Chapter 1

INTRODUCTION

1.1 Overview

1.1.1 Greenhouse Monitoring, Parameters

Climate is the main factor which affects the growth of plants. Different vegetables and fruits have different seasons in which they grow and Different regions in the earth has extreme adverse climatic conditions or different environmental status which makes it uncomfortable to grow the crops with specific requirement. To protect the plants from the several environmental conditions like ultraviolet radiation, wind, hailstorm, insect and pest attacks, a method called Greenhouse were developed as solutions to overcome the problem which has made possible to grow all vegetables and fruits in all the seasons throughout the year.

Greenhouse are structures made of transparent materials where the user use to grow the plants with the required climatic conditions. There are numerous advantages of growing crops under greenhouse over the open field. The yield can be 10-12 times more, off-season production, protected from insects and disease from them, less requirement of water[1]. The greenhouse can maintain the favourable environmental conditions/parameters like temperature, humidity, solar radiation etc., for the respected crops. The parameters inside the greenhouse are sensed and measured using different sensors. The in-house environmental status can be monitored from remote areas making user comfortable when they are not at home and in case of Automated Greenhouse, it can take care of itself by monitoring the in-house environmental status according the specified conditions.

1.1.2 Internet of Things

Internet of Things is a technology which gives platform to inter-related computing devices, sensory devices, digital machines, mechanical devices etc., to produce and exchange the data via the Internet. The technology has increased the opportunities in innovation and there are huge devices connected in internet resulting in producing tremendous volume of data.

The applications of IoT is not limited to a particular stream and are distributed in nature. Many of the IoT applications requires real-time-processing. There are numerous applications of IoT which produces real time IoT data. These data are only limited to a specific domain or unused later for further purpose[2]. There are many streams of IoT where the data is being generated daily.

- **Traffic information:** The data produced by the connected vehicles, Traffic management system[3], Temporary and Permanent vehicular traffic counting, Average Speed of the vehicles, Type of Vehicle, location(longitude and latitude and altitude), Pedestrians information, Path history, etc.
- **Parking spaces:** Parking Time, Car Number which is vehicle registration plate, Amount to be Paid for parking which is different for different types of vehicles, Parking Slot, Occupancy Rate which involves monitoring of parking areas[4]**.**
- **Smart Homes:** Temperature, Relative Humidity, Motion Detection, Air Quality, Fire Detection, Luminosity, Irrigation System, Electrical Hazard[5].
- **Infrastructure Application:** Monitoring changes in the structural conditions of the infrastructures like Railway Tracks, Bridges, Wind Farms etc., [6].
- **Environmental Monitoring:** Monitoring the air quality, water quality, Atmospheric Conditions, Soil Conditions, Wildlife moments and their habitats. Earthquake & Tsunami early warning systems[7].
- **Metropolitan Scale Deployment:** Structural health monitoring, Waste Management like intelligent waste containers, Monitoring of Air quality, Noise Monitoring: To measure the noise production quantity, Traffic Monitoring: By installing GPS in vehicles, City Energy Consumption, Smart Lighting: Intensity of Street light, Smart Parking: Sensors on the road and intelligent displays to give the best path for motorists, Automation and salubrity of buildings, Water supply[8].
- **Medical and Healthcare:** Emergency Notification System, Remote health Monitoring like Heart rate monitoring, Blood pressure monitoring systems, Blood glucose monitoring systems, Smart watches etc.
- **Transportation:** Intra vehicular communication, smart traffic control, smart parking, Electronic toll collection, logistics, Safety and Road assistance
- **Security and Emergencies:** Perimeter access: Control the access to the restricted areas.

- **Radiation Levels:** To detect the leakage alerts in the nuclear power stations etc., Liquid presence: Detection of liquid in warehouses, Data centers, Sensitive Building grounds, etc. Explosive and Hazardous gases: Detection of gas leakage in homes, industrial environments, mines, chemical factories.
- **Logistics:** Item Location, Quality of shipment condition, Storage Incompatibility Detection, Fleet Tracking.
- **Retail:** Smart product management, intelligent shopping application, NFC Payment, Supply Chain Management.
- **Smart Agriculture and Animal Farming:** Greenhouses, Wine quality enhancing, Compost which deals with preventing fungus by controlling of temperature and humidity levels in alfalfa, hay, straw, etc. Animal Tracking, Toxic gas level.
- **Wearable Devices:** Smart watches which collects the data about the users, Gestures, Glasses
- **Smart Kitchen:** Grocery ordering devices Amazon Dash, Hiku, Genican. Flatware and utensils like Hapifork, vessyl. Inventory devices like Neo, Eggminder. Voice activated digital assistants like ubi, echo, ivee sleeks. Cooking devices like ChefSteps Joule Sous Vide. Range digital thermometer, crock-pot wemo smart wifi slow cooker, evercook, automated pressure cooker, mellow.

1.1.3 Data Abstraction & Semantic Representation

Data abstraction is the process of reducing multiple data to simplified version without altering the meaning. Whereas in Semantic representation, the data is represented in a meaningful form which can be understood by human.

The data abstraction methods can be obtained by Pre-processing Techniques like Signal Preprocessing in which methods like Low pass filter & High pass filter cuts the current signal with the cut-off frequency[9]. Mathematical/Statistical Preprocessing techniques uses the output data and performs the operation like mean, median, variance, standard deviation, correlation, Integration etc.

Dimensionality Reduction is applied to minimize the length and size of the data. Methods like Discrete Fourier Transformation transforms the signal from time domain to frequency domain. Piecewise Aggregation Approximation (PAA) takes equal sized frames as parameter, and calculates the averages of the frames in the original data[10]. Symbolic

Aggregate Approximation (SAX) transforms the time series data to series of letters which can be said as word. SAX is applied to the PAA output[11].

Semantic Reasoning methods like Reasoning in which data is passed through the rule engine. Based on the set of rule, operation on the data is carried out[12]. Clustering process to group the same type of objects[13]. Frequent Item set Mining which involves the method to find the values that co-occur frequently[14].

The data produced by these IoT devices remain unused for long and no information is gained from these data. In-order to get the information and use it for further future processing, abstraction is used. Depending on the frequency, the data is abstracted to string form from the numerical form.

1.1.4 Sensors considered for the project

In our work, Virtual Sensor programs are written which acts like Greenhouse sensors and produces the data. Sensors sense the change in the environmental parameters of the greenhouse. The ardunio reads the data from the input ports after being converted to a digital form by the ADC. With the help of respective sensors relays, Microcontroller performs needed action. P. S. Asolkar designed a greenhouse monitoring and controlling system for the crops Cucumber, Tomato, Brinjal, Papaya and Chilies [15]. They have used SY-HS220 humidity sensor, LM35 temperature sensor, Soil moisture sensor which is two copper probes, LDR (Light intensity Sensor), MQ5 gas sensor for CO_2 and actuators to sense the greenhouse parameters. K Rangan developed an embedded system approach to monitor greenhouse [16]. They have used LM35 temperature sensor for sensing temperature and the range of the temperature is 0° C to 150° C. Lijun Liu in their greenhouse used BH1750 sensor for measuring luminance used SHT11 sensor for measuring temperature and humidity [17]. Teemu Ahonen in their greenhouse monitoring system used SHT75 sensor for measuring temperature and humidity, TAOS TSL2561 luminosity sensor which converts light intensity to voltage and TGS4161 sensor to measure carbon dioxide [18]. Liang Ying used 18B20 temperature sensor to measure the temperature of soil [19]. Maro Tamaki used HF-LP02 sensor to measure the solar radiation inside the greenhouse [20].

Table 1.1: Sensors used in monitoring parameters of the greenhouse.

Name	Operating Range	Supply Voltage(V)	Interface	Accuracy
LM35	-55°C to +155°C	4 to 30 V	Analog	0.5° C
SHT75	-40°C to +125°C 0 - 100% RH	2.4 to 5.5V	Digital	±3° C ±1.8% RH
MQ5	200-10000 ppm	4.9 to 5.1V	Analog	-
BH1750	1 – 65535 lx	3.3 to 5 V	Digital	-
SHT11	0 - 100% RH -40 °C to +125°C	2.4 to 5.5 V	Digital	±3% ±0.4°C
TSL2561	0 – 40000 Lux	2.7 to 3.6 V	Digital	-
TGS4161	350 to 10000 ppm	5V	Analog	±20% ppm
18B20	-55°C to +125°C	3.0 to 5.5 V	Digital	±0.5° C
SEN 13322	0% - 100%	5V	Digital	±0.5%

1.2 Motivation

The scope to the IoT is growing day by day because of its feature Machine to Machine Communication. There are billions of devices connected to the internet and the numbers are increasing day by day. According to Gartner's Forecast IoT devices produce 2.5 quintillion bytes of data. The data production is going to increase as there will be 25 billion devices connected to the internet by 2020 [21]. The connected devices produce a huge amount of data which are either unused or limited to a specific domain. The applications of these are distributed in nature and there is necessity of real-time processing of data. The applications require novel methods capable of interpreting patterns and make accurate decisions to the current situations with minimum latency. Real time data are needed to be processed with a novel approach and represented in the form of high level knowledge. The term Real time data analysis notifies that the data has to be analyzed before it is stored [22].

