IJCTA, 9(18) 2016, pp. 8953-8960

Using IoT, Machine Learning & Cloud for Predictive Pollution Analysis in a Smart City

Pranav Kamble*, Vivek Pawar* and Priya Singh*

ABSTRACT

When new age technologies like IoT, Machine learning, and Cloud computing are converged, we can have applications which have far more impact than these individual technologies can. In this paper we will see how we can use these 3 technologies to build a smart predictive pollution analytics systems which can be part of any modern day smart city.

Keywords: IoT, Cloud Computing, Machine Learning, Pollution, Smart City

I. INTRODUCTION

A Lot of advancements is happening in the fields of IoT, Machine learning and Cloud computing. Each of these fields is providing new opportunities to solve existing problem.

IoT has made it possible to take intelligent computing to devices making them smart in their functioning.

Machine learning has enabled to derive knowledge and intelligence from existing data from devices, systems. The learning is used to build intelligence in devices or systems.

Cloud computing has brought enormous computing infrastructure at click of button to today's applications. Today to build application of scale, new applications don't need to build infrastructure of their own. They can use the power of cloud with its IAAS, PAAS and SAAS. This makes it possible to quickly build applications of scale.

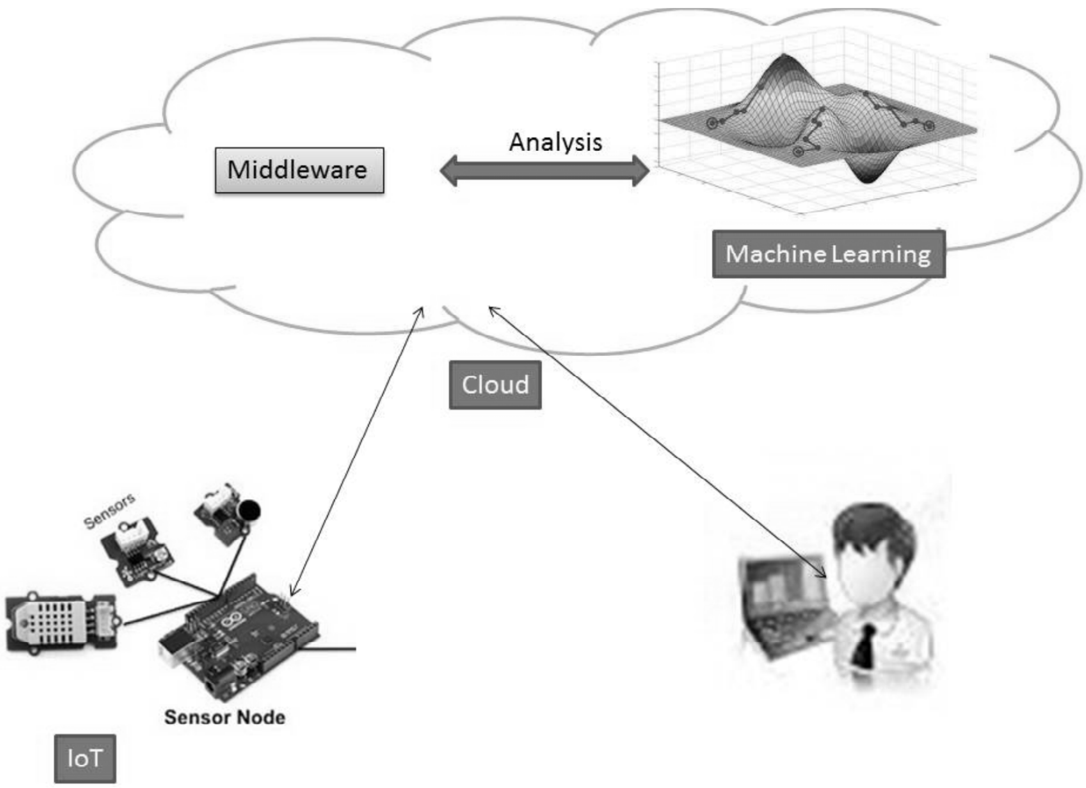
All this converges together to create more effective solutions. In this paper we present case of Air quality prediction system for smart city using these technologies. The demonstrated solution collects and publishes the air quality data in the real time using IoT. We use machine learning techniques to analyze historic data collected by IoT devices and predict future values for certain aspects data using cloud infrastructure and storage services.

