variations in the data's centroid, skewness, probability distributions [9] etc. For our proposed model, Algorithm 2 explains how to create a unified representation from a dataset.

- *What*: This is the subject of this attribute. This is sensor-specific information such as wind direction, temperature, and air speed.
- *When*: This represents when data is collected. This is datetime data, such as today, yesterday, 10 a.m., 5/5/2021, and so on.
- *Where*: This refers to where the information is coming from. Office, Indoor, kitchen 1, and so on.

Algorithm 2: $W^5HH(D_i)$

```
/* This algorithm generates result (unified format such as
   'what-when-where') based on a series of function.          */
```

Input: Datasets D_i
Output: U_{Di} unified represented data
$U_1 \leftarrow getWhatData(D_i)$;
$U_2 \leftarrow getWhenDataTaken(D_i)$;
$U_3 \leftarrow getWhereDataComing(D_i)$;
$U_4 \leftarrow getWhoGenerating(D_i)$;
$U_5 \leftarrow getWhatCondition(D_i)$;
$U_6 \leftarrow getHowLong(D_i)$;
$U_7 \leftarrow getHowMuch(D_i)$;
$U_{Di} \leftarrow add(U_1, U_2, U_3, ..., U_7)$; `/* add() combine the input data */`
return U_{Di}

– *Who*: This attribute contains information about the data's creator. Sensor id number may be good identification number. Some example of this type of data are tempS3, humidityS4, 3,5 etc.
– *Why*: This property represents why or in what condition the data is generated. This symantec data depends on sensor sate or activity. For example, Motion, Acceleration, Cold weather etc.
– *How Long*: This property carries how long the data is being generated like one hour, five minutes etc.
– *How Much*: This is a multi-valued feature that is dependent on meta data. It reflects the amount of resources required. For example, data length, data type etc.

It is worth emphasizing that these are a few examples of possible knowledge attributes; the framework isn't restricted only to these. Other systems that use the suggested framework to add knowledge to a dataset can add additional dynamic and distinctive features to it. Finally, the dataset represented in a unified way is compared to goal knowledge to determine its relevance.

Goal Representation The key issue of IoT is to choose appropriate data mining techniques for a variety of purposes in a single or several data domains. In the health sector, assessing health information for the sickness prediction system and providing emergency assistance to patients in smart patient rooms is a sample of applications. Goals may also change depending on the dataset. Goals derived from healthcare data, for example, differ from those derived from transportation data. We examine a set of goals G for the proposed model, and a specific dataset D_i may be required to meet one or more goals. Goal knowledge, as contrasted to data knowledge, is nearly generic and predetermined. That's why goal is described with a variety of representable factors R_{Gi} of a specific goal G_i, as shown below.

Algorithm 3: $goalRepresentation(G_i)$

```
/* This algorithm generates goal knowledge based on a given goal.
   */
```

Input: A Goal G_i
Output: A list of goal knowledge
$R_1 \leftarrow goalName(G_i)$;
$R_2 \leftarrow goalDomainType(G_i)$;
$R_3 \leftarrow goalOutputType(G_i)$;
....
$R_i \leftarrow goalContext(G_i)$;
return $add(R_1, R_2, R_3, ..., R_i)$; /* *add* function return the cumulative representation of the factors */

- *goalName*: Indicates understanding about the process's goal.
- *goalDomainType*: Information related to target dataset's data type.
- *goalOutputType*: Specified by the goal, which reflects model knowledge.
- *goalContext*: The goal's context, which represents information about a dataset's target data domain.
- *goalCoverage*: Knowledge of the location of target data that will fulfill an objective.

The values of the aforementioned attributes collected from the user interface are used to instantiate a specific objective. Using the methods in Algorithm 3, the goals are evaluated and associated knowledge is retrieved. $R_{Gi} = \{R_1$=prediction, R_2=nominal, R_3=classification, R_4=diabetes, R_5=Paris, ...$\}$ is an example of goal representation. However, a goal's qualities may be modified to enable for the generation of more information about a goal. Finally the best matching algorithm will be selected after the goal representation.