Because of the large number of specific technologies, the valuable data or the information from the sensors remains unused or limited to specific application domains [2] (e.g. Parking spaces which consists of Time, Car Number, Parking fee, Slot, Occupancy Rate, traffic information, bus timetables, event calendars, waiting times at events, parking spaces, GIS databases etc.).

By the techniques like Data Mining and Knowledge Discovery in Database, the generated data can be converted to knowledge. The techniques provide solutions for finding the information hidden in the IoT data. The information can be used to improve the performance of the system and quality of service. The results show that data analysis algorithms that can be used to make IoT more intelligent, thus providing smarter services.

Efficient stream reasoning mechanisms are required to interpret the meaning of events in a context-aware fashion and share such meaning across applications. A sophisticated mechanism is required to make these data available to the end user in useful way.

1.3 Objective

The objective of work is to implement a novel method to represent the data from device point of view to user point of view by abstracting the data from low level to high level abstraction using data abstraction and topic modelling techniques.

To aggregate the data before storing by identifying useful data without losing any sensitive data by feature extraction and context awareness mechanism.

1.4 Scope

IoT has a concept Machine to Machine Communication and interactions with humans. This has enabled bringing all day-to-day things over network and can be accessed and controlled from any part of the world with the help of internet from the smart devices like mobile, laptop, wearable devices etc.

Due to the variable climatic conditions, lack of water and pests, majority are moving to the Greenhouse cultivation. In Spain and Italy, It has been estimated that around 25,000 ha and 18,500 ha area are under Greenhouse cultivation. The crops like Tomato, Strawberries, Watermelon, Cucumber, Capsicum, Beans, etc., are being cultivated.

As there are many advantages of Greenhouse cultivation over open field cultivation, majority population of the world are moving to Greenhouse cultivation. The data produced by from these will have no further application if it remains as raw data. Hence there should be a mechanism to extract meaningful interpretation from the raw data which can help in Production of more crops, Business Intelligence, Supply chain management etc.

1.5 Existing System

Methods like Adaptive Clustering for IoT data streams[13] clusters the data from different sources and use probability distribution techniques to plot the graph. A new category is found by the turning points of the graphs. Reasoning nodes[12] perform reasoning over the data and stores the data according to the rule sets provided. To find the abstraction of the raw data, methods like Piecewise aggregate approximation are used to split the input data into window segments[23] and then Symbolic aggregate approximation to determine the equivalent symbol for the coefficient. There are methods that extract the abstraction from the existing data which are stored in the database, like Cloud based centralized architecture[24] which stores the raw data from the sensors. The data will be huge and these architectures will cause delay for providing the required service and wastes many resources.

1.6 Proposed System

The proposed system takes the raw data from the sensor in real time. Based on the set of rules, the data is divided accordingly and given abstraction. The abstracted form of data is then cleansed by applying some of the cleansing methods. Then the data is passed to LDA algorithm, a topic modelling method is applied to automatically discover the topics in the data and to know the correlation among the parameters of the Greenhouse.

1.7 POs Attained

- Scholarship of Knowledge (PO 1)

  Through this project, by studying and implementing the abstraction for IoT data. I have gained in-depth knowledge in IoT.
- Critical Thinking and Problem Solving (PO 2, PO 3)

  The design of the project phase involved in identifying the steps involved in performing abstraction and applying LDA.
- Research Skill (PO 4)

Various types of problems and their existing solutions for abstraction and semantic representation were known.

- Usage of Modern Tools (PO 5)

  Have used python packages such as nltk (natural language toolkit), genism, lda, itertools and corpora.

- Project management (PO7)

  Project management is done through project planning and adhering to the plan, which has enhanced the project management skills.

- Communication (PO8)

  The oral and written communication was enhanced through project presentation and report writing.

Chapter 2

LITERATURE SURVEY

D D Nangare et al.[25] constructed three greenhouses to with different heights and shade to determine the effect on the crops. While growing the plants in the open field, several constraints were found. Pests like Helicoverpa armigera attacked the tomatoes which decreased the yield. Incidence of whitefly, aphid, mites and thrips were also noticed. And also due to the climatic conditions like extreme low/high temperature in the semi-arid region there was inferior quality of fruits and was not able to meet the growing demands of the vegetables. Hence, there was a need of growing crops in the covered structure in order to increase the yield.

They constructed a bamboo framed structure with green shade nets with different heights and different shades, so that they can modify the in-house environmental conditions with reduced labour and could grow the crops in any season. There was significant increase in yield and quality of the fruits. It was observed that there was less disease and pest attack in shade net houses compared to open field growth of all crops.

The crops which are grown in open field cannot get the required growth conditions in proportion like humidity, soil moisture, temperature etc., because of unfavorable weather conditions and hence monitoring of these parameters are required. Greenhouse is one of the solution to grow plants under natural environmental conditions. Cultivation of vegetables in a protected environment could be used to improve yield quality and quantity.

P. S. Asolkar et. Al.[15] monitored the in-house parameters Soil moisture, temperature, Light intensity, Humidity, CO_2 Gas of the greenhouse with the help of Atmega32 microcontroller as central processing unit and GSM Communication. The sensors used to sense the parameter are LM35, SY-HS220, Two copper probes, LDR and MQ5 gas sensor respectively. The Crops selected for analysis and prediction of the greenhouse were Papaya, Brinjal, Tomato, Cucumber and Chilies. By implementing the monitoring system in Greenhouses, they can provide automatic controlling techniques which are effective for improvement in crop production compared to old growing methods resulting in reduction of human efforts required for growing crops in open field. They have presented Global system for mobile communication (GSM) system for controlling and monitoring of greenhouses. The method was found to be very effective as it allows parameters of greenhouse like moisture, humidity, temperature etc., to be controlled from a remote area to the desired location. They found that productivity of the crops was much better when the environmental conditions were controlled. They also estimated the total power consumption for controlling process for particular crops.

Daniel Puschmann et. al.[13] introduced a method which was able cluster the IoT data streams from various sources by adaptive clustering method. IoT technology has grown vast and has caused production of large quantity of real-time data. Due to this, there was a requirement of effective technique to process the large data streams of IoT to obtain actionable information. The problem with many stream clustering methods were, they need to know in advance the number of clusters that can be found within a data stream. Hence they introduced adaptive clustering algorithms. Based on the changes in the data stream, the algorithm clusters the data. The distribution of the data gives good indications of the categories. Number of clusters required to group the data is given by probability distribution curve and shape.

In probabilistic distribution, the directional change in the graph is said as turning point (tp). The turning point signifies the beginning of the new category. By this approach, wider and sparser cluster are obtained in the areas with less data points, whereas smaller and denser cluster is obtained in the areas with many data points. Possible initial centroids are considered by centers of the graph of these areas.

They have proposed a method to determine the number of clusters that can be found in a stream based on the data distribution. To group the similar incoming data from the streams, online clustering mechanism is used. According to different criteria like Similarity and homogeneity data is usually clustered. The approach remains flexible to drifts by adjusting itself as the data changes. The method can be applied to real-time traffic data. According to the data distribution, the results allow clustering, labeling and interpreting. It enables to flexibly process huge volumes of dynamic data.

Payam Barnaghi et. al.[26] in their work presented the challenges in finding and extract actionable knowledge from raw data of Web of Things. They stated that there are 20 quintillion bytes of data being produced every day from different sources which contains audios, videos, images and textual contents. The data are collected from the sources like

Smartphones, Sensor Devices, Social medias, wearable devices, etc. which are represented in numerical form or the symbolic descriptions of occurrences in the physical world. The quality, quantity, validity of the data and trust of the data are the challenges in the big data especially in the situation where data is made available to multiple users as there will data related to environment, events and people and hence privacy and security are key concerns. When multiple parties access and process the data, dealing with these issues are difficult. The large number of data coming from the sensory devices requires more bandwidth and reliable Quality of Service solutions. Hence, methods like preprocessing which can be aggregation, summarization, abstraction can help to reduce the size of the data at source level. Knowledge discovery requires set of mechanisms. As different users provide variety of data, there can be inconsistency in the data like errors in reading the data. The data are dynamic in nature and is difficult to discover the knowledge from it.

Frieder Ganz, Daniel Puschmann, et al.[27] proposed techniques for Information Processing and information Abstraction in IoT. They have presented a number of techniques for processing the data and to transform it to higher level abstraction from raw unstructured-data. As the sensor devices generate numerous amount of data, a particular technique cannot be applied for the dynamic data to extract the information. They have provided a survey of the requirements and solutions to extract meaningful and higher level information from raw sensor data. Several methods have been presented from semantic web, machine-learning, pattern recognition and data mining to abstract the information. Pre-processing methods like signal pre-processing which cuts the certain parts of the signal before and after certain frequency value. Mathematical/Statistical Pre-processing methods like min-max value and mean-median values, the values are filtered out. Dimensionality Reduction methods like Discrete Fourier Transformation, Piecewise Aggregation Approximation and Symbolic Aggregate Approximation are used to reduce the length and size. Feature Extraction methods such as Clustering, Semantic Reasoning & Representation: The metadata, data and its related context information are represented in a linked graph model. The technical solutions they provided were to use the tools Rapid Miner, WEKA, SAMOA and Orange.