Let's go ahead and get started.

II. PROPOSED SYSTEM

An IoT device with a controller and multiple sensors capable of measuring various gas levels like CO2, CO and O2 is used for collecting real-time data. Many such devices are placed all over city. Each device uploads the data of that area over cloud. Machine learning is applied to the collected data from the devices. Various statistical outputs about air quality/pollution is generated and shared with citizens/authorities. This enables citizens to understand condition of specific areas. The authorities can understand the whole city by analyzing the pattern across multiple areas and take corrective actions.

* Persistent University, Persistent Systems Ltd., *E-mails: pranav_kamble@persistent.com; vivek_pawar@persistent.com; priya_singh@persistent.com*



Top Level architecture diagram

2.1. Role of IoT

IoT has enabled to put computing power to small everyday devices. This has made it possible to program these devices and thus make them smart.

2.1.1. Devices

Various types of devices that can be programmed are

Sensors:

Numerous sensor devices are available in market. For e.g. Temperature, humidity, CO2, CO, Rain, pressure, Sound, height, Motion, Geo-location/GPS etc.

Motors:

Various motors which can be connected to smart computing boards are available. These motors allow programs to control their rotational speed, direction, time etc.

Switches:

These switches allow on/off operations to be controlled via programs.

Various Electromechanical devices like this in various industry segments like health , manufacture , safety, wearables , home automation are available in market now. The above devices themselves don't have computing power. However they can be attached to small computing boards which can be easily programmed.

2.1.2. Computing boards

The popular computing boards are from Arduino and Raspberry pi. These boards have a processor, storage and various ports to which other above devices can be connected. This are small in size, require less power, costs less and can provide basic computational power to devices to make them smart. Programming these boards is now pretty much advanced. We no more have to write assembly level code to ports to control devices. These board come with IDE in which we can write programs in high level languages like java , embedded C , Python , JavaScript. Also each of the devices comes with set of libraries to program them. Using this all together we can create a smart device connected to computing board and programmed using a high level language.

The hardware device in picture consists of an 'Arduino UNO' board which has 'ATmega328P' microcontroller. It shows various sensor devices that can be connected to the board. The board can be programmed to operate these devices.