Algorithm Knowledge Extraction Researchers [23] have praised a number of data mining algorithms, including Hierarchical Clustering (HC), Decision Tree (DT), K-Means, Featured Clustering (FC), Support Vector Machine (SVM), Random Forest (RF), Naive Bayes (NB), Multilayer Perceptron (MLP), Hidden Markov Model (HMM), and K-th Nearest Neighbor (KNN). It will differ when supervised and unsupervised/clustering algorithms will be necessary. It depends completely on the data domain and the goal. For example, forecasting parking space availability may necessitate supervised classification techniques, whereas clustering may be required to discover parking locations based on customer behavior. Algorithms of the same kind, on the other hand, have varied requirements, which affect the learning model's performance dramatically. SVM, is popular for dealing with data points that are non-linear, although it has a larger computing cost than the DT or KNN classifiers. DT, on the other hand, is sensitive to all numeric datasets while being an HC approach. As a result, expertise in data mining algorithms is essential for IoT data mining automation.

Algorithm 4: $algorithmKnowledge(K_i)$

```
/* This algorithm generates algorithm knowledge based on a given
   algorithm.                                                    */
```

Input: An algorithm A_i
Output: A list of algorithm knowledge
$K_1 \leftarrow getDataType(A_i)$;
$K_2 \leftarrow getProcess(A_i)$;
$K_3 \leftarrow getOutputType(A_i)$;
....
$K_i \leftarrow getSensitivity(A_i)$;
return $add(K_1, K_2, K_3, ..., K_i)$;

Algorithm 5: $aggregateData(D, G, A, U_D, R_G, K_A)$

Input: Datasets(D), Goals(G), Algorithms(A), Unified data (U_D), goal represenation(R_G), algorithm knowledge(K_A)
Output: A set of KV_{pair} which is the key value pair of S_{Image}
Initilize *maxSimilarityD*, *maxSimilarityA*, *maxCombineSimilarity* to a negative value;
while *there is a* G_i *in* G **do**
  foreach $D_i \in D$ **do**
    $\eta \leftarrow similarity(R_{Gi}, U_{Di})$;
    if $\eta >= maxSimilarityD$ **then**
      $e^D \leftarrow add(Di)$; $maxSimilarityD \leftarrow \eta$;
  foreach $A_i \in A$ **do**
    $\eta \leftarrow similarity(R_{Gi}, K_{Ai})$;
    if $\eta >= maxSimilarityA$ **then**
      $e^A \leftarrow add(Ai)$; $maxSimilarityA \leftarrow \eta$;
  while *there is a* e^{Di} *in* e^D **do**
    while *there is a* e^{Ai} *in* e^A **do**
      $\eta \leftarrow similarity(U_{e^{Di}}, K_{e^{Ai}})$;
      if $\eta >= maxCombineSimilarity$ **then**
        $e^{DA} \leftarrow add(Di)$; $maxCombineSimilarity \leftarrow \eta$;
    $KV_{pairs} \leftarrow KV_{pairs} \cup (e^{DA} \times G_i)$
return KV_{pairs}

To meet the dynamic selection of a suitable data mining technique for a particular dataset or purpose, the suggested model incorporates both supervised and unsupervised data mining algorithms. Let's say A is a collection of algorithms, and where, $A = A_1, A_2, A_3, ..., A_i$ represents a number of data mining algorithms (supervised or unsupervised). Algorithm 4 is used to extract the knowledge characteristics K_{Ai} of any such algorithm A_i.

Data Aggregation After the knowledge extraction steps, Algorithm 5 is used to aggregate the data. The main task of this algorithm is to aggregate the data based on the input. The inputs will be the goal representation, algorithm knowledge, and unified data. We choose the technique indicated in [23] after analyzing different similarity calculation approaches. The while loop in Algorithm 5 is used to iterate the goals. The first for loop is used to calculate the greatest similarity score $maxSimlarityD$ by comparing knowledge about a specific objective to all datasets in D and finally the selected data sets is stored in e^D. The second for loop is used to calculate the greatest similarity score $maxSimilarityA$ and it is stored in e^A.

