Chapter 13

Internet of Things and Analytics

**Abstract**

Internet of Things (IoT) has become one of recent trends in computing field as there is a flood of sensors and mobility in everyday life activities. Most of the wearables are used for tracking of different activities like running, walking and sleeping. Monitoring of such activities is possible through data analytics. Due to reasons such as wide adoption of mobiles, reduction in the cost of sensors has lead to the paradigm of IoT. The different components that are involved in IoT are computer networks, sensors, cloud computing and analytics. These components need to integrated together to develop an application for IoT systems. IoT has to be aligned to the data analytical platforms like Hadoop and Hive for developing applications. The different applications of IoT include smart grid, smart mobility, activity analysis and air-quality analysis. Activity analysis consists of analyzing the typical activities of a person like walking, running and sleeping. Air quality analysis involves analyzing the different components of the air such as CO2, NO2 and others for monitoring. In this chapter, firstly an overview of IoT and its components are highlighted followed by the case studies related to it. Two case studies namely air-quality analysis and activity analysis are discussed as a part of this chapter.

# Internet of Things and Analytics

In the past years, sensors were used for tracking data in retail stores, oil and gas industry. But, over the recent years a sudden hype is made on IoT. It is because, the sensors cost have drastically reduced and easily accessible. The bandwidth costs of internet usage have drastically come down where each individual in the world can access all the information in the smartphone. In this chapter, an introduction to Internet of things (IoT) and analytics is discussed.

## *Introduction to IoT*

IoT refers to the network of physical things that are connected to each other using the internet as the backbone for communication. It is the connectivity and the communication among the objects that form the word ‘Internet of Things’ or ‘IoT’. It is estimated that the number of things that are connected to each other will be atleast more than 20 billion by 2020. As the volume of the data increases, the main concern lies in the storage, analysis and making decisions based on the analysis. The scope of IoT is not only limited to this but it also involves concerns such as types of devices added to the network, types of data generated, scalability of the network etc [1] [2]. In this section, a brief overview of IoT and its components are discussed before analytics is discussed in the next sections.

The main aim of IoT is to gather information from various devices and different domains and offer service based on the data available. Everyday objects termed as ‘Things’ in IoT can offer services based on the data they collect by addition of capabilities such as assigning a virtual address space, self-organization and communication among the other devices. For improving the quality of services, additional capabilities such as awareness of the context, autonomous and others need to be incorporated. The services need to be offered without human intervention. RFID is one of the popular sensors used for identifying things, understand the context of the environment of things and take necessary actions. For example, an RFID can be used in a warehouse for a product tracking. It can be used to see if it has left the warehouse and automatically update the inventory in the warehouse database without the human intervention. Thus, sensors such as RFID transfer the physical entities into virtual address space for services. The application logic part of the device is then transformed into *smart environment* such as smart homes, smart city, smart grid, smart power etc. In this way, IoT helps to offer services with things communicating to each other.

Long back the computer cost was high when it was invented and took more space, an entire room to fit in and was also complex to use. But, due the advances in both hardware and software, computers can be carried ranging from homes, cars and offices. People around the globe are able to leverage the benefits of such advances in technology which are easy to use and less expensive. IoT helps in enabling the people to use more services from the advances in technology by connecting the existing machines and things to the cloud for streamline processing. The important question that will arise is why IoT now?

It is mainly because due to the advances in the hardware, IoT has become more popular now [2]. However, some of the key highlights that made IoT popular now days are as follows:

* **Innovation in Mobiles:** One of the main reasons for the adoption of IoT in many fields of computing and engineering is the innovations in mobile technology. People around the globe started migrating from the traditional desktop browsing to mobile browsing and no company could turn down this trend. So, it was a crucial development of migration of software services that run in the backend to adapt to the mobility and deliver the services dynamically. Though, it may be overstated or hyped in the digital world, it is still the critical innovations that lead to the adoption of IoT.
* **Open APIs:** Due to the proliferation of social media and other e-commerce platforms such as facebook, twitter, amazon, instagram, flickr etc, large number of APIs were used by these platforms. As the number of users grew in each one of these platforms, new innovations and creative ideas were needed to expand the usage of the platforms and thus APIs used by them were opened for the developers and integrate with their applications. Thus, open APIs played a major role in developing android based applications that require for integration of services into various other applications.
* **Ease of coding:** Earlier years of computing in the areas of embedded programming, mobile device application development needed experts from both the areas for developing an application. But, due to the advances in programming languages and scripting such as Python, Julia, R, node.js and others, it has now become easy for programmers to learn the programming language on their own and need not depend on others while developing an application.
* **Easy of availability:** Ten years ago IoT was used with the help of small motes which was expensive. But now, sensors are integrated into a single board where multiple sensors can be integrated on a single board and analysis can be carried out easily. The data services the needed to be used with cloud also used to be expensive compared to the current services available with Amazon EC2, S3 services on the cloud.
* **Rise of Data science:** Many areas of computing were addressed in diverse fields of engineering such as data science, cloud computing, machine learning, artificial intelligence and others. Due to the diverse interest in interdisciplinary engineering fields, IoT also came into the picture of main role and continued to be integrated into many other fields of computing.