Keiichi Yasumoto et al.[28] surveyed Real-Time processing Technologies of Internet of Things Data Streams. They stated that there will be more than 50 billion connected devices by 2022. The offline analysis of the stored big-data which are available now will cause delay and wastes money resources.

They described several use case scenarios for the real-time utilization of IoT data like live street view, Broadcasting of Ultra-realistic live sports based on User Generated Contents, Real time tracking of pedestrians, Anomaly detection of seniors living alone in real-time. The key challenges faced was to capture the real world events, at anytime and anywhere. Networking technology to enable direct flow of the data between producers and consumers in real-time and aggregating the data selecting only the necessary streams. As existing IoT platforms do not completely support both distributed and on-site processing. They have proposed a new framework called Information Flow of Things (IFoT). Based on distributed processing among IoT devices, IFoT is used for real-time processing, organizing and analysing of IoT data streams in scalable manner. In IFoT, higher level streams and raw data streams after aggregating, merging and processing are called flows and treated in the same way. Aim of the IFoT was to solve some of the technical issues like unified manner information handling, Processing and analysing flows near them and distributing between the devices in real-time and integrating different flows into higher-level flow and providing it in real-time. By using different layered components, the issues were solved. The layered components were namely IFoT-PO3-Engine, IFoT-Neuron and IFoT-Curator.

IFoT Neuron is abstraction of a sensing device and flow source. It captures the data and processes data in real time and sends them as a flow. IFoT-PO3-Engine offers functions which executes high load tasks that includes online learning and complex event processing. IFoT-Curator executes human-edited instructions. They have proposed a new framework called Information Flow of Things to capture abstraction of real-world-data.

Altti Ilari Maarala et al.[12] presented method for Semantic Data provisioning and Reasoning for the Internet of Things, the goal was to represent the data produced by the IoT nodes to the machine understandable manner, as it facilitates interoperability with various systems and applications and to present the data such that its meaning can be explained and shared efficiently. Real time requirements of IoT data heterogeneity and dynamic nature are challenges for applying the technologies like reasoning and interoperability. They have specified the approaches which are able to deliver semantic data from IoT nodes to distributed reasoning engines. Reasoning over such data are performed. They collected Global Positioning System (GPS) data from taxi drivers. They detected several events from the observations like turns, traffic jams, stopping for a long time, speeding, sudden acceleration and strong acceleration, deceleration etc. The experiments were carried out with both distributed reasoning nodes and single reasoning node. The operations were carried out by collecting a set of messages from IoT nodes. Reasoning is performed over the data by reasoning engine. RDF database stores the reasoned knowledge.

Adnan Akbar et al.,[22] developed an automatic context aware method. It was based on clustering for finding threshold values for context event processing rules. Most of the IoT applications requires real time analysis and are distributed in nature. They generate huge amount of data. The goal was to extract higher level knowledge from the IoT raw sensor data. Based on the threshold values, rules for Complex Event Processing (CEP) are set. There is no automatic method to find the optimized threshold values. For real-time dynamic IoT environments, the threshold vales will always be changing. There are many drawbacks of CEP like, the threshold values has to be set manually. The optimal threshold value cannot be found automatically. When the Threshold value is set, it cannot be changed at run time.

They developed Micro Complex Event Processing (μCEP) to run on low processing hardware which can update the rules on the run. The CEP consists of the two components namely μCEP and Adaptive Clustering. μCEP engine is capable of detecting independent incoming Events of different types. It generates a Complex Event by correlating all the events. Event Collector collects the information coming from the specified source. It is entry point of the μCEP engine. Complex Event Detector controls the event detection and produces the expected outputs. The Complex events are received by Complex Event Publisher. It delivers the data to the selected Data Sinks. Adaptive Clustering groups the similar objects based on the predefined rules.

Daniel Puschmann et al.[23] introduced a novel approach for recognizing the relation between the heterogeneous type of data. The aim was to take out the information from real-world data and to find the correlation between them. To reduce the dimensionality of streaming data, they have used Piecewise aggregate approximation (PAA). The data sets used were of traffic and Weather data. The traffic data streams and weather data streams were collected from the city of Aarhus. PAA takes equal sized frames as parameter and calculates the averages of the frames in the original data. They are represented by their mean value. Symbolic Aggregate Approximation determines the equivalent symbol for the coefficient. It is applied to the PAA output. The obtained information from the pattern along with mathematical values are translated to higher level abstraction. Virtual documents are created to group the higher level abstractions from different sources within a certain time frame. Then to find the correlation between the Latent Dirichlet allocation is used.

DM Blei et. al. [29] described generative probabilistic model which is called latent Dirichlet allocation(LDA) for discrete data. The goal of the work was to find the short representations of the discrete data which gives well processing and essential statistical relationship that helps for the tasks like summarization, classification, similarity novelty detection, etc. They stated that LDA can be viewed as dimensionality reduction technique. The proposed method reduces all documents which are in paragraphs to vector of real numbers. The real numbers represents the counts in ratio.

Chapter 3

REQUIREMENT SPECIFICATION

3.1 Functional Requirements

This section describes the functional requirements of the work, which describes the input, output and the functionality of the system. Following is the brief discussion on functional requirements.

Input: In order to perform the abstraction and semantic representation, the input to the system is the raw data produced by greenhouse sensors. Virtual sensor programs are written which acts like the greenhouse sensors, the data are stored in the csv file.

Output: Abstracted form of the data that can be understood by human and the correlation among the data so that future decisions can be made with the help of these information.

Input 2: The output produced by the abstraction is stored into a file and is further pre-processed with the steps tokenize, stop words and stemming in order to give it to LDA for topic modelling and to find the correlation among the data.

Output: Correlation among the data is known.

The data is read from the nodes in analog values which are converted to the equivalent sensor values. The values which are out of the sensor output range are removed.

The raw numerical values are converted to text format from the numerical values with the help of the provided threshold range values called as abstraction and are stored to text files.

LDA performs topic modelling on the abstracted text and provides the relation among the parameters.

3.2 Non-functional Requirements

Through the non-functional requirements, the requirements that improves the quality of activity recognition system is discussed:

Timeliness: The time taken for training and testing the system is one of the main criteria for the usability of the system. Faster the training and testing performed by the system, higher is the usability of the system. Hence, the system must be designed and implemented in such a way that, the time taken during training as well as the testing phase has to be less. This improves the usability of the system.

Accuracy: The performance of the data abstraction and correlation model is evaluated based on the accuracy it provides. Therefore, higher accuracy is expected for the correlation.

Scalability: The system must to be able to produce outputs in real-time.

3.3 Hardware Requirements

RAM: 2GB or higher. Since the system considers large files for the processing. In order to allow for smoother the working of the system, RAM of 2GB or higher is desired.

Storage: 100GB or higher. This contributes to the amount of storage space required by the instance of the operating system installed, the tools used for the implementation of activity recognition system, as well as storage space for the files of the data-set.

Processor: Intel i5 with 4CPU cores and operating frequency of CPU at 2.60GHz is used. Higher the CPU frequency, and the number of CPU cores, faster the system performs.

3.4 Software Requirements

Operating System:

- **Ubuntu:** Ubuntu is an open source operating system for computers. It is a Linux distribution based on the Debian architecture. Ubuntu operating systems can be used for servers, cloud and personal computers. The operating system allows the software tools, libraries and programming language to be used for the activity recognition system.

Programming Language

- **Python:** Python is a high-level, general-purpose, interpreted programming language. Python supports multiple programming paradigms that include object-oriented, functional and procedural. Python also provides a large library of API's that can be used to make programming much simpler. Python interpreter is available for many operating systems.

Software Libraries Required

- **Numpy:** It is a library written in the python in order to work on large multidimensional arrays and matrices. It also provides with functions that allows use of high level mathematical functions to operate on the multidimensional matrices.
- **Nltk:** Nltk (Natural Language Toolkit) is a suite of library used for natural language processing.
- **Nltk.tokenize:** It is a text processing library called Tokenizer is used to split the sentences to words from a body of text. It also removes punctuation symbols from the text. Tokenize library breaks up the sequence of strings to pieces called worlds.
- **Stop_words:** These are words that are filtered out eg: a, an, in, the, are, etc., before or after processing of natural language. The text after tokenization may contain the stop words which are useless. There are multiple set of stop words available in multiple languages.

- **nltk.stem.porter:** It is a library of python for removing the common inflexional endings and morphological words from the stop words. There will be many worlds which are same or carries the same meaning.
- **Gensim:** It is a Vector Space model used to handle large collections of text. It is used to extract semantic topics from the document or text.

3.5 Other Requirements

- **Data-Set of Greenhouse IoT data:** To train the system and perform the operation, data set of the Greenhouse IoT data of the different parameters are required.