In the next step, a combined similarity between e^D and e^A is calculated and finally KV_{pairs} is generated. This KV_{pair} is the building block of our S-image. It should be noted here that the dimension of the situation image depends on the user application requirements and goal size. If the number of goal is large then Algorithm 5 will generate more KV_{pairs} so, to map the pair higher dimension of situation image will be required. After selecting the appropriate algorithm based on the given goal, Algorithm 5 will generate the key-value pair (KV_{pairs}) where each key represents the goal and the value represents the result of the dataset based on the selected algorithm and the given goal. Finally Algorithm 1 execute the $S_{Image}(KV_{pairs})$ which is actually 2-d spacial mapping between the key-value pairs to the 2-d plane.

4 Implementation and Result

In order to justify our technique, we performed several experiments. Our experiments focus mainly on two aspects: (1) an evaluation of the proposed framework's performance with various targets using some off-line data sources; and (2) testing the method in a real-time situation.

4.1 Setup for Experiments

We investigated a variety of application goals, datasets, and data mining technologies in order to execute the experiment.

To address the variety of data sources in the context of the Internet of Things, the datasets were gathered from various fields. There are over 150 different techniques to build supervised algorithms [19], as well as several data mining algorithms. Existing implementations of data mining methods using the Python Libraries [22] are evaluated here. It should be noted that before sending the data for knowledge extraction, we removed any *null* and missing values. The first experiment focuses on some general goals, whereas the second experiment focuses on a real-time case.

Analysis of the Proposed Framework's Performance Using Offline Data Sources In this experiment, the following goals are considered.

- G1: Community recommendation based on weather condition.
- G2: Favorite route prediction.
- G3: Traffic accident prediction.
- G4: Prediction of parking availability.

As the primary data sources, the City Pulse [24] and City of Chicago [20] smart city data sources were utilized. Although different IoT datasets are available now but we selected this two data set because in our experiment we need csv format data as well as API hook so that our process can call realtime. We found these nice features in our selected datasets. We also used the Amazon EC2 service (3.2 GHz Core i7, 16GB memory, Windows Server os). We ran our tests in two modes (offline and real-time), one in batch mode and the other in real mode. We created a specific component in our system to accept data from the data set and transmit it to our model in offline mode. Table 1 demonstrates the CityPlus and City of Chicago data source details.

Table 1: CityPlus and and City of Chicago data source details.

Dataset ID	**Name**	**Instances**
D1	Road Traffic Data 1	7306
D2	Road Traffic Data 2	83721
D3	Road Traffic Data 3	14000
D4	Road Traffic Data 4	9397
D5	Pollution Data	11569
D6	Weather Data	16369
D7	Parking Date 1	55285
D8	Parking Data 2	53267

In our framework, Algorithm 5 will generate three type of informations, 1) e^D = datasets 2) e^A = mining algorithms 3) e^{DA} = datasets and mining algorithms. Table 2 shows the outcomes from these sets. It is evident that several datasets and mining techniques may be chosen for each purpose, as indicated in the last column of Table 2. As, G_1 is a clustering issue, and many clustering techniques have been chosen as candidates in e^D. The last column of Table 2 shows the dataset and data mining technique that were chosen for each purpose. Any of the algorithms might have been chosen for the task if the procedure had been done manually. The suggested framework, on the other hand, delivers the optimal

Table 2: Selection by aggregateData() Algorithm.

Goals	**Output** e^D	**Output** e^A	**Output** e^{DA}
G1	D5, D6	HC, FC, K-Means	D5, HC
G2	D1, D2, D3, D4	NB, DT, RF	D2, DT
G3	D1, D2, D3, D4	NB, DT, RF	D1, DT
G4	D7, D8	NB, DT, RF	D8, RF