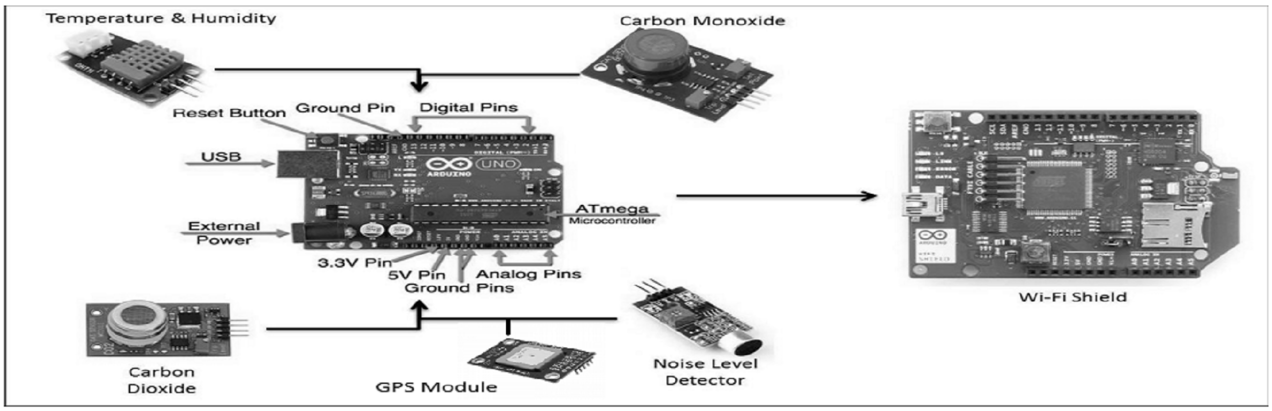


Figure 3.1: Arduino UNO board with different sensors. A 'Wi-Fi shield' for communication

2.1.3. Communication

Once we have the device connected to board we can program it control the devices. However standalone hardware system is of little use if its not connected to any backend systems. We need to collect the data the various sensor devices collect and store it somewhere .Then we need to apply machine learning on that data.

For this the hardware board needs to send data to backend system. Our backend system is hosted on cloud.

So the hardware system needs to send data to the cloud. For this it needs to have communication system to the internet. Physical connectivity to backend is typically provided using Bluetooth ,Wi-Fi , GSM

In our case we have used Wi-Fi. Using a Wi-Fi shield connected to Arduino board, the board can connect to a Wi-Fi network which can connect it to internet and in turn to cloud.

After ensuring link level connectivity to internet we need to decided on protocol to transfer data to internet .

We have following options for it HTTP, MQTT, CoAP, Custom

The standard way of sending data to internet ie HTTP has its own limitations. Because of that many new protocols have emerged. Of this MQTT is most popular. Let's see why we need to use MQTT over HTTP

2.1.3.1. MQTT vs HTTP

While designing, communication is the most crucial aspect to focus on. We want our solution to be quick in action. Delay in communication between the machines impact the solution to a great extent as the decisions is taken on the information exchanged amongst them.

A lightweight protocol, MQTT, was introduced to achieve the same. It works on the Publish/Subscribe (Pub-Sub) model in contrast to HTTP which works on the Request/Response model. Basically, MQTT is designed for low latency, assured messaging and efficient distribution . HTTP is not optimized for low power usage or minimizing the amount of bytes flowing .

The MQTT is more effective than HTTP for following scenarios

- connectivity is intermittent
- bandwidth is at a premium
- Hardware devices need to send data reliably without requiring code retry logic

Use of MQTT is always advisable for applications which are posting timely data with very low latency.

2.2. Role of Cloud Computing

2.2.1. Middleware PAAS

Considering peculiar nature of IoT data, a middleware application needs to be a powerhouse which drives IoT. Middleware application is responsible for routing, transformation and integration of the data. It is also responsible for the presentation logic and data storage. A middleware should be a complete package that caters to the requests, publishes the data, provides security, monitors hardware connections etc.

While developing an application that fulfills all these requirements, developer might end up developing an infrastructure. Maintaining such an application is a different job in itself! We can skip the time consuming task of developing a full-fledged middleware application and may avail the ready-to-use platforms provided by various cloud providers like Amazon AWS IOT and IBM Watson services. Cloud providers offer extensive platforms and services for application development. We can easily develop an application and integrate various services in it seamlessly. In our application we use IBM cloud.

2.2.2. Presentation of the data

This part of the solution is responsible for presenting the data in an interactive UI. The data or information is presented in the form of simple charts, graphs or tables. Again, this implementation differs according to the business needs. This layer of the solution may also be complimented with IBM services. IBM provides various templates to present the data in different forms.

2.2.3. Middleware application using IBM Watson

As discussed in the paper earlier, we make use of the pre-defined platform made available by IBM Watson for IoT application development. A step by step guide to create an application on IBM Watson is described below.