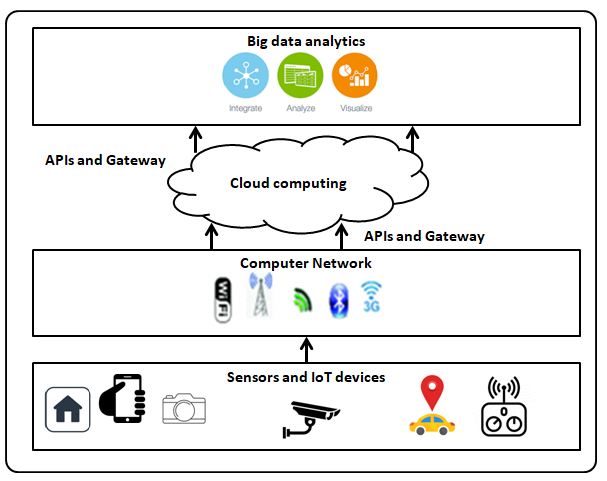
Even though, there are many other reasons of IoT coming into picture of current world the above key points highlight the important ones. The other challenges such as privacy, security, interoperability, user-friendly design coexist with IoT development. These challenges are research based and thus not encountered as a part of this book.

## *Components of IoT*

The architectural elements of IoT system depends on the domain and the applications involved. Some of the examples of IoT where architecture reference model is used for developing applications are smart home, smart traffic, smart health and smart transportation. Rather, than calling it as the architectural framework or the reference model, the components become a necessary part of IoT. In this section, the different components of IoT are discussed based on the figure 13.2.

The necessary components of IoT are sensors, cloud, networking and finally the analytics [3]. The basic IoT devices and sensors form the low-level component of IoT where various sensors such as camera, video surveillance, gps, smart home sensors are involved. These sensors communicate with each other with the help of computer networks such as Wifi, Bluetooth, 3G, 4G and other communication standards. The data gathered from the sensors are moved to the cloud further using the APIs and gateways. The APIs and gateways follow standard communication protocols such as TCP/IP for communication between the sensors and the cloud. From the cloud the different big data analytical applications extract the data and build various machine learning models for further analytics. The different kinds of dashboards and APIs are used in big data analytical applications for communicating with cloud.

Some of the big data analytical applications include smart home, smart grid, and smart transportation. For example, in the case of smart home application the data is gathered from various devices inside the home and stored in the cloud. The analytics application built at the top layer gathers all the data related to home from the cloud and then carry out analysis. Alerts and notifications are sent to the users if any necessary actions that need to be taken care of like switching off lights, fans. In this way, the different components of IoT help in carrying out analytics. Typically, the components specified in this section can be narrowed down further for domain-specific analytics in IoT systems.

****

**Fig. 13.2. Components of IoT**

## *Analytics and IoT*

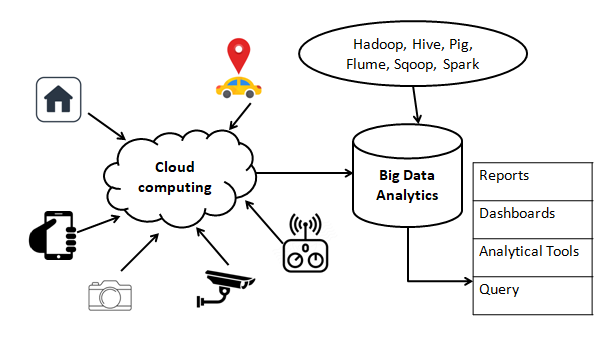
In the previous sections, an overview of IoT and its components were discussed. The data that is generated by the sensor devices needs to be analyzed and cannot be ignored. Since, it is estimated that 20 billion devices will be connected by 2020, the volume and variety of data to be analyzed will also become more [4]. Here, variety refers to the heterogeneity of the data coming from various sources of sensors in the IoT environment.