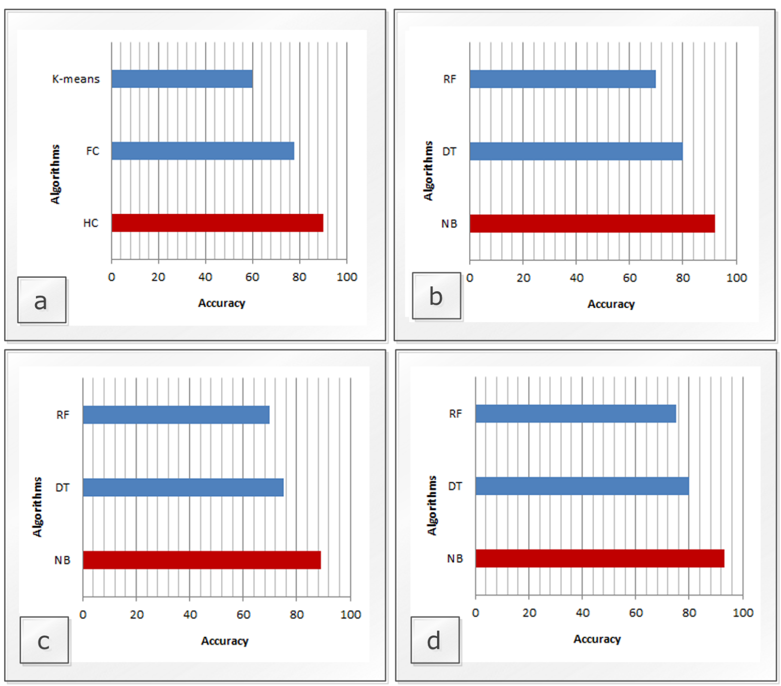

Fig. 3: For each goal the performance comparison of mining algorithms in e^A, with the red color horizontal bar representing the suggested method by the proposed method. Figures (a-d) provide a comparison of G_1 to G_4 accuracy.

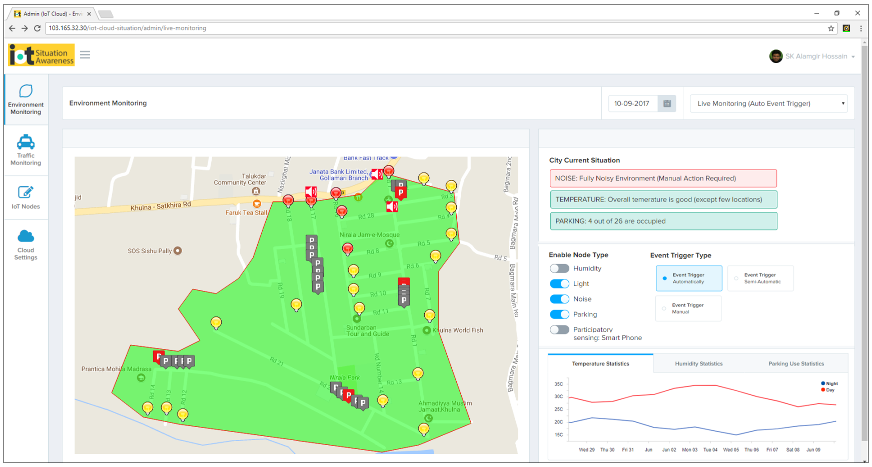

Fig. 4: Web application prototype for real-time situation monitoring.

choice among the potential algorithms. The performance of our method is shown in Figure 3 which is around 89%-93%.

We utilized our test bed site for the real case (details of our test bed location may be found in our previous work [14]). The size of the region is around $0.45km^2$. A map view of the deployment is shown in Fig. 4. Distinct pins on the map indicate different nodes in this map view. We put nodes to measure light levels, ambient noise levels, ambient temperature, and traffic presence, particularly automobile presence. Following the selection of the test bed location, our key difficulty was to install the sensor nodes in acceptable locations while also providing continuous power. It is more practical to use solar energy to charge batteries; we used a tiny solar panel to charge the node batteries. The primary issue with solar power is that the battery power may run out throughout the night. Therefore, we positioned our nodes on roadside lampposts. Finally, we employed a metal enclosure to shield the sensor node from the elements and human intervention. In this case, the mobile phone serves as an edge processing unit. If a certain event takes place in the city, the system will notify city authorities who have subscribed to that service. Users can also report such occurrences, which will be instantly spread to other qualified subscribers. To assess the performance of the scenario detection approach, we monitored the environment for four weeks and computed the number of events that happened vs the number of events detected by our system. Our prototype was able to recognize 176 occurrences out of about 181 distinct events such as high temperature, low humidity, and a hot sunny day. Due to connection delays and server load, our algorithm may sometimes provide false information. It should be mentioned that the entire test was conducted throughout the summer months, with no rain or other natural calamities.