Chapter 4

SYSTEM DESIGN

4.1 High Level Design

4.1.1 System Design

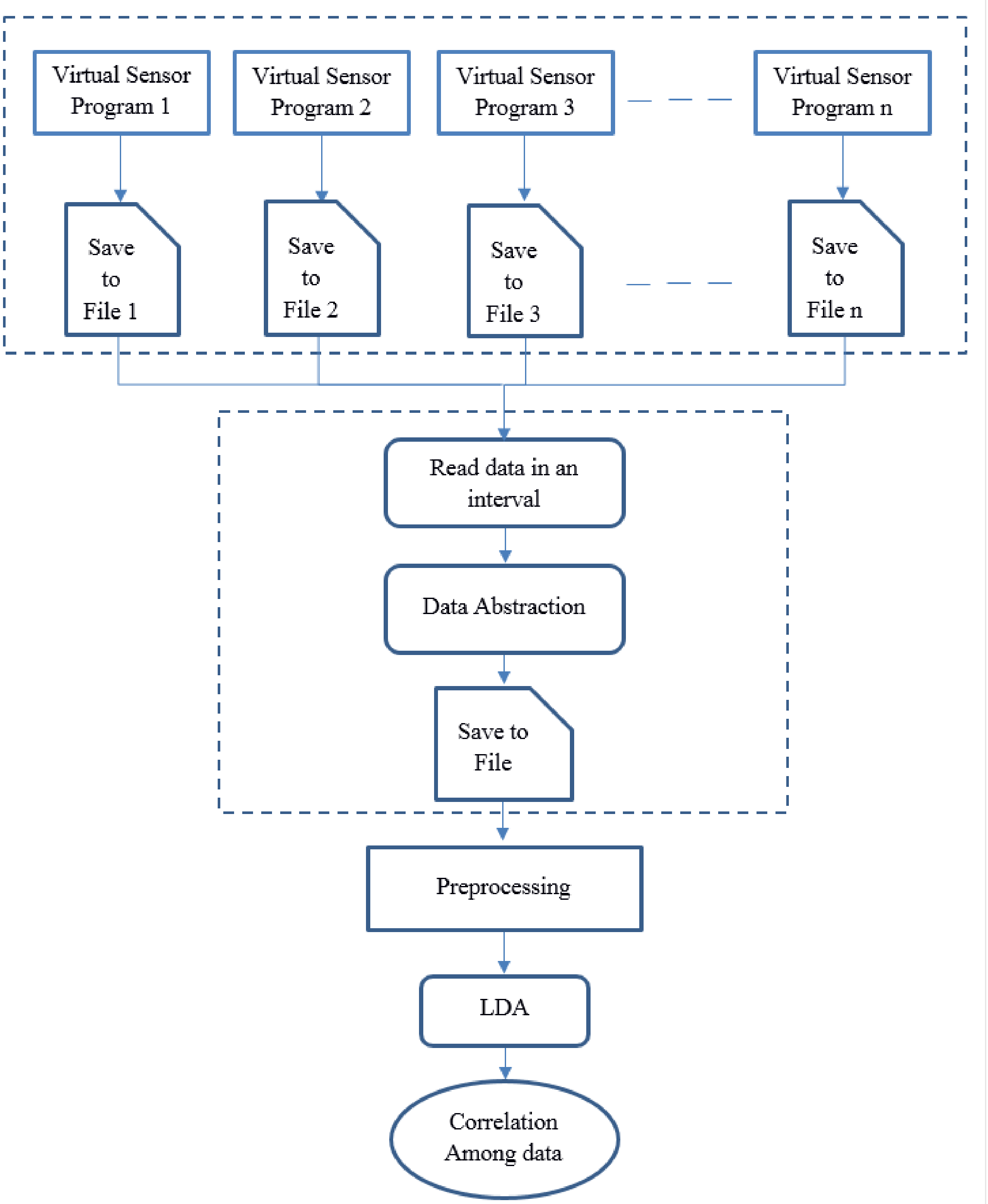


Fig 4.1: Architecture of the system

The overall system architecture for information abstraction is shown in figure 4.1, which represents the overview of the model.

Following is the brief description of the system design for Information abstraction and correlation.

- Virtual sensor programs are created which acts like the sensors of the greenhouse producing real time data.
- While generating the data, randomness is introduced so that there will be some outlier.
- The generated data are stored in separate files for different sensors.
- With a specific time interval, the data is read from the file and is converted to the respective sensor parameters unit.
- There are 2 types of sensors. Analog sensor and Digital sensors. A 10 bit analog sensor gives data in the range from 0 to 1023 which is then converted by the Analog to Digital converter to normal form of the data as per the parameter. Whereas the digital sensor gives the data in normal form.
- The value is converted to string format with the help of threshold range of the respective parameters.
- The semantic or abstract form of the value is stored in a separate file.
- The data has to be cleaned for further processing.
- There are 3 techniques applied in preprocessing namely Tokenizing, Removing stop words and stemming the words.
- In order to obtain the correlation among the data, LDA is applied.

4.2 Detailed Design

4.2.1 Data Flow Diagrams

Data flow diagram is a graphical representation that depicts how the data flows through the system and among various sub-processes within the system. It gives a clear picture of what data goes into a process and what comes out.

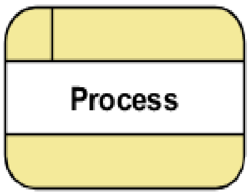
DFD includes different symbols representing external entities, processes, data stores and data flow. Different people have proposed a different set of symbolic representation for DFD. The symbols proposed by Gane and Sarson are used to depict data flow diagrams and the symbols are as follows:

External Entity

Entity represents data source or destination.

D
DataStore

Data store represents database or the place that holds data between processes.



Process represents the task or function that does some processing on the input data to produce and output.

Data flow represents the path of data flow between entities, data store and processes.

Level 0:

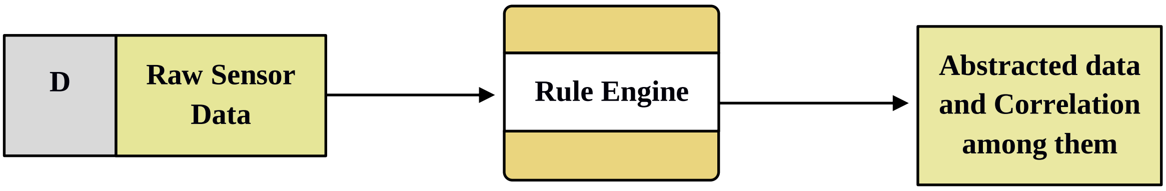


Figure 4.2: Data Flow Diagram level-0

The Context diagram or the top level data flow diagram representing the whole system as a single process is as shown in figure 4.2.

The rule engine performs the operation on the raw data which is in numerical form to get the abstracted data and to get correlation among them.

Level 1:

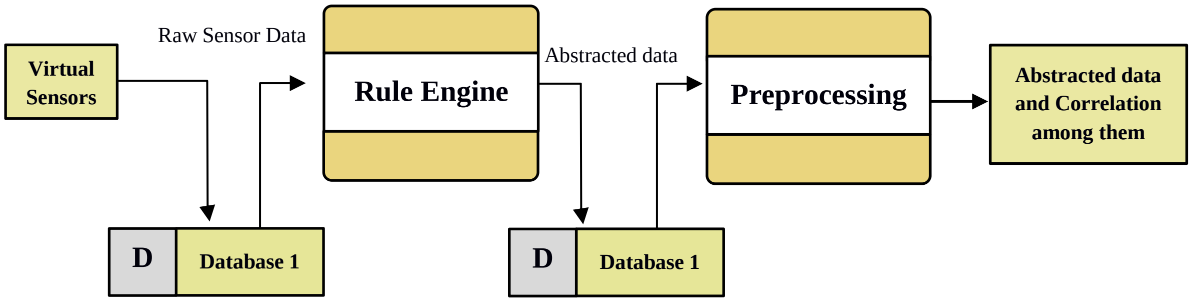


Figure 4.3: Data Flow Diagram level-1

The level-1 Data flow diagram depicts a level more detailed flow of data than level 0, among the sub-processes within the system as shown in figure 4.3.

- The sensor data is written to the files in the database according to the parameters range and sensors bit resolution.
- Data is read from the files with a specific interval of time and sent to rule engine to perform the operation on the data. The Abstracted data is stored in a separate file in the database.
- Preprocessing step cleans the abstract data to further perform operation on it and later the correlated data is stored in the database.

