Modeled Sensor Database for IOT Aiming Efficient Data Storage

Bhawna Sharma[1*], Sheetal Gandotra[2], Romika Manhas[3], Mehreen Choudhary[4], Nitika Gupta[5]

[1]Associate Professor, [2] Associate Professor, [3]BE Student, [4] BE Student, [5] BE Student

Department of Computer Engineering, Government College of Engineering and Technology, Jammu, J&K, India.

Abstract – The Internet of Things (IoT) is becoming ubiquitous in our everyday lives, implying that more technologies will generate data. Various environmental attributes such as temperature, humidity, light, etc. are monitored using IOT sensors. These sensors produce data periodically and storing this huge amount of data in a database is becoming a huge challenge in the data storage infrastructure. Prior research has proposed compression algorithms and signature techniques to reduce the data storage but they do not specify how the data patterns are defined. Since daily sensing of the environment exhibits similar patterns, data generated is highly redundant .Therefore, this paper, proposes a low storage system that stores data models rather than storing raw data points. Polynomial models are developed that fit a sample data set and then store these data models with the corresponding time periods that captures the behavior of the sensor data. Different polynomial equations are used to generate the data models. Two different strategies are implemented to use these polynomial equations for obtaining data models aiming more accuracy and better computation speed. Any desired value can be obtained using these models.

Index Words – Internet of things (IoT), sensors, database, data models, wireless sensor network.

I. INTRODUCTION

Internet of Things (IoT) where 'things' i.e. sensors and devices transmit data directly to the Internet has become an enabling technology eco system with several application areas like Smart Home, Smart Farming, Smart Grid, Industrial Internet, Connected Health, Smart Supply Chain etc. Current innovations in technology focuses on controlling and monitoring of different tasks. These are increasingly emerging to reach the human needs. Most of this technology is focused on efficiently supervising and controlling different activities. An efficient environmental monitoring system is required to monitor and assess the conditions in case of exceeding the prescribed level of parameters (e.g., noise, CO levels). The data obtained from different sensors is heterogeneous [10]. Also, the sensor data are redundant as the environment that the sensors are monitoring

Generally doesn’t change for a short period of time and hence, same data is produced by the sensors [12]. Sensors collect data over time and render real-time data. The real time data when processed can bring profit to IoT. Thus, when an unanticipated event occurs, a timely exception detection can be helpful to make decisions to decrease and avoid the loss of data. Moreover, as the production cost of sensors is very low, large number of sensors are installed in the environment to get complete statistics about the surroundings. Also, the sensors are collecting data at frequent intervals resulting in large amount of data generation. Therefore, sensors obtain huge amount of heterogeneous, continuously streaming and geographically dispersed real time data. This results in the issue of communication overhead and database will be heavily loaded, growing very fast and performance will drop if all of this data is stored.

Hence, there is a need to find a way to efficiently store this massive volume of data. In addition to the data generated by the IoT objects, there is metadata that defines and describes these objects, such as object identification, location, services provided, etc. Also, IoT data, unlike the traditional Internet data, this data possess the time and space attributes that provides the dynamic state changes of an objects location over time. Therefore, communication, storage and process will play an important role in designing the data management solutions for IoT. IoT data storage mainly have three schemes: local, distributed, and centralized. In the local scheme of data storage, each sensor has its own database unit. Thus, the sensor nodes in the wireless sensor network (WSN) runs its own database management system. In the distributed systems, data is stored in some sensor nodes in WSN and this is done using distributed technologies [11]. Intermediate tools are used to provide data access. In the centralized scheme, the data of the network is collected by a node, then sent to a data center where all the data of the network is stored. Since the storage capacity and battery power of the sensors is very limited, the local and distributed form of data storage is not suitable for IoT systems as IoT produce huge data. Which indicates that the centralized form of data storage is more suitable for IoT applications. Most of the data generated from the sensors is redundant and is still being stored as we do not have any system yet that can detect the data and avoid redundancy. Therefore, this paper proposes a model based data storage design solution for IoT applications that can resolve the problem of huge scale and complex IoT data.