The main difference between IoT and analytics is the data sources for analysis. In the case of IoT the data sources can be sensors like camera, surveillance, RFID, temperature, humidity, Co2 etc. The data captured from these sources are analyzed to see the impacts like air quality (Co2 sensor), asset tracking (RFID and surveillance), smart home (electric bulb and switches sensors) and other areas. In the case big data analytical applications, the data sources are from social media (twitter, facebook, instagram), databases (customer data, inventory data and historical data) [5]. Even though the data sources are different in both of the areas, heterogeneity remains the same. The data formats are different and may need to be converted to a uniform format for analysis. In this section, a brief overview of different analytics that fall into the category of IoT systems are first discussed before diving into the actual case study of analytics in IoT.

### Big data analytics

Big data refers to the data that is characterized with volume, variety and velocity, generally referred to as 3 Vs. Analytics with such type of data requires advanced techniques rather than the traditional way of analysis i.e. migrating from the SQL based analysis to more complex analysis. Advanced types of analytics is required because of different types of data that are involved in Big data namely unstructured, semi-structured and structured data. For example, in order to do sentimental analysis on twitter data, access to tweets (unstructured), user information (structured) and tagging (semi-structured) are needed. The analytics needs to be carried out fast in real-time. Thus, IoT interconnecting with Big data analytics helps in fast analytics and the information can be shared among different platforms in a unified way [5] [6].

Both Big data analytics and IoT are inter-dependent on each other. As the number of the devices that get added into the IoT network, more are the chances and opportunities for big data analytics to be carried out. An interface of relationship between IoT and Big data analytics can be depicted as shown in the figure 13.3.1.



**Fig. 13.3.1. Relationship between IoT and Big data analytics**

As shown in figure 13.3.1., firstly the data is gathered from various devices and stored in the cloud. Since, the data is gathered from different sources and varies in formats, it needs to be converted into a uniform source for data analytics. The various data analytical methods that can be performed on this uniform dataset can be clustering, regression, classification etc. These methods are discussed in the earlier part of the book in Part 2.

### Real-time analytics

Earlier attempts in streaming data analytics suggested that such type of analytics can be carried out only on high performance computing systems. The primary techniques that are employed in streaming data analytics using distributed cloud platform are data parallelism and incremental processing. In data parallelism, the data is divided into smaller chunks are analytics is carried out on such parallel data. Incremental processing is batch analytics where the data is divided into a number of batches and analytics are carried out incrementally in a batch. IoT in streaming data analytics play an important role in bringing the devices that generate the data at faster rate to the cloud platform for analysis. For real-time analytics with IoT, incremental processing and data parallelism is less sensible as the data generated at the sources can be analyzed rapidly. Fog computing one of the recent advanced technologies is gaining impact in such analysis. However, with the integration of IoT with necessary storage and analytical platforms real-time streaming analytics can be formed.

### Bringing analytics to IoT

As discussed in the earlier sections of the chapter, IoT and analytics play an important role in developing smart solutions. IoT involves components like Arduino, Raspberry pie and hardware programming. The scope of this book is limited to analytics and its types and thus specific hardware implementations, sensors information are not included in this chapter. However, a case study on the dataset collected from various sensors for analyzing air quality is presented. Using this case study as the example, the readers of the book are suggested to carry out analytics using the machine learning techniques that are discussed in Part 2 of the book.

## *Air quality analysis*

Air pollution monitoring is gaining worldwide importance because of dense pollutants concentration in many areas. The sources of air pollution are due to the increase in vehicle population, fuel usage from the factories etc. Even though many programs are initiated for regulatory analysis of air quality by the government agencies, there is still lack of man power, instruments and cost viability. However, machine learning methods are helpful in analyzing such pollutants in the air for air quality monitoring.

Machine learning techniques are used for air quality monitoring and forecasting by using the earlier data of air quality. Most of the machine learning models used for air quality analysis use the historical data, and come up with a precise knowledge representation based on the relationship among the data. In this section, air quality analysis using regression modelling approach is discussed. The dataset is gathered from UCI machine learning repository [7]. Since, the scope of the book is limited to data analytics, any sensor or hardware programming to gather the data is not discussed as a part of this section. The main aim is to show how machine learning can be used as an integral part of IoT.