Level 2

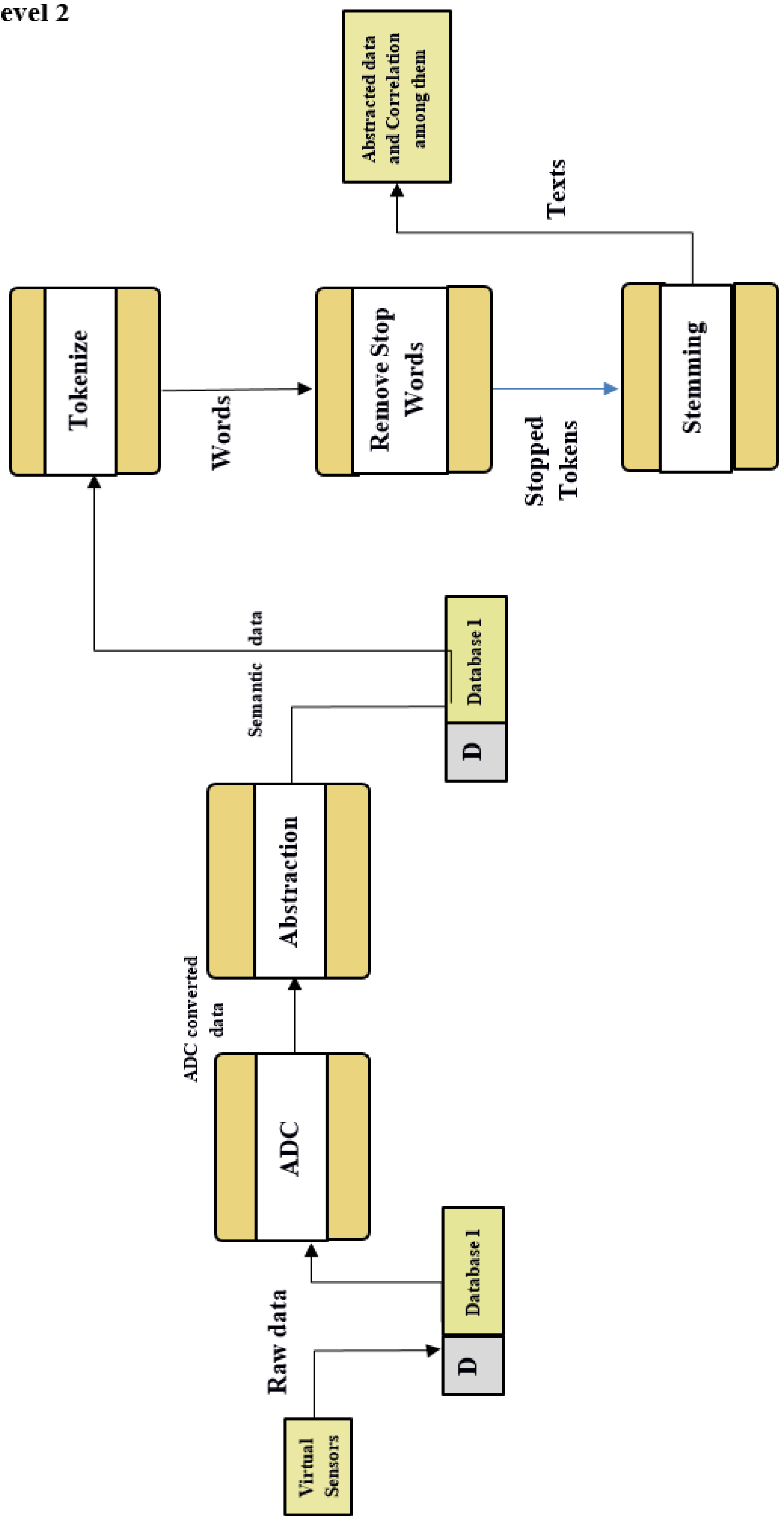

Virtual Sensors
Raw data
D
Database 1
ADC
ADC converted data
Abstraction
D
Database 1
Semantic data
Tokenize
Words
Remove Stop Words
Stopped Tokens
Stemming
Texts
Abstracted data and Correlation among them

Figure 4.4: Data Flow Diagram level-2

Fig 4.4 depicts the level 2 data flow which gives much detailed flow of data among each and every sub-task in the system and the brief description of it is as follows.

- For real-time processing, we need greenhouse data. The numerical data is produced according to the type or sensor, analog or digital, by the virtual sensors and stored in separate files in the database.
- The data is read by the program with the given time interval. As the analog sensors used are of 10 bit resolution, the range of data will be within 0 to 1023 which needs to be converted to meaningful data.
- Analog to Digital converters are the one which converts the analog signal to respective unit format. There are different types of ADC for different types of sensors.
- The unit format data are given a semantic name according to the rules specified.
- The semantic form of the data is stored in the in the database which is retrieved for data cleaning process.
- The data cleansing process consists of three steps namely Tokenizing, Removing of stop words and Stemming.
- Tokenization splits the text or sequence of string to words.
- Stop words remove the common language terms and repeated words.
- Stemming is the process of removing the words which are similar.
- The cleaned data is further given to LDA algorithm to obtain the correlation among the data.

Chapter 5

IMPLEMENTATION

5.1 Overview of Technologies Used

Python:

Python is a high-level, general-purpose, interpreted programming language. Python supports multiple programming paradigms that include object-oriented, functional and procedural. Python also provides a large library of API's that can be used to make programming much simpler. Python interpreter is available for many operating systems.

Latent Dirichlet Allocation:

Latent Dirichlet Allocation (LDA) is a generative probabilistic model in which according to the given corpus the topics are assigned to the document.

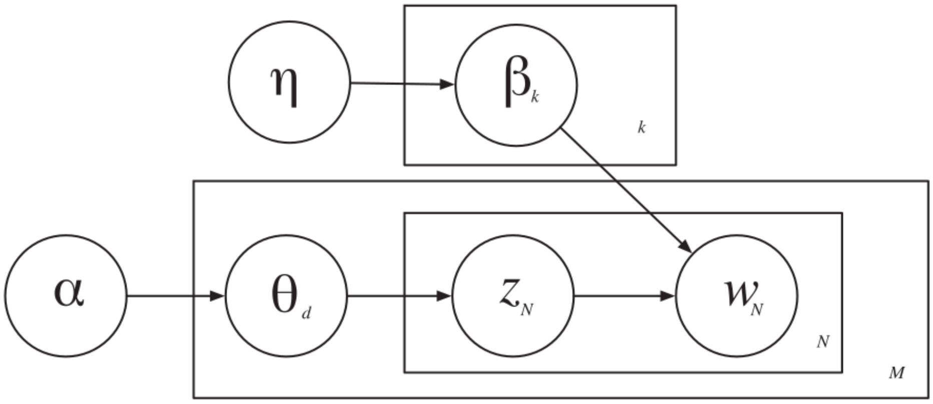


Figure 5.1: Plate model Representation of LDA

Fig shows the plate model representation of LDA. Where α is per document topic distribution and β is per topic word distribution. w is the word in the document, N is the collection of the words represented by vector W_N, M is the total document set, k represents the number of topics which is fixes at the initial, β is the multinomial distribution of words which represents the topics, θ is drawn from the Dirichlet prior. z_{ij} is the topic that is most likely to have generated w_{ij}. Here, j is the word count and i is document count.

There are three levels in LDA model which can be said as documents, terms and topics. The document contains many terms and topic is distribution over terms. At first, LDA determines the number of words in a document. Then it determines the mixture of topics in that document. There will be multiple topics defined and by using each topics multinomial distribution, output words to fill the documents word slot.

The main intension of LDA is topic distribution and to find the relation among the documents, topic and terms.

Internet of Things:

Internet of Things is network of the physical devices where the devices have the ability to generate and exchange the data over the network. The technology makes it possible to access, monitor and control the devices remotely. With the connectivity of devices and improved tracking one can benefit in real-time data. The technology helps individuals and organizations by reducing work and cost through automatic and efficient processes. Convergence and growth of data, processes makes the internet more relevant creating more number of opportunities for decision making and business.

Here, virtual sensor programs are written which acts like the sensors used in greenhouse.

Data Analytics:

Data Analysis is the process of examining the data to draw a conclusion from it. It involves applying algorithms, formulas etc., to obtain the insights. Here data abstraction is used to convert the numerical sensor value to equivalent string format. The abstracted data is given to LDA to find the correlation in the processing.

Machine Learning:

Machine learning is a field of computational science which compromises of numerous technologies to build a system which can learn from the data provided and make predictions or decisions when met with the new situation. It is a technique that allows the system to learn from training provided to it and improve by itself from the experience without actually being programmed.

The machine learning tasks are of basically two type’s namely supervised and unsupervised learning. The supervised learning involves training the algorithm with the dataset that contains correct labels or outputs to the respective set of inputs, where the algorithm learns the patterns that match the input to the given output. Based on this learning it predicts the output for a given set of new inputs. In case of unsupervised learning, the algorithm is provided with only the set of inputs without any output mapping. The algorithm learns the hidden pattern in and among the input parameters and uses it in a grouping or clustering the given data.

Artificial Intelligence:

Artificial Intelligence is way to make the computers, computer controlled devices or software to think intelligently. By defining the rule engines, a computer can perform operations based on these set of rules.

5.2 Implementation Steps for Data Abstraction and finding Correlation

5.2.1 Sequence Diagram

Fig depicts the sequence of operations performed for abstraction and finding correlation.