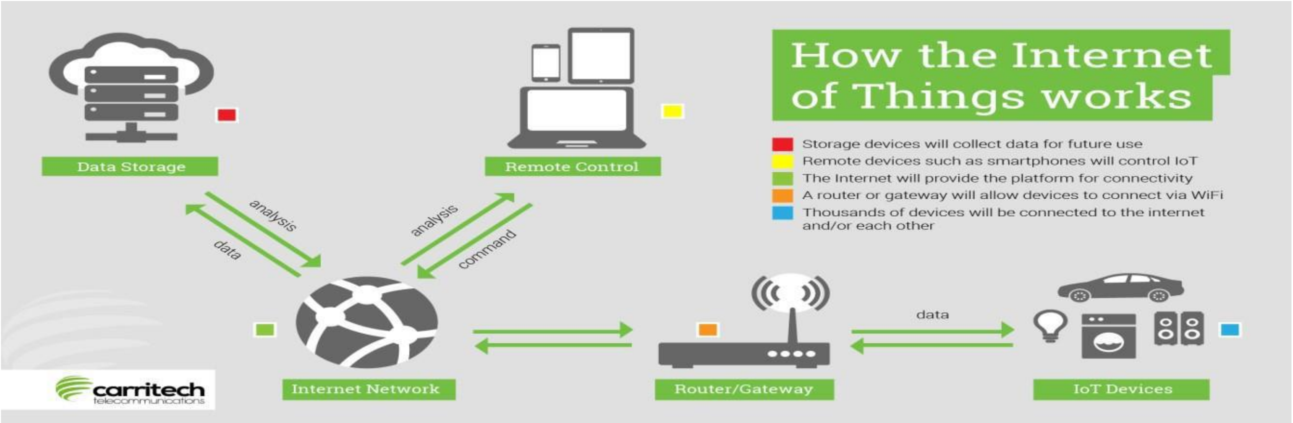

How the Internet of Things works
🟥 Storage devices will collect data for future use
🟨 Remote devices such as smartphones will control IoT
🟩 The Internet will provide the platform for connectivity
🟧 A router or gateway will allow devices to connect via WiFi
🟦 Thousands of devices will be connected to the internet and/or each other
Data Storage
Remote Control
analysis
data
analysis
command
Internet Network
Router/Gateway
data
IoT Devices
carritech telecommunications

Fig 1: IoT Environment

II. SYSTEM OVERVIEW

The basic components of the system are as follows:

1. Sensor Network
2. Sink Node or Aggregate Node
3. Gateway
4. Internet
5. Database System

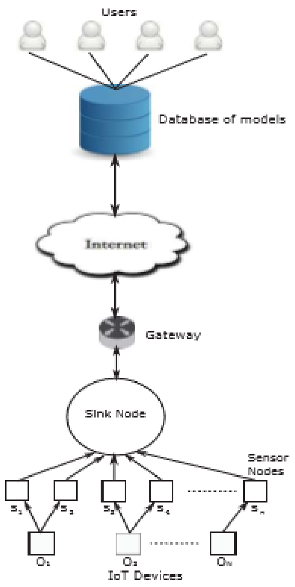


Fig 2: A typical IoT database

The data collected from various devices in sensor network are communicated to the end user via internet. The data communication between sensors and internet is done via gateway. The gateway nodes convert raw data to standardized format. For reducing raw data in an efficient manner, a model-based scheme can be well-established that can predict the data and answer queries of the user just by selecting the correct model for the data instead of accessing a database that stores all the processed data.

III.DATA MODELS USED

In the aspect of avoiding redundant data generated from the sensors, data models are created using polynomials while the sensor node is providing new samples [1]. When adding points to the polynomial, the algorithm tries to add as many points to the polynomial by adding degrees to the polynomial to fit the data. If the data does not fit, the polynomial keeps adding degrees until the maximum number of degrees is added. If the value point does not fit within the maximum number of polynomial degrees, the polynomial is stored with the timestamp, then, a new polynomial of degree zero is created to fit the next sample, and the process is restarted. This algorithm is an online segment construction based on live machine learning. Categories for model elements are created if an attribute in the IoT object has changed, but this research has not reused their models for future time intervals. Rather, they create new polynomials for each time interval unlike this project that tries to predict future trends to decrease sensor traffic.

A Time Series database service has been created to support pattern searches in the IoT domain [2]. For query processing, TSaaS (Time Series as a Service) partitions the pattern into segments, and searches for a similar pattern using a systematic technique.