1. Signup on IBM Watson Bluemix services (Sign-in if already signed up)
2. Go to ‘catalog -> Boilerplates -> Internet of Things Starter boilerplate’. It provides ‘node.js’ sdk platform with NoSQL ‘Cloudant’ service for database. Include the name and organization and click ‘create’.
3. After creating an instance, again go to ‘catalog -> Internet of things -> IoT foundation cloud service’ to add a service which manages the IoT devices. Click on ‘add’ to bind this service to our application.
4. Launch dashboard of IoT foundation cloud service, add your Arduino UNO as registered device, you will get following information

   <Organization_ID>, <Device_Type>, <Device_ID>, <Authentication_Method>, <Authentication_Token> Add these details to your hardware code. This information is crucial as the device uses this information to connect and register itself to the IotF cloud service.

5. Return to ‘Dashboard’ and click on the newly created application.
6. Click on ‘route’. This will open ‘Node-RED flow editor’.

You can define the flow of data inside this editor. You can manipulate data coming from the IoT device, add it to the database, call APIs, handle HTTP requests etc. A broad spectrum of functions is offered in node-RED.

Node-RED flow editor

Node-RED provides a graphical UI to add nodes which have some predefined behavior e.g., ‘httpRequest’ node is for accepting http requests hitting a certain url. This greatly simplifies the programming. A node.js project gets updated each time we add any node or route on Node-RED. This gives the programmer the freedom to focus on the business logic than on the connections and routes. Nodes with different behaviors allow us to integrate many other services seamlessly without us having to worry about integration and configuration, which we have to carry out manually otherwise. This is a great feature of application development on cloud provided by IBM Watson!

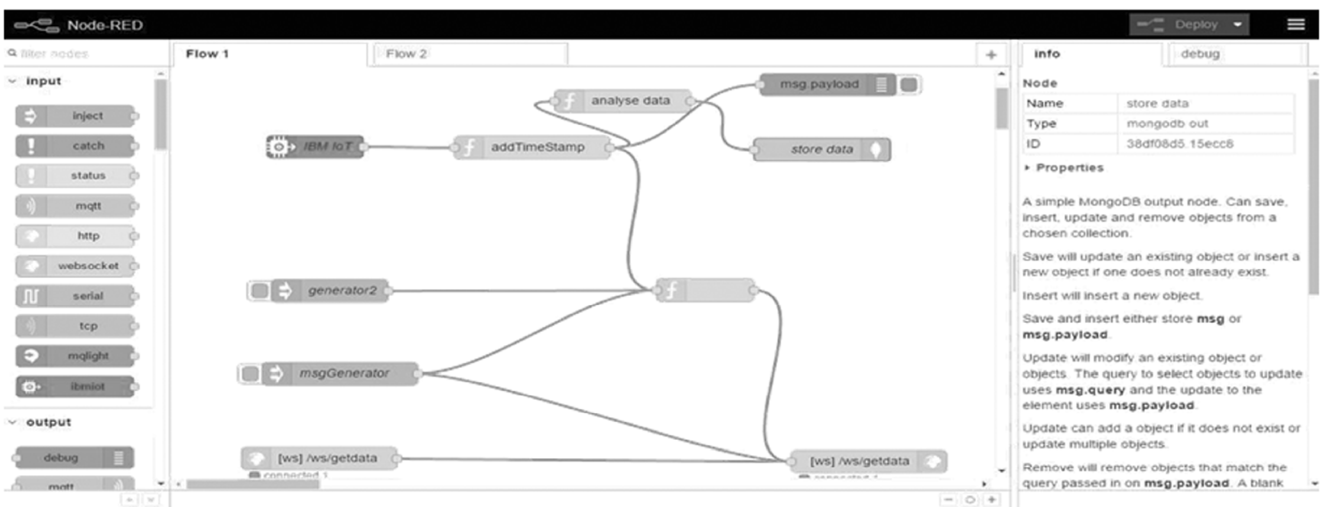


Figure 6.1: Sample view of Node-RED flow editor

2.2.4. Analytics in IBM Watson

IBM Watson offers wide range of services that can be integrated within an application. These are ready to use services and can be consumed either by adding nodes in Node-RED flow or by API calls.