It can be explained as below:

- Raw data from Greenhouse is required for processing. As the processing is for real time, a greenhouse has to be constructed in order to get the real time data. Instead of creating a greenhouse, virtual sensor programs are created which acts like real time greenhouse sensors producing data for interval of 100 milliseconds.
- The raw data being produced are stored to different files.
- The data is then read in 5 minutes interval once.
- Analog sensors gives the output in 0 to 1023 range which has to be converted to the respective sensors parameter unit to make it meaningful.
- The numerical value is then converted to the equivalent string format i.e., abstracted form and stored to a separate file.
- Tokenizer splits the string to substring using regular expression.
- Stop words removes the repeated words from the document and words like conjunctions which are normally used in natural language.
- Stemming is the process of removing the similar words.
- LDA algorithm obtains the preprocessed data semantic data from the file.

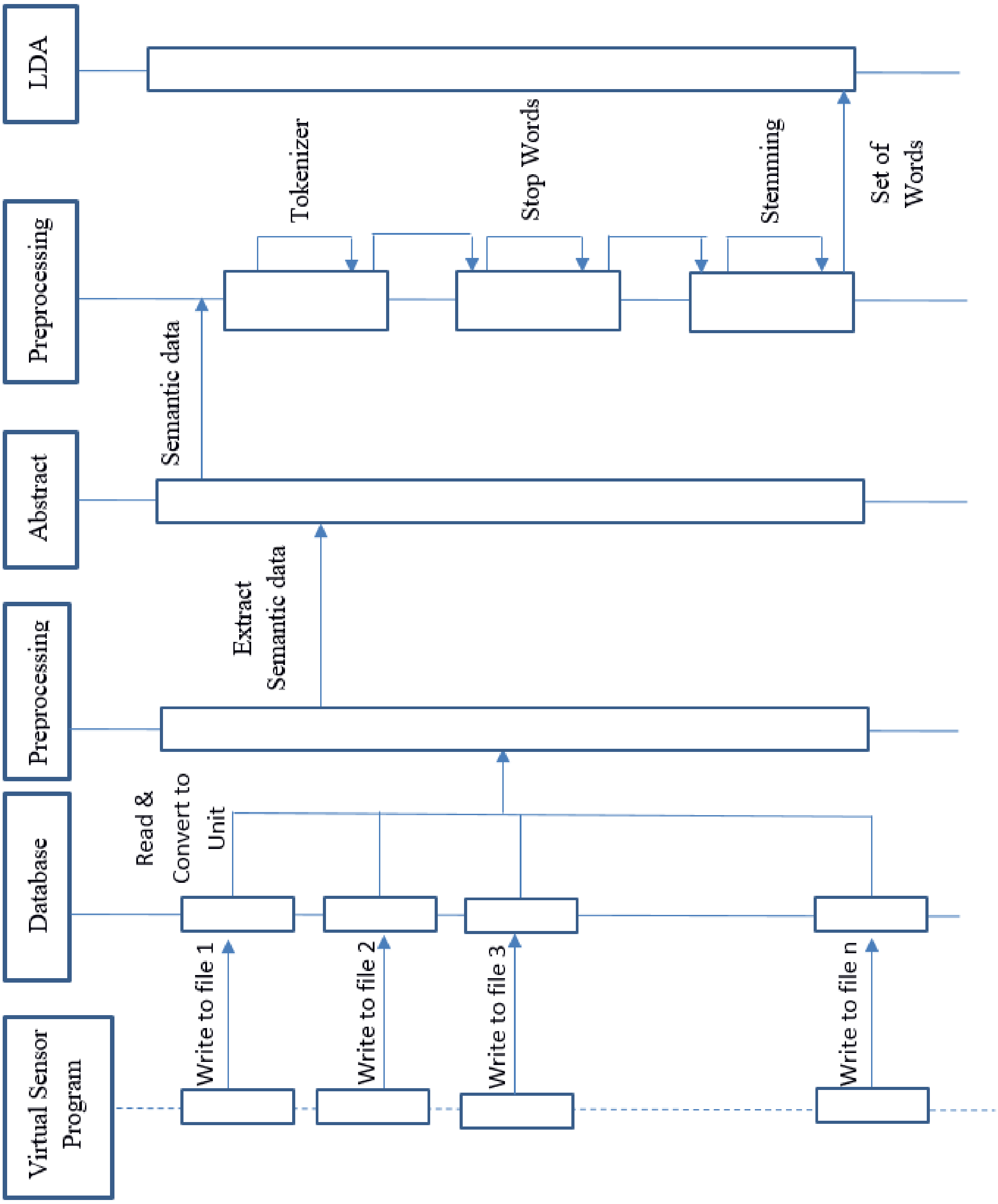


Figure 5.2: Sequence Diagram

5.2.2 Functional Description of Module

Module to read the dataset

- **read _csv:**
  - Input: Path of the dataset in the directory.
  - Reads the Column of the CSV file and returns the data.
  - Output: Dataset in the form of Pandas data frame.

Module for Data Pre-processing

- **Analog to Unit conversion:**
  - Input: Raw sensor data ranging from 0 to 1023.
  - ADC (Analog to digital Conversion): There are different types of analog sensors for respective parameters.
  - Output: Converted data to Respective parameter unit.

- **Data Abstraction:**
  - Input: Parameters unit data which will be in numerical form.
  - Based on the threshold value, the numerical data is converted to the string format.
  - Output: Semantic form of the data which is represented in string form.

Preprocessing

- **Tokenization:**
  - Input: Semantic data which is in the form of string format.
  - The stream of strings or sentences are broken into individual words.
  - Output: Breaking of a sentence or stream of text to words.

- **Stop Words:**
  - Input: Set of tokenized words.
  - Stop words remove the repeated words and most common words used in a language.
  - Output: Words after removal of stop word.

- **Stemming:**
  - Input: Set of words which is processed by stop words.
  - Stemming process removes the derivationally related form and inflectional form of a common rooted word.
  - Output: Set of words which do not have inflection.

Latent Dirichlet Allocation:

LDA undergoes 3 steps

a) It determines the number of words in the document.

b) It determines the mixture of topics in the document.

c) Using multinomial distribution of each topic, output words to fill the documents slot.

5.3 Implementation details of the Modules

There was the requirement of the real-time greenhouse streaming data in order to perform the operation. Instead of setting the greenhouse, virtual sensor programs are created which acts like the greenhouse data. Table 1.1 shows the sensors used in greenhouse which are taken into consideration.

The sensors considered have analog and digital sensors in the list. Digital sensors directly senses the environment and gives the output. Whereas analog sensors, based on the bit resolution, range of sensing, etc., they sense the environment and give the output. The 10 bit resolution sensor gives the output range from 0 to 1023 resulting in 1024 total values which is equivalent to 2^{10}. Whereas 16 bit resolution sensor gives value in the range 0 to 65535 which is equivalent to 2^{16}. There are sensors available which works in 8 bit, 10 bit, 16 bit, etc. According to the type of sensor, analog or digital, different programs are written which generates the data with an instance of 100 milliseconds.

While generating the data, there is some randomness introduced in order to get variations.

The data is read from the every file in 3 minutes interval each. To get the data in the respective unit form, there has to be equivalent converter to convert the analog signal to respective parameter unit value. Different sensors have got different analog to respective unit converter methods. According to the parameter and the unit conversion, the data is converted to meaningful unit form.

The data is in the numerical form still now. Based on the set of conditions, the data is converted to string format.

The conditions are specified in the table:

Table 5.1: Conditions considered for the parameters

Parameters	Low	Normal	High
CO2	<700 ppm	700 – 1000 ppm	>2000ppm
Inside Humidity	<40	40 – 45 %	>45%
Outside Humidity	<50 %	50 – 55 %	>55 %
Luminosity	<1000 lux	1000 – 2000 lux	2000 lux
Radiation	<977 W/m^2	977 - 1000 W/m^2	<1000 W/m^2
Soil Moisture	<20%	20 – 30 %	>30 %
Soil Temperature	<20 °C	20 – 25 °C	>25 °C
Temperature Inside	<20 °C	20 – 25 °C	>25 °C
Temperature Outside	<25 °C	25 – 30 °C	>30 °C

Based on the above conditions, the numerical data is converted to the respective string format and stored in a text file.

For e.g., if the inside humidity is < 40%, then the abstracted format of the data is "*humidity_inside_is_low*". If it is within the range 40 and 45, then append the text "*humidity_inside_is_normal*" else if it is greater than 45%, then append the text "*humidity _inside_is_high*".

According the value and ranges of particular parameters, the respective semantic data is appended to file which is used for further processing.

Preprocessing:

- **Tokenizer:** Tokenization is the act of breaking down the set of strings into pieces. It may contain words, symbols, phrases, keywords, etc. The abstracted data is given as the input to this model. It breaks the strings to form words.