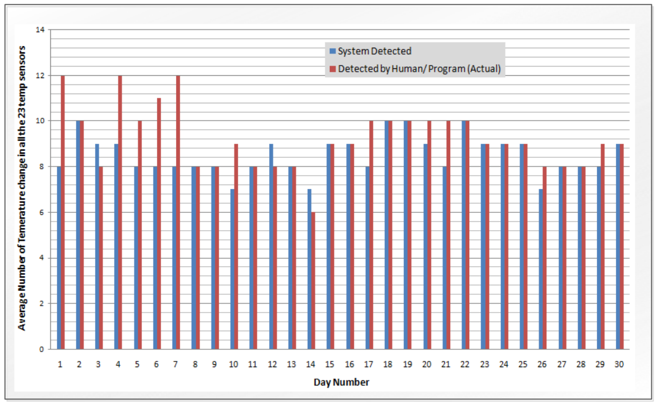

Fig. 5: Performance in detecting situations.

Figure 5 depicts the performance of situation detection over temperature sensors in our prototype test case. During the monitored days, there were a total of 5506 temperature increases or decreases. We choose 25°C as the threshold temperature because we conducted our test on sunny days. To determine the actual temperature rise or decrease from the threshold, a second sensor is installed in the same area as the nodes. Based on this study, we discovered that our cloud system gave results that were almost equal to the original data. It should be noted that our system occasionally displays a larger increase or decrease than the real value since the temperature sensor may function poorly during a continuous run or in intense sunlight.

According to the results of the preceding experiment, the suggested framework provides a comprehensive data aggregation technique for IoT. It demonstrates that in a dynamically changing IoT environment, the suggested framework is capable of selecting an appropriate mining algorithm for the target tasks. The findings indicate the framework's adaptability to a variety of datasets, as well as its applicability for a real-time situation.

5 Conclusion

Situation Awareness (SA) guarantees that the correct information reaches the appropriate people at the right time, allowing for faster and more effective communication, particularly in an emergency. When given a huge amount of IoT data, it is a complex task to select and process appropriate data and present it in an appropriate way. In situation awareness, it is essential to process the low-level IoT data in a manner so that all the complexity is hidden from the decision-makers. Only abstract information should be displayed to the user interfaces. In this research, we introduced a novel data aggregation approach, in which processed data is converted to an image-like two-dimensional representation. According to the experiment, multiple data mining techniques were chosen for processing various datasets in order to meet a variety of application goals. The experimental findings proved the viability of our data aggregation method. Although we examined our system in a test bed environment, we think that we need to test the approach in a bigger scenario like monitoring a whole city. So in the future, we will further examine the system in more realistic and larger scenarios. Still, we believe that the suggested paradigm would lead to a new era of IoT-based smart city research.

References

1. Soroush Abbasian Dehkordi, Kamran Farajzadeh, Javad Rezazadeh, Reza Farahbakhsh, Kumbesan Sandrasegaran, and Masih Abbasian Dehkordi. A survey on data aggregation techniques in iot sensor networks. *Wireless Networks*, 26(2):1243–1263, 2020.
2. Abrar Alkhamisi, Mohamed Saleem Haja Nazmudeen, and Seyed M Buhari. A cross-layer framework for sensor data aggregation for iot applications in smart cities. In *2016 IEEE International Smart Cities Conference (ISC2)*, pages 1–6. IEEE, 2016.
3. Ari-Veikko Anttiroiko, Pekka Valkama, and Stephen J Bailey. Smart cities in the new service economy: building platforms for smart services. *AI & society*, 29(3):323–334, 2014.
4. Barry Boehm. Anchoring the software process. *IEEE software*, 13(4):73–82, 1996.
5. Stephen Boyd. Alternating direction method of multipliers. In *Talk at NIPS Workshop on Optimization and Machine Learning*, 2011.
6. Rajni Chauhan and Vrinda Gupta. Energy efficient sleep scheduled clustering & spanning tree based data aggregation in wireless sensor network. In *2012 1st International Conference on Recent Advances in Information Technology (RAIT)*, pages 536–541. IEEE, 2012.
7. Laizhong Cui, Shu Yang, Fei Chen, Zhong Ming, Nan Lu, and Jing Qin. A survey on application of machine learning for internet of things. *International Journal of Machine Learning and Cybernetics*, 9(8):1399–1417, 2018.
8. Mica R Endsley. Design and evaluation for situation awareness enhancement. In *Proceedings of the Human Factors Society annual meeting*, volume 32, pages 97–101. SAGE Publications Sage CA: Los Angeles, CA, 1988.
9. Mohamed Medhat Gaber, Adel Aneiba, Shadi Basurra, Oliver Batty, Ahmed M Elmisery, Yevgeniya Kovalchuk, and Muhammad Habib Ur Rehman. Internet of things and data mining: From applications to techniques and systems. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3):e1292, 2019.
10. Wensheng Gan, Jerry Chun-Wei Lin, Han-Chieh Chao, Athanasios V Vasilakos, and S Yu Philip. Utility-driven data analytics on uncertain data. *IEEE Systems Journal*, 14(3):4442–4453, 2020.
11. Ananda Mohon Ghosh, Debashish Halder, and SK Alamgir Hossain. Remote health monitoring system through iot. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pages 921–926. IEEE, 2016.
12. Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