One of the features of our solution is to examine the historical pollution trends and give the best time observed in the past days which has minimal CO_2, CO, temperature, humidity and noise level in any area. This feature has a very specific need of an analytics method which analyses numerous factors (minimum CO_2, CO etc.) from the huge amount of data collected. It should return the best possible time slabs in which pollution should be at minimum levels. This may seem easy but in reality dealing with so much of data is not an easy task. It takes a lot of time and space to apply logic on the data and extract information from it. ‘Tradeoff analytics’ deals with such scenarios and solves these problems by various mathematical and filtering techniques.

We can directly add ‘Tradeoff analytics’ node inside Node-RED flow editor. We can model our data according to the format specified by Tradeoff analytics and feed the same to ‘Tradeoff Analytics’ node. The service accepts and returns the data in JSON format. Exchange of data is made very easy and one can easily parse the JSON object from the response. The service can also be consumed using an API, however, a direct introduction of a node inside the flow makes it faster to exchange the data. It reduces the average turn-around time of the process.

Tradeoff Analytics

Tradeoff Analytics helps people to make better decisions by taking into account multiple, often conflicting, goals that matter when making that choice. In the Pollution monitoring and reporting system, we are making use of tradeoff analytics to get the best time to visit a particular area. We are providing Tradeoff service with a data averaged out in 1-hour slabs from past 7 days. The goal is to extract the time-slab having minimized CO2, CO, temperature and humidity with lowest possible noise level in an area. Similarly authorities may be interested in knowing the peaks, i.e. when pollution is at its highest level. When this data is combined with geo location information, authorities may order locations based on various parameters to make smart decisions.

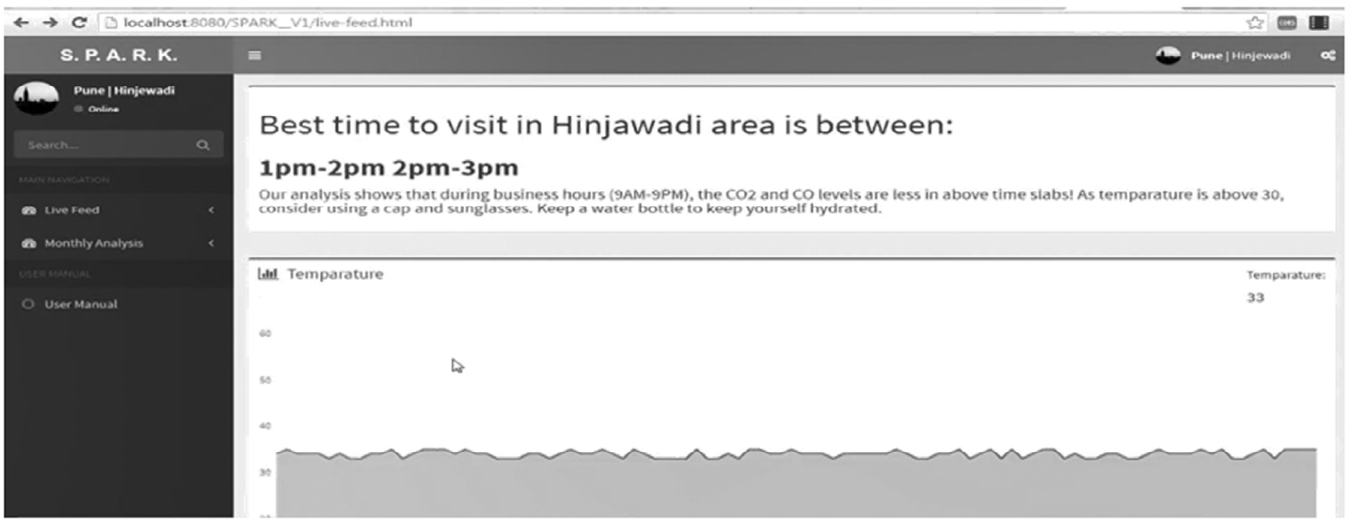


Figure 6.2: A snapshot from Pollution Monitoring and reporting System (S.P.A.R.K.)