- e.g., below is the content of the document

  doc = “soil_moisture_is_normal radiation_is_normal temperature_inside_is_low temperature_outside_is_low humidity_inside_is_high co2_is_normal luminosity_is_normal humidity_outside_is_high soil_temperature_is_normal soil_moisture_is_normal radiation_is_normal temperature_inside_is_normal temperature_outside_is_normal humidity_inside_is_high co2_is_normal luminosity_is_normal humidity_outside_is_normal soil_temperature_is_normal soil_moisture_is_normal radiation_is_normal temperature_inside_is_high temperature_outside_is_high humidity_inside_is_low co2_is_normal luminosity_is_normal humidity_outside_is_low soil_temperature_is_normal”

- After tokenization the data looks like

  tokens = ['soil_moisture_is_normal', 'radiation_is_normal', 'temperature_inside_is_low', 'temperature_outside_is_low', 'humidity_inside_is_high', 'co2_is_normal', 'luminosity_is_normal', 'humidity_outside_is_high', 'soil_temperature_is_normal', 'soil_moisture_is_normal', 'radiation_is_normal', 'temperature_inside_is_normal', 'temperature_outside_is_normal', 'humidity_inside_is_high', 'co2_is_normal', 'luminosity_is_normal', 'humidity_outside_is_normal', 'soil_temperature_is_normal', 'soil_moisture_is_normal', 'radiation_is_normal', 'temperature_inside_is_high', 'temperature_outside_is_high', 'humidity_inside_is_low', 'co2_is_normal', 'luminosity_is_normal', 'humidity_outside_is_low', 'soil_temperature_is_normal']

- **Stop Words:**

  Here, Stop words remove the same words which appear twice, as there is no common words of natural language in the doc.

  ```
  stopped_tokens = ['soil_moisture_is_normal', 'radiation_is_normal',
  'temperature_inside_is_low', 'temperature_outside_is_low',
  'humidity_inside_is_high', 'co2_is_normal', 'luminosity_is_normal',
  'humidity_outside_is_high', 'soil_temperature_is_normal',
  'soil_moisture_is_normal', 'radiation_is_normal', 'temperature_inside_is_normal',
  'temperature_outside_is_normal', 'humidity_inside_is_high', 'co2_is_normal',
  'luminosity_is_normal', 'humidity_outside_is_normal',
  'soil_temperature_is_normal', 'soil_moisture_is_normal', 'radiation_is_normal',
  'temperature_inside_is_high', 'temperature_outside_is_high',
  'humidity_inside_is_low', 'co2_is_normal', 'luminosity_is_normal',
  'humidity_outside_is_low', 'soil_temperature_is_normal']
  ```

- **Stemming:**

  Here, there are no words that are in derivationally related form or inflectional form of a common rooted word.

Chapter 6

TESTING, EXPERIMENTAL ANALYSIS AND RESULTS

6.1 Unit Testing

Unit testing is a part of software testing that is conducted to ensure that each small unit or module of the system performs as designed and expected. Following are the unit test cases are written for the implemented system.

Test case 1: Test case to check the generation of the data which acts like daemon.

For each virtual sensor program, it is expected to run until user stops it.

Test Results: pass

Test Case 2: Test case for the conversion of numerical data to semantic form

There are multiple values being read from the file simultaneously which has to be checked for the conditions and map it to the respective string and write it to the text file. The data has to be either low, normal, high or abnormal. The process was checked with different values and got the results.

Test Result: Pass

Test Case 3: Test case to check the working of tokenizer.

The document obtained from the abstraction process has to brake up to pieces to form individual words. The strings were manually replaced with other attributes and punctuations to check the working of tokenizer.

Test Result: Pass

Test Case 4: To check the working of stop words.

The document set contains the words processed by the tokenizer. The stop words has to remove the words which are repeated and the common words used in natural language. Manually inserted some conjunctions to the file and checked whether it is removed by stop word process.

Test Results: Pass

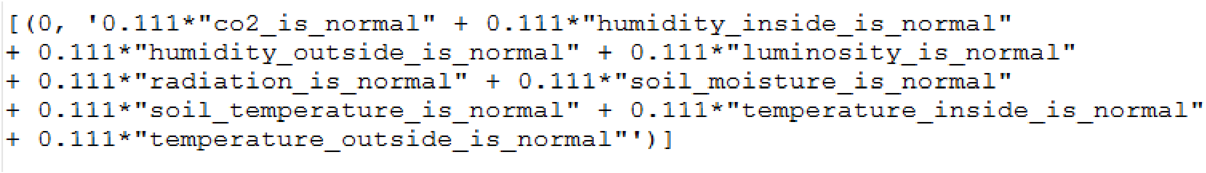
6.2 Results and Discussion

Greenhouse parameters which are monitored in the greenhouse are considered namely temperature, humidity, co2 concentration, luminosity, radiation, soil moisture and soil temperature. In order to get the real-time data, virtual sensor programs are written which acts as sensors in the greenhouse and creates the data according to the conditions specified in the table 1.1. There can be situation where greenhouse sensors gives some error values such as out of range, inconsistent value etc. The analog sensors gives the output in the range of 0-1024. In some cases, it may go beyond this range or it might give unrealistic values. For example: The normal room temperature exists within 20° C to 35° C. If the output of the sensor gives the temperature is beyond this value, then we can assume that there is some error with the sensor values. So, the virtual sensor programs are introduced with an outlier concept where 80% of the values are within the normal range and 20% of the values are out of this range. The data creation is continuous and can be only stopped by breaking it or with any external command. The raw data created by different virtual sensor programs are stored in different files each.

`The data are stored in the files with the numerical range of 0-1023 if the sensor is analog or else the respective unit sensor output value. The data is read in the interval of 3 minutes. In the case of analog sensor, the raw sensor values are converted to respective parameter unit according to the sensor conversion formula and checked with the respective conditions provided. Then the value is assigned with the equivalent text phrases and saved in different files. For example: if the analog value for the parameter "inside temperature" is 163, the equivalent temperature value the greenhouse which is 30° C is given converted to text phrase "inside_temperature_is_normal".

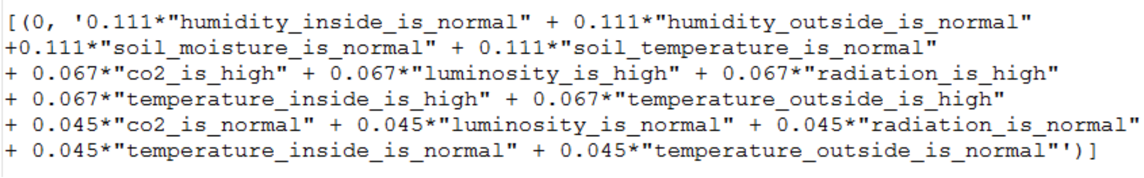
`Then the text file is read by the LDA module which calculates the probability of occurrences of every phrases. The correlation among the parameters of the greenhouse is found by it. LDA reads 10 text files at a time in which each individual text file contains 10 samples of abstracted data. The LDA algorithm was tuned to produce 1 topics on each run with various data sets.

Results obtained from LDA with changes in the parameter.



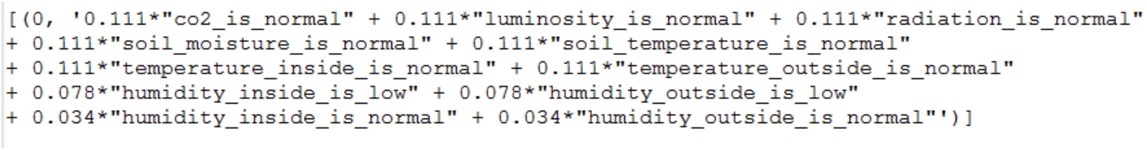
```
[(0, '0.111*"co2_is_normal" + 0.111*"humidity_inside_is_normal"
+ 0.111*"humidity_outside_is_normal" + 0.111*"luminosity_is_normal"
+ 0.111*"radiation_is_normal" + 0.111*"soil_moisture_is_normal"
+ 0.111*"soil_temperature_is_normal" + 0.111*"temperature_inside_is_normal"
+ 0.111*"temperature_outside_is_normal"')]
```

Fig 6.1: LDA results when all the parameters are kept normal



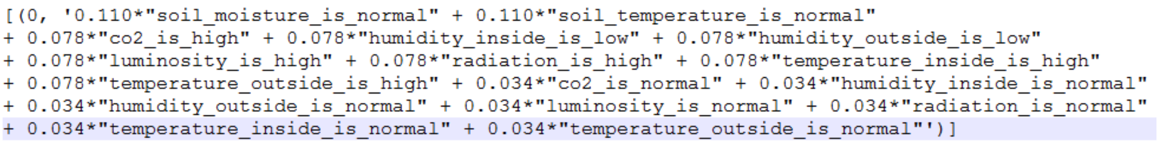
```
[(0, '0.111*"humidity_inside_is_normal" + 0.111*"humidity_outside_is_normal"
+0.111*"soil_moisture_is_normal" + 0.111*"soil_temperature_is_normal"
+ 0.067*"co2_is_high" + 0.067*"luminosity_is_high" + 0.067*"radiation_is_high"
+ 0.067*"temperature_inside_is_high" + 0.067*"temperature_outside_is_high"
+ 0.045*"co2_is_normal" + 0.045*"luminosity_is_normal" + 0.045*"radiation_is_normal"
+ 0.045*"temperature_inside_is_normal" + 0.045*"temperature_outside_is_normal"')]
```