13. SK Alamgir Hossain, Md Anisur Rahman, and M Anwar Hossain. Detecting situations from heterogeneous internet of things data in smart city context. In *Science and Information Conference*, pages 1114–1127. Springer, 2018.
14. SK Alamgir Hossain, Md Anisur Rahman, and M Anwar Hossain. Edge computing framework for enabling situation awareness in iot based smart city. *Journal of Parallel and Distributed Computing*, 122:226–237, 2018.
15. Yin Jie, Ji Yong Pei, Li Jun, Guo Yun, and Xu Wei. Smart home system based on iot technologies. In *2013 International conference on computational and information sciences*, pages 1789–1791. IEEE, 2013.
16. B Karthikeyan, M Velumani, R Kumar, and Srinivasa Rao Inabathini. Analysis of data aggregation in wireless sensor network. In *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, pages 1435–1439. IEEE, 2015.
17. Li Li and Amir Ghasemi. Iot-enabled machine learning for an algorithmic spectrum decision process. *IEEE Internet of Things Journal*, 6(2):1911–1919, 2018.
18. Saraju P Mohanty, Uma Choppali, and Elias Kougianos. Everything you wanted to know about smart cities: The internet of things is the backbone. *IEEE Consumer Electronics Magazine*, 5(3):60–70, 2016.
19. Amril Nazir. Seamless automation and integration of machine learning capabilities for big data analytics. *Int. J. Distrib. Parallel Syst.*, 8(3):1–18, 2017.
20. City of Chicago. City of chicago open data. Technical report, https://data.cityofchicago.org/. Last accessed date: 26/05/2018.
21. Dijana Oreski, Stjepan Oreski, and Bozidar Klicek. Effects of dataset characteristics on the performance of feature selection techniques. *Applied Soft Computing*, 52:109–119, 2017.
22. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.
23. Nitin Pise and Parag Kulkarni. Algorithm selection for classification problems. In *2016 SAI Computing Conference (SAI)*, pages 203–211. IEEE, 2016.
24. Dan Puiu, Payam Barnaghi, Ralf Tönjes, Daniel Kümper, Muhammad Intizar Ali, Alessandra Mileo, Josiane Xavier Parreira, Marten Fischer, Sefki Kolozali, Nazli Farajidavar, et al. Citypulse: Large scale data analytics framework for smart cities. *IEEE Access*, 4:1086–1108, 2016.
25. Kehua Su, Jie Li, and Hongbo Fu. Smart city and the applications. In *2011 international conference on electronics, communications and control (ICECC)*, pages 1028–1031. IEEE, 2011.
26. Tamanna Tabassum, SK Hossain, Md Rahman, Mohammed F Alhamid, M Anwar Hossain, et al. An efficient key management technique for the internet of things. *Sensors*, 20(7):2049, 2020.
27. Jonathan Charity Talwana and Huang Jian Hua. Smart world of internet of things (iot) and its security concerns. In *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 240–245. IEEE, 2016.
28. Shamim Yousefi, Hadis Karimipour, and Farnaz Derakhshan. Data aggregation mechanisms on the internet of things: A systematic literature review. *Internet of Things*, 15:100427, 2021.