Comparison of representative techniques like piecewise constant approximation, adaptive piecewise constant approximation, slide filters, from categories based on constant, linear, nonlinear and correlation models according to data reduction and prediction accuracy[The study concludes that constant and linear models outperform the others in the presence of small variations in the data. As compared to these results, the data models created in this work are based on linear and higher degree polynomials.

There exists a technique that sends and receives information using signatures instead of raw sensor data [4]. The signature is a string representation implying that something is present or absent at a particular sensor. The problem of data communication is overcome because only binary data is transmitted; but how the data patterns are defined is not specific enough. In our application, the sensor readings are understood and analyzed with the proposed model-based method that may learn patterns out of it and further forecast the real time data.

IV. SENSOR DATA MODELS

A. Generating Data Models

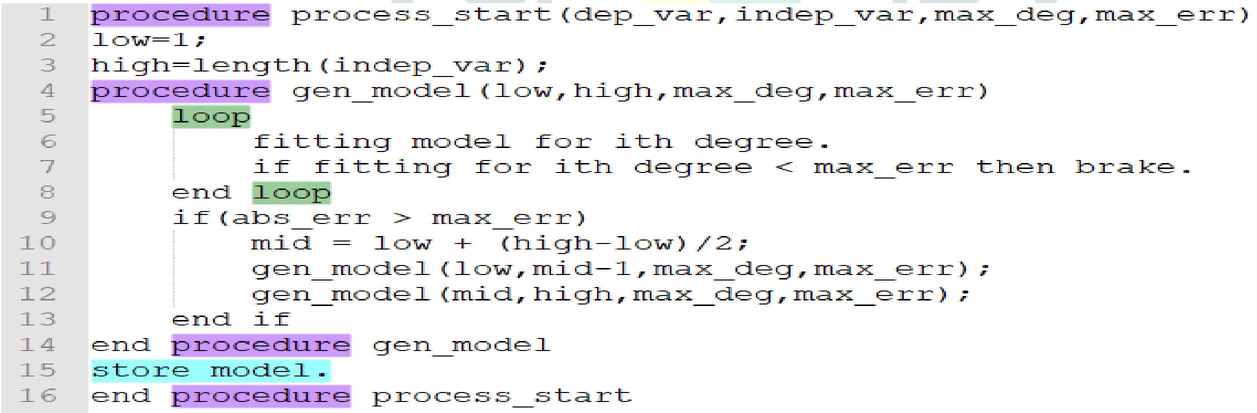
The data models are produced from a set of raw data points, and they are stored in the database against a time interval - this data model is considered an effective depiction of raw data points that were observed at timestamps which lie within this time interval. The mathematical models, M1, M2, M3,

MN, are polynomial equations that are stored in the database in the form of numeric coefficients of the

Equation with the corresponding time periods T1, T2, T3, TN. Let us denote these coefficients as un, un-1, un-2, and u0 where n is the degree of the polynomial equation. The coefficients of a polynomial is found that fits the raw data points by minimizing the mean error of these data points from the value given by the model. The models will be stored in the form of their polynomial coefficients as they will be mathematical functions of the form as shown in Equation 4.1, where the data value P at an input time period x is calculated by providing the coefficients un, un-1, un-2, ..., u0 of the mathematical polynomial model of degree n [6].

$$P_1 = u_0 + u_1x + u_2x^2 + \ldots + u_nx^2 \qquad (4.1)$$

Algorithm 1 Generation of Sensor Data Model



```
procedure process_start(dep_var,indep_var,max_deg,max_err)
low=1;
high=length(indep_var);
procedure gen_model(low,high,max_deg,max_err)
    loop
        fitting model for ith degree.
        if fitting for ith degree < max_err then brake.
    end loop
    if(abs_err > max_err)
        mid = low + (high-low)/2;
        gen_model(low,mid-1,max_deg,max_err);
        gen_model(mid,high,max_deg,max_err);
    end if
end procedure gen_model
store model.
end procedure process_start
```

Generation of Sensor Data Model (Algorithm 1) describes a general algorithm to generate data models for the real world sensor data. The procedure *process_start* takes four arguments: a dependent variable, *dep_var*, which could be any sensor parameter like temperature, pressure, humidity, etc., an independent variable, *indep_var*, which is the time, maximum degree, *max_deg*, which is the degree up to which the models are stored, and the maximum error, *max_err*, which is the maximum error that can be obtained in predicted values using this algorithm. This procedure initializes two variables, *low* and *high*, as *low* = 1 and *high* = the length of the *indep_var*. It contains another procedure *gen_model*, which takes four arguments: *low*, *high*, *max_deg* and *max_err*. In this procedure a loop is generated which tries to fit a polynomial model to the data point stating from degree one up to the *max_deg* while keeping in concern that the mean error of the particular model, *abs_err*, is less than the *max_err*. If *abs_err* comes out be greater than *max_err*, then the data set is divided into two halves (binary division) and apply the same procedure to these two halves [6]. In this way models can be fitted to the data points until the desired polynomial models is achieved. And finally these models are stored in a spreadsheet with their corresponding time stamps and use them to predict any value needed at any time.