2.3. Machine Learning: Predictive Analysis Usingpolynomial Regression

We used Tradeoff analytics to get the best time slab by analyzing historical data. However, this cannot be used for predicting the probable pollution levels in specific time slabs in the upcoming days. To do so, we make use of machine learning techniques. We implemented these techniques to predict probable pollution levels. From this we can also imply that, if the pollution and noise levels are too high, then there is a possibility of a traffic jam in that area.

The Pollution monitoring and reporting System stores timely pollution data for analysis. Predicting the pollution values is a supervised machine learning problem. This problem can be solved using techniques like regression algorithms, moving square etc.

Forecasting is a method or a technique for estimating future aspects of a things or the operation. It is a method for translating past data or experience into estimates of the future. Forecasts are important for short-term and long-term decisions.

There two broad categories of forecasting techniques:

- *Quantitative forecasting:* ***These methods are based on analysis of historical data and assuming that the past patterns in data can be used to forecast future data points.***
- Qualitative forecasting: ***These techniques employ the judgment of experts in specified field to generate forecasts.***

We used quantitative forecasting methods.

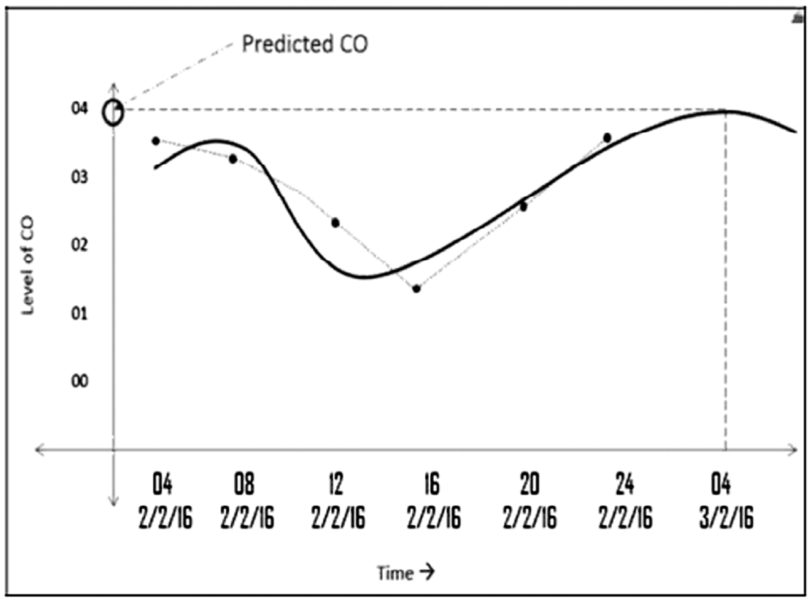
2.3.1. Regression Analysis

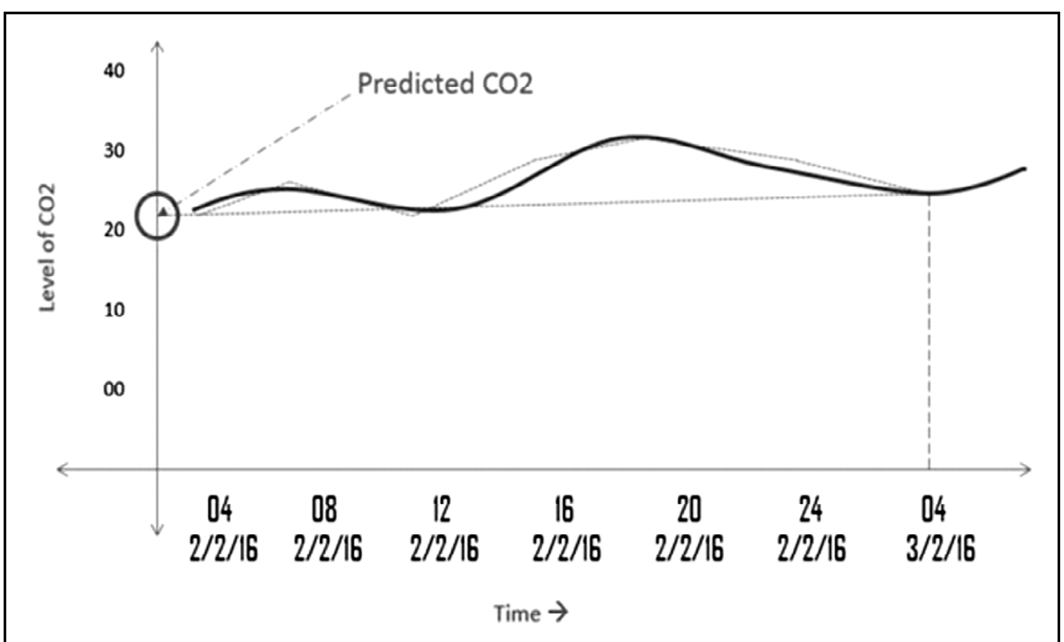
Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (predictor). This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables.

2.3.1.1. Polynomial Regression for Pollution Monitoring and Reporting System

Data sensed by the system cannot be fitted on a Straight line. We have fitted a curve for CO_2, CO with the help of polynomial regression analysis by using saved dataset. As polynomial regression model is trained, we use trained model to predict the levels of CO_2 and CO.