**Dataset information**

The data set consist of 9358 instances collected from an air quality sensor device with five metal oxide sensors. The device was located in an Italian city where areas were polluted for the period of 2004-2005. The attributes that are present in the data set are as follows:

* Date (DD/MM/YYYY)
* Time (HH.MM.SS)
* True hourly averaged concentration CO in mg/m^3 (reference analyzer)
* PT08.S1 (tin oxide) hourly averaged sensor response (nominally CO targeted)
* True hourly averaged overall Non Metanic HydroCarbons concentration in microg/m^3 (reference analyzer)
* True hourly averaged Benzene concentration in microg/m^3 (reference analyzer)
* PT08.S2 (titania) hourly averaged sensor response (nominally NMHC targeted)
* True hourly averaged NOx concentration in ppb (reference analyzer)
* PT08.S3 (tungsten oxide) hourly averaged sensor response (nominally NOx targeted)
* True hourly averaged NO2 concentration in microg/m^3 (reference analyzer)
* PT08.S4 (tungsten oxide) hourly averaged sensor response (nominally NO2 targeted)
* PT08.S5 (indium oxide) hourly averaged sensor response (nominally O3 targeted)
* Temperature in Â°C
* Relative Humidity (%)
* AH Absolute Humidity

### Normalize air quality data for regression

The dataset considered for the regression analysis of air quality data need to be normalized first. Normalization of the data is important because the scales in which the values of the data are present will be inconsistent. In this section, air quality data is first extracted from the UCI repository and normalized.

The data is first extracted from the repository and converted to a list. The different attributes of the data present are:

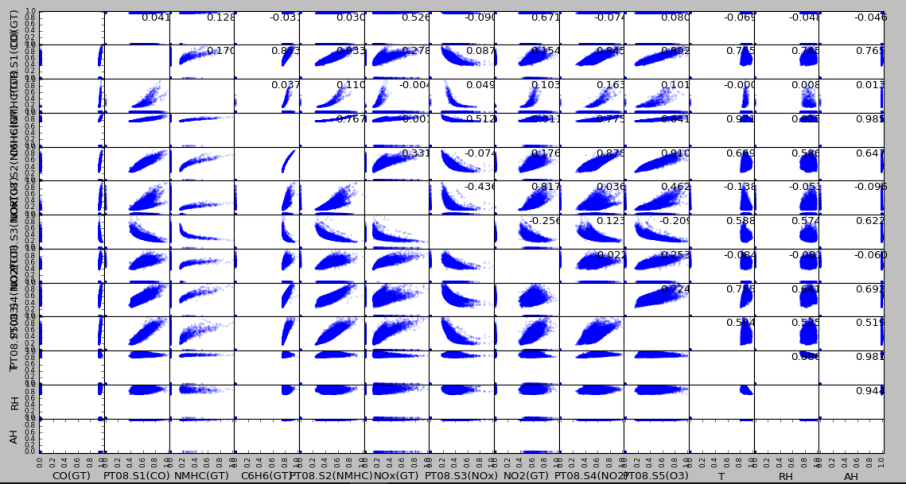
*‘CO(GT)’,'PT08.S1(CO)','NMHC(GT)','C6H6(GT)','PT08.S2(NMHC)','NOx(GT)','PT08.S3(NOx)','NO2(GT)','PT08.S4(NO2)','PT08.S5(O3)','T','RH','AH'*

These features of the data are used for the analysis of air-quality data. Initially, all the features all collected in a matrix where each entry in the matrix represents the feature value. The data is first converted to a data frame and the minimum and the maximum attribute values are obtained. The normalization of the values is carried out using the expression *value/(max-min).*

Since, the features of the data considered are in different range of values normalization is needed. Otherwise, the plot of the values will be scattered and not clearly visible. Thus, in this case the normalization of the values are important in this case of air quality analysis.

**import** pandas **as** pd  
**import** numpy **as** np  
**import** csv  
**import** matplotlib.pyplot **as** plt  
**from** pandas.plotting **import** scatter\_matrix  
**from** sklearn **import** preprocessing,model\_selection  
**from** sklearn.linear\_model **import** LinearRegression  
  
csv\_file=open(**'AirQualityUCI\_req.csv'**,**'r'**)  
data = list(csv.DictReader(csv\_file))  
  
attr\_list=[**'CO(GT)'**,**'PT08.S1(CO)'**,**'NMHC(GT)'**,**'C6H6(GT)'**,**'PT08.S2(NMHC)'**,**'NOx(GT)'**,**'PT08.S3(NOx)'**,**'NO2(GT)'**,**'PT08.S4(NO2)'**,**'PT08.S5(O3)'**,**'T'**,**'RH'**,**'AH'**]  
matrix=np.zeros([9357,len(attr\_list)])  
i=0  
j=0  
**for** item **in** data:  
 **for** attr **in** attr\_list:  
 matrix[i][j]=item[attr]  
 j=j+1  
 i=i+1  
 j=0  
dframe=pd.DataFrame(matrix,columns=attr\_list)  
n\_attr=len(attr\_list)  
min\_attr=np.zeros([n\_attr])  
max\_attr=np.zeros([n\_attr])  
attr\_values=np.zeros([13,9357])  
  