Fig 6.2: LDA results with variable parameters1



```
[(0, '0.111*"co2_is_normal" + 0.111*"luminosity_is_normal" + 0.111*"radiation_is_normal"
+ 0.111*"soil_moisture_is_normal" + 0.111*"soil_temperature_is_normal"
+ 0.111*"temperature_inside_is_normal" + 0.111*"temperature_outside_is_normal"
+ 0.078*"humidity_inside_is_low" + 0.078*"humidity_outside_is_low"
+ 0.034*"humidity_inside_is_normal" + 0.034*"humidity_outside_is_normal"')]
```

Fig 6.3: LDA results with variable parameters2



```
[(0, '0.110*"soil_moisture_is_normal" + 0.110*"soil_temperature_is_normal"
+ 0.078*"co2_is_high" + 0.078*"humidity_inside_is_low" + 0.078*"humidity_outside_is_low"
+ 0.078*"luminosity_is_high" + 0.078*"radiation_is_high" + 0.078*"temperature_inside_is_high"
+ 0.078*"temperature_outside_is_high" + 0.034*"co2_is_normal" + 0.034*"humidity_inside_is_normal"
+ 0.034*"humidity_outside_is_normal" + 0.034*"luminosity_is_normal" + 0.034*"radiation_is_normal"
+ 0.034*"temperature_inside_is_normal" + 0.034*"temperature_outside_is_normal"')]
```

Fig 6.4: LDA results with variable parameters3

10 files containing 10 samples of abstracted text phrases each are passed to LDA. At first all the parameters are kept normal and checked with the results. Figure 6.1 shows the result in which probability of every parameter were kept to normal. In this test, the probability of every parameter was obtained same.

In second test, the parameters like inside temperature, outside temperature, luminosity, radiation and co2 of some files are changed from normal to high. The result obtained are shown in figure 6.2. It can be noted that inside temperature is high when the outside temperature is high which has got the same probability and are related. Radiation is high and luminosity is high has got the same probability and are related to each other and co2 are correlated to each other. In the third test, inside humidity and outside humidity are changed from normal to low in some of the files and obtained same probability which is shown in figure 6.3. In the fourth test, inside temperature, outside temperature, co2, luminosity and radiation are changed from normal to high. Inside humidity and outside humidity are changed from normal to low and noted that the parameters like inside temperature, co2, radiation and luminosity changes when the outside temperature is changed. Inside humidity is affected by the outside humidity.

It can be noted from the above results that if the parameters are related to each other, the probability associated with the parameters changes and if they are highly related, then probability are very much closer.

The following correlations were found out.

1) Outside temperature is high is highly correlated with inside temperature being high. This is also related to both inside and outside humidity as high and other factors like radiation and CO2 both being high.

2) Humidity inside being low is related to humidity outside being low, normal luminosity and normal soil moisture and CO2 being normal.

3) Also, temperature is normal, co2 and humidity normal is related to all other attributes being normal.

Chapter 7

CONCLUSION

There are numerous number of devices connected to the internet which are regularly produce and exchanging huge amount of data and either limited to specific domain or unused later. There is an effective methodology required to process the real time data and extract the information in it.

We have implemented a novel approach to extract the Information hidden in the raw IoT data from Greenhouse and to find the correlation among the data. The process involves collecting the raw data from the sensory devices in a particular frequency and representing the equivalent text form for respective sensors data. The semantic form of the data is further preprocessed with the techniques tokenization, removal of stop words and stemming. Further the abstracted data is given to LDA, a topic modelling method to find the correlation among different parameters of the Greenhouse. Hence, LDA can be used to find the correlation among the greenhouse parameters.

References

[1] S. Pandey and A. Pandey, “GREENHOUSE TECHNOLOGY,” *Int. J. Res. - GRANTHAALAYAH*, vol. 3, pp. 1–3, 2015.

[2] R. Tönjes *et al.*, “Real time iot stream processing and large-scale data analytics for smart city applications,” in *poster session, European Conference on Networks and Communications*, 2014.

[3] M. Lakshminarasimhan, “Advanced Traffic Management System Using Internet of Things.”

[4] A. Khanna and R. Anand, “IoT based smart parking system,” in *Internet of Things and Applications (IOTA), International Conference on*, 2016, pp. 266–270.

[5] M. Al-Kuwari, A. Ramadan, Y. Ismael, L. Al-Sughair, A. Gastli, and M. Benammar, “Smart-home automation using IoT-based sensing and monitoring platform,” in *Compatibility, Power Electronics and Power Engineering (CPE-POWERENG), 2018 IEEE 12th International Conference on*, 2018, pp. 1–6.

[6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[7] S. Li, H. Wang, T. Xu, and G. Zhou, “Application study on internet of things in environment protection field,” in *Informatics in Control, Automation and Robotics*, Springer, 2011, pp. 99–106.

[8] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014.

[9] A. Arora *et al.*, “Exscal: Elements of an extreme scale wireless sensor network,” in *Embedded and Real-Time Computing Systems and Applications, 2005. Proceedings. 11th IEEE International Conference on*, 2005, pp. 102–108.

[10] E. J. Keogh and M. J. Pazzani, “Scaling up dynamic time warping for datamining applications,” in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000, pp. 285–289.

[11] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, “A symbolic representation of time series, with implications for streaming algorithms,” in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, 2003, pp. 2–11.

[12] A. I. Maarala, X. Su, and J. Riekki, “Semantic data provisioning and reasoning for the internet of things,” in *Internet of Things (IOT), 2014 International Conference on the*, 2014, pp. 67–72.

[13] D. Puschmann, P. Barnaghi, and R. Tafazolli, "Adaptive clustering for dynamic IoT data streams," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 64–74, 2017.

[14] Y. Qin, Q. Z. Sheng, N. J. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, "When things matter: A survey on data-centric internet of things," *J. Netw. Comput. Appl.*, vol. 64, pp. 137–153, 2016.

[15] P. Asolkar and U. Bhadade, "Analyzing and predicting the green house parameters of crops," *Int. J. Comput. Appl.*, vol. 95, no. 15, 2014.

[16] K. Rangan and T. Vigneswaran, "An embedded systems approach to monitor green house," in *Recent Advances in Space Technology Services and Climate Change (RSTSCC), 2010*, 2010, pp. 61–65.

[17] L. Liu and Y. Zhang, "Design of greenhouse environment monitoring system based on Wireless Sensor Network," in *Control, Automation and Robotics (ICCAR), 2017 3rd International Conference on*, 2017, pp. 463–466.

[18] T. Ahonen, R. Virrankoski, and M. Elmusrati, "Greenhouse monitoring with wireless sensor network," in *Mechtronic and Embedded Systems and Applications, 2008. MESA 2008. IEEE/ASME International Conference on*, 2008, pp. 403–408.

[19] G. Yun-feng and others, "Greenhouse environment monitoring system design based on WSN and GPRS networks," in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015 IEEE International Conference on*, 2015, pp. 795–798.

[20] M. Tamaki *et al.*, "Daily CO2 changes monitored by an environment sensor unit in a greenhouse with bitter gourd," in *Intelligent Informatics and Biomedical Sciences (ICIIBMS), 2015 International Conference on*, 2015, pp. 109–112.

[21] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," *CISCO White Pap.*, vol. 1, no. 2011, pp. 1–11, 2011.

[22] A. Akbar, F. Carrez, K. Moessner, J. Sancho, and J. Rico, "Context-aware stream processing for distributed IoT applications," in *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, 2015, pp. 663–668.

[23] D. Puschmann, P. Barnaghi, and R. Tafazolli, "Using LDA to Uncover the Underlying Structures and Relations in Smart City Data Streams," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1755–1766, 2018.

[24] R. Lea and M. Blackstock, "City Hub: a cloud based IoT platform for Smart Cities," 2014.

[25] D. Nangare, J. Singh, V. Meena, B. Bhushan, and P. Bhatnagar, "Effect of green shade nets on yield and quality of tomato (Lycopersicon esculentum Mill) in semi-arid region of Punjab," *Asian J. Adv. Basic Appl. Sci.*, vol. 1, no. 1, pp. 1–8, 2015.

[26] P. Barnaghi, A. Sheth, and C. Henson, “From data to actionable knowledge: Big data challenges in the web of things [Guest Editors’ Introduction],” *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 6–11, 2013.

[27] F. Ganz, D. Puschmann, P. Barnaghi, and F. Carrez, “A practical evaluation of information processing and abstraction techniques for the internet of things,” *IEEE Internet Things J.*, vol. 2, no. 4, pp. 340–354, 2015.

[28] K. Yasumoto, H. Yamaguchi, and H. Shigeno, “Survey of real-time processing technologies of iot data streams,” *J. Inf. Process.*, vol. 24, no. 2, pp. 195–202, 2016.

[29] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *J. Mach. Learn. Res.*, vol. 3, no. Jan, pp. 993–1022, 2003.