Bellow figures show the how model is fitted the curve and prediction of CO_2 and CO.





Above figure shows the fitting of curve for CO_2, CO and prediction of CO_2, CO. We have fitted the curve by using polynomial regression with past data about level of CO_2, CO sense for every four hours of day. Once model is trained (Curve is fitted) that model is used for prediction of level of CO_2, CO.

III. CONCLUSION

We have seen the technologies involved in building a IoT application and also the support provided by cloud platforms. We have seen how machine learning can be applied for predictive analytics. We have also seen how we can converge the advanced technologies of Cloud, IoT and Machine learning to have a effective solution. Many smart solutions are possible with this convergence. The opportunities are immense.

REFERENCES

[1] Kathleen Holm, “Using MQTT Protocol Advantages Over HTTP in Mobile Application Development,” 19 october 2012. [Online]. Available: *https://www.ibm.com/developerworks/community/blogs/sowhatfordevs/entry/using_mqtt_protocol_advantag es_over_http_in_mobile_application_development5?lang=en.*

[2] Ioannis N. Athanasiadis, Kostas D. Karatzas, and Pericles A. Mitkas,” Classification techniques for air quality forecasting”, ECAI workshop 2006.

[3] Niharika, Venkatadri M, and Padma S.Rao, “A survey on Air Quality forecasting Techniques”, IJCSIT, Vol. 5, No.1, pp 103-107, 2014.

[4] Tarun Rao, N Rajasekhar, Dr. T V Rajinikanth ,”An efficient approach for Weather forecasting using Support Vector Machines”, ICCTS vol. 47 2012.

[5] Gunita Yadav, and Dr. Nitin Mishra,”Air Pollution Trend Analysis Using Sen Estimator Method”, IJARCSSE, Vol. 5, Issue 7, July 2015.

[6] TH. Slini, K. Karatzas, and A. Papadopoulos,” REGRESSION ANALYSIS AND URBAN AIR QUALITY FORECASTING: AN APPLICATION FOR THE CITY OF ATHENS”, vol. 4, No 2-3, pp 152-162, 2002.