**for** i **in** range(13):  
 attr\_values[i]=dframe[attr\_list[i]]  
  
**for** i **in** range(n\_attr):  
 *#print(matrix[i])* min\_attr[i] = np.min(attr\_values[i])  
 max\_attr[i] = np.max(attr\_values[i])  
  
print(min\_attr)  
print(max\_attr)  
  
**for** i **in** range(len(attr\_values)):  
 attr\_values[i]=(attr\_values[i]-min\_attr[i])/(max\_attr[i]-min\_attr[i])  
*#print(attr\_values)*print(attr\_values.shape)  
attr\_values\_new=attr\_values.transpose()  
print(attr\_values\_new.shape)  
*#print(attr\_values)*df=pd.DataFrame(attr\_values\_new,columns=attr\_list)  
axes = scatter\_matrix(df, alpha=0.2, figsize=(45, 30),diagonal=**'histo'**)  
corr = df.corr().as\_matrix()  
**for** i, j **in** zip(\*plt.np.triu\_indices\_from(axes, k=1)):  
 axes[i, j].annotate(**"%.3f"** %corr[i,j], (0.8, 0.8), xycoords=**'axes fraction'**, ha=**'center'**, va=**'center'**)  
plt.show()  
  
To see the normalized values as a plot, the correlation of the data features is calculated initially. The *corr()* method is used on the data frame for obtaining the correlation among the features of data. The correlations obtained are used to plot the data set. A scatterplot of the normalization of the data is as shown in the figure 13.4.1.

It can be observed in the figure 13.4.1 where in each cell, the correlation among the features is displayed. For example, the correlation between 'PT08.S1(CO)' and other features is 94%, C6H6(GT) and other features is -0.04. The highest correlation values between the features is taken and those features are considered for the regression analysis.

****

**Fig. 13.4.1. Normalization for air quality data**

### Regression model for Air quality data

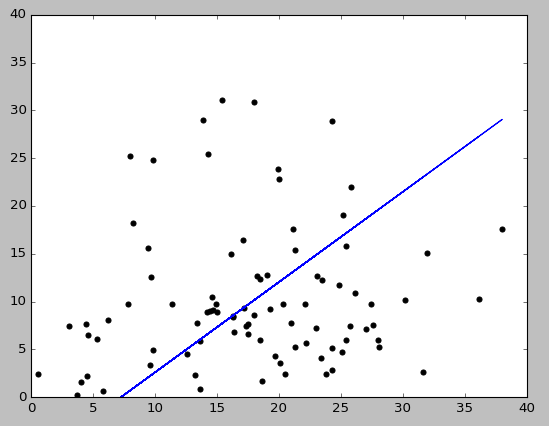
In this section, the regression modelling is done for the normalized air quality data. Initially, the following modules required for the regression model are listed as follows.

* matplotlib
* numpy
* pandas
* sklearn
* LinearRegression

The csv file is first read and converted into a list of values with the features as listed in the dataset described in the previous section. A matrix of values is created from the attribute list in the dataset. From the scatter plot shown in the figure 13.4.1, the features selected for the regression are T and C6H6(GT). These features are selected for the analysis because the correlation between them is 56%. Since, it is the highest correlation obtained in the scatter plot, these two data features are selected.

The regression model is first trained on the dataset by splitting the test and the training dataset. The fit() function is used to fit the model of the regression obtained using the model() function. A plot is obtained for the regression model obtained using the matplotlib module. The output of the plot is as shown in the figure 13.4.2. It can be observed from the figure 13.4.2 that the model obtained is fitting the values.

**import** matplotlib.pyplot **as** plt  
**import** numpy **as** np  
**import** csv  
**import** pandas **as** pd  
**from** sklearn **import** preprocessing,model\_selection  
**from** sklearn.linear\_model **import** LinearRegression  
  
csv\_file=open(**'AirQualityUCI\_req.csv'**,**'r'**)  
data = list(csv.DictReader(csv\_file))  
  
attr\_list=[**'CO(GT)'**,**'PT08.S1(CO)'**,**'NMHC(GT)'**,**'C6H6(GT)'**,**'PT08.S2(NMHC)'**,**'NOx(GT)'**,**'PT08.S3(NOx)'**,**'NO2(GT)'**,**'PT08.S4(NO2)'**,**'PT08.S5(O3)'**,**'T'**,**'RH'**,**'AH'**]  
matrix=np.zeros([9357,len(attr\_list)])  
print(data[1][**'CO(GT)'**])  
i=0  
j=0  
**try**:  
 **for** item **in** data:  
 **for** attr **in** attr\_list:  
  