B. DATA MODELS AIMING MORE ACCURACY

In the previous section, the general polynomial equation was used (Equation 4.1) to generate data models for the raw data points. But if more accuracy is required in the predicted data values using the generated data models, more equations can be used along with the existing general polynomial equation. Also if the data collected bears a complex relationship with the time and makes it difficult to predict accurate values then the following bivariate polynomial equations can be used to generate more accurate data models.

$$P_2 = u_0 + u_1x + u_2y + u_3xy \qquad (4.2)$$
$$P_3 = u_0 + u_1x + u_2y + u_3x^2 + u_4y^2 + u_5xy \qquad (4.3)$$

In the above equations x is the independent variable, time, whereas y is the data points of the model which is generated by using the Equation 4.1. Again if the accuracy is not up to the mark then the below mentioned three variable equation is used to generated data models

$$P_4 = u_0 + u_1x + u_2y + u_3z \qquad (4.4)$$

In the above equation, x is again the independent variable, time, y is the data points of the model which is generated by using the Equation 4.1, and z is the data points of the model which is generated by using the Equation 4.2 or Equation 4.3. In this way all these polynomial equations are used to create more accurate and precise data models for the given data set. Even if the data points in the data set are very closely placed then also these equations are very helpful to produce efficient data models for the data set.

V. DIFFERENT STRATEGIES

There are two different strategies to work upon in order to generate data models. These are:

- Accuracy oriented
- Performance oriented

In Accuracy oriented strategy, our focus is only to generate models as accurate as possible. In this technique, less importance is given to the processing parameters involved. All the equations can be exploited as many time as needed in order to obtain minimum mean error and thus getting highly accurate predicted values.

In Performance oriented strategy, our concern, along with producing data models, is also on the processing power and time. The goal of this strategy is to use the algorithm and the polynomial equations in such a way that the system takes minimum time to process it and uses less computation to generate models thus making it an efficient technique in terms of processing parameters.

Both these techniques uses the same polynomial equations as declared preciously. But how they use it, is the main point of difference between them.

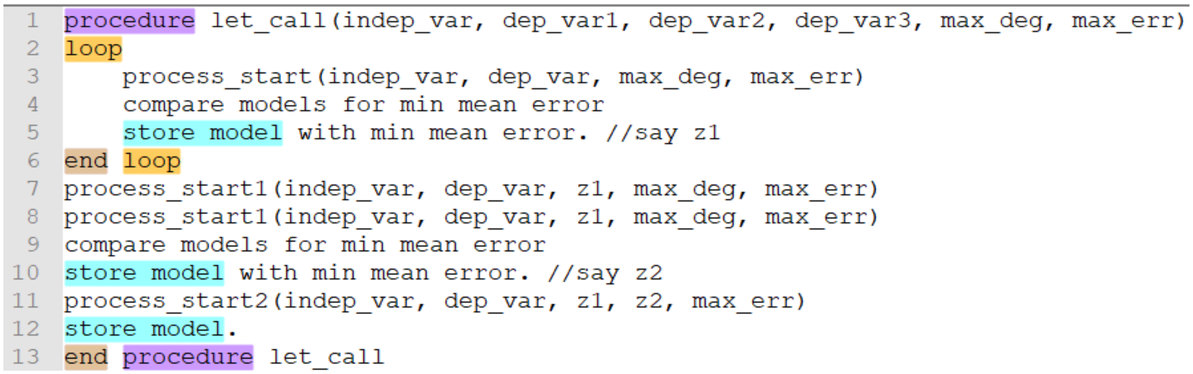
Accuracy oriented strategy contains number of comparisons of these equation for each dependent variable to choose the best case with minimum error. Whereas in performance oriented strategy, any combinations is chosen without