 matrix[i][j]=float(item[attr])  
  
  
 j=j+1  
 i=i+1  
 j=0  
**except** Exception:  
 **pass**dframe=pd.DataFrame(matrix,columns=attr\_list)  
  
x=np.array(dframe[**'T'**].values.reshape(9357,1))  
y=np.array(dframe[**'C6H6(GT)'**].values.reshape(9357,1))  
  
x\_train,x\_test,y\_train,y\_test=model\_selection.train\_test\_split(x,y,test\_size=0.99)  
  
clf=LinearRegression()  
clf.fit(x\_train,y\_train)  
accuracy=clf.score(x\_test,y\_test)  
print(**"Accuracy: "**+str(accuracy))  
  
plt.scatter(x\_train,y\_train,color=**'black'**)  
pred=clf.predict(x\_train)  
plt.plot(x\_train,pred,color=**'blue'**)  
plt.xlim(0,40)  
plt.ylim(0, 40)  
plt.show()  
  
y\_axes=np.concatenate([np.array(i) **for** i **in** y\_train])  
y\_pred=np.concatenate([np.array(i) **for** i **in** pred])  
X=[]  
**for** i **in** range(len(y\_axes)):  
 X.append([y\_axes[i],y\_pred[i],y\_axes[i]-y\_pred[i]])  
print(X)



**Fig. 13.4.2. Air quality regression model**

In this section, air quality analysis was carried out using the regression model. The regression model was chosen the data analytical technique because forecasting of values can be done using it. However, initially the values of the data features are normalized so that consistent values are used for regression. Normalization need not be applied for all the cases of analytics in IoT. It is carried out only when needed. In the next section, another case study on activity analysis relating to IoT is discussed.

## *Activity Analysis*

Wearable devices in the world are increasing day-by-day because of IoT solutions getting smarter. Wearable devices are used for activities like running, walking, treadmill tracking, monitoring heart beat rate, blood pressure monitoring etc. The data generated from these devices can be used for analytics. For example, the data generated from the wearable devices can be used to analyze the number of kilometres covered during walking, running, jogging etc.

In this section a case study is discussed on the activity analytics. A random data set is created for the analytics purpose here. In reality, the wearable devices use sensors like gyroscope, accelerometer for fetching the data during the activities. In this case study, the random dataset assumed consists of measurements obtained by the gyroscope and accelerometer. These measurements are created on the basis of analysis of real-time values from the devices.

Activity analysis is carried out in two parts namely:

* Plotting activities
* Modelling activities

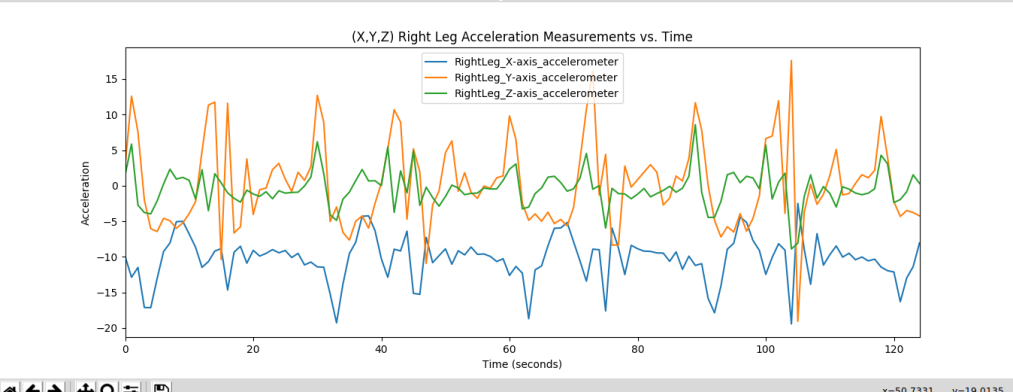
### Activity plots

In this section, the plots of the different activities are obtained using the matplotlib module. Initially, the data is read from the csv file and converted to a data frame. From the data frame, the accelerometer values along x-axis, y-axis and z-axis are obtained and plotted versus the time as shown in the figure 13.5.1.1.

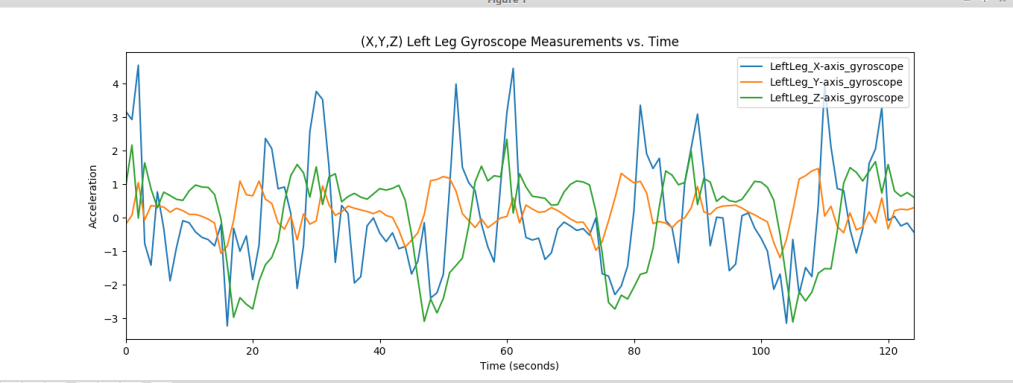
The measurements indicate that the graph indicates a walking pattern. The pattern of walking is obtained because of the accelerometer values obtained along all the axes. This is evident by the fact that the spacing between the peaks is about constant. If someone is walking at an irregular pace (i.e. slow-fast-slow progression) then there can be a change of frequency (more on frequncy later).

A similar plot is obtained as shown in the figure 13.5.1.2 for the gyroscope analysis. It can also be seen in the figure 13.5.1.2 that the peaks indicate that the person is walking. Since, the frequency is not changing much it can be inferred that activity is walking and not others. In this way, activity analysis can be carried out in the context of IoT.

**import** pandas **as** pd  
**import** seaborn **as** sns  
**import** matplotlib.pyplot **as** plt  
  
df = pd.read\_csv(**'activity-data.csv'**)  
print(df)  
  
ax = \  
df[[**"RightLeg\_X-axis\_accelerometer"**, **"RightLeg\_Y-axis\_accelerometer"**, **"RightLeg\_Z-axis\_accelerometer"**]].plot(title = **"(X,Y,Z) Right Leg Acceleration Measurements vs. Time"**,  
 figsize=(16,5));  
  
ax.set\_xlabel(**"Time (seconds)"**)  
ax.set\_ylabel(**"Acceleration"**);  
plt.show()

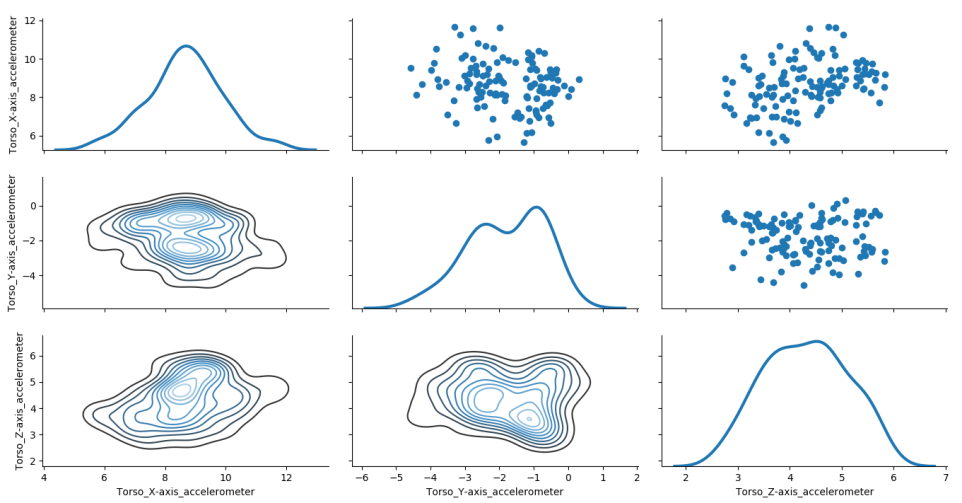
**Fig. 13.5.1.1. Log acceleration in activity analysis**

ax = \  
df[[**"LeftLeg\_X-axis\_gyroscope"**, **"LeftLeg\_Y-axis\_gyroscope"**, **"LeftLeg\_Z-axis\_gyroscope"**]].plot(title = **"(X,Y,Z) Left Leg Gyroscope Measurements vs. Time"**,  
 figsize=(15,5));  
ax.set\_xlabel(**"Time (seconds)"**)  
ax.set\_ylabel(**"Acceleration"**);  
plt.show()



**Fig. 13.5.1.2. Gyroscope measurements in activity analysis**

g = sns.PairGrid(df[[**"Torso\_X-axis\_accelerometer"**, **"Torso\_Y-axis\_accelerometer"**, **"Torso\_Z-axis\_accelerometer"**]],size =2.5, aspect=2.0)  
g.map\_upper(plt.scatter)  
g.map\_lower(sns.kdeplot, cmap=**"Blues\_d"**)  
g.map\_diag(sns.kdeplot, lw=3, legend=**False**);  
plt.show()



**Fig. 13.5.1.3. Conditional probability estimation in activity analysis**

The figure 13.5.1.3 shows a pair of grid plot for the conditional probabilities along the X,Y,Z dimensions of the person's acceleration. It shows the correlation with each other. It can be seen that the distributions are centered close to each other in the. The top triangle shows the conditional relationship between the dimensions as a scatter plot. Since the signals are approximately normal, these features can be used in feature modeling further.

### Activity count

In the previous section, analytics was carried out only from the perspective of different activities. But now, in this section the number of times the activities are done is calculated. For example if it walking how many steps were taken from the user is counted. To accomplish this, initially the data frame is loaded and the different features are obtained.

The different features obtained here are leg accelerometer, gyroscope analysis and torso acceleration. For each of these feature names, the data is split into different sections first such as *data\_home* and *user\_home*. These section of the data are analyzed one by one using the split() function. Using this function, the length of each data is computed that gives the number of times each activity has occurred.

**import** os  
**import** csv  
**import** re  
  
**def** generate\_feature\_names():  
 *'''Creates feature names for dataframe header'''* feat\_names = []  
 **for** unit\_label **in** [**"Torso"**, **"RightArm"**, **"LeftArm"**, **"RightLeg"**, **"LeftLeg"**]:  
 **for** sensor **in** [**"accelerometer"**,**"gyroscope"**,**"magnetometer"**]:  
 **for** position **in** [**'X'**,**'Y'**,**'Z'**]:  
 feat\_names.append(unit\_label + **"\_"** + position +**'-axis\_'**+ sensor)  
 **return** feat\_names  
  
**def** load\_segment\_names(home, data):  
 *'''Loads activity data for a specificed subset'''* **return** [filename **for** filename **in** os.listdir(home + data)]  
  
feat\_names = generate\_feature\_names()  
print(feat\_names)  
  
data\_home = **"/…/data/"**user\_data = **"a09/p7/"***# load data for a single user that is walking in a parking lot*file\_names = load\_segment\_names(data\_home, user\_data)  
walk\_file = data\_home + user\_data + file\_names[0]  
*#df = pd.read\_csv(walk\_file, names = feat\_names)*csv\_file\_orig=open(walk\_file,**'r'**)  
data=csv\_file\_orig.read()  
*#print(data)*cs\_data=re.split(**',|\n'**,data)  
print(cs\_data)  
print(len(cs\_data))  
**for** i **in** range(len(cs\_data)):  
 **try**:  
 cs\_data[i]=float(cs\_data[i])  
 **except**:  
 print(cs\_data[i])  
print(len(cs\_data))  
print(cs\_data)  
print(feat\_names)  
  
  
csv\_file=open(**"activity-data.csv"**,**'w'**)  
writer = csv.writer(csv\_file, delimiter=**','**)  
writer.writerow(feat\_names)  
print(len(feat\_names))  
  
i=0  
**while** i<len(cs\_data):  
 writer.writerow(cs\_data[i:i+45])  
 i=i+45

# Exercises

1. Describe the different components of IoT.
2. Create a chart of data sources for different IoT applications.
3. For the air quality analysis, experiment the regression analysis without normalization.
4. Create a chart of sensors and its bandwidth for the activity analysis case study.
5. Describe the different types of analytics that can be carried out in an IoT system.

# References

1. Kopetz, H. (2011). Internet of things. In *Real-time systems* (pp. 307-323). Springer, Boston, MA.
2. Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer networks*, *54*(15), 2787-2805.
3. Kortuem, G., Kawsar, F., Sundramoorthy, V., & Fitton, D. (2010). Smart objects as building blocks for the internet of things. *IEEE Internet Computing*, *14*(1), 44-51.
4. Da Xu, L., He, W., & Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, *10*(4), 2233-2243.
5. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, *29*(7), 1645-1660.
6. Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012, December). Future internet: the internet of things architecture, possible applications and key challenges. In *Frontiers of Information Technology (FIT), 2012 10th International Conference on* (pp. 257-260). IEEE.
7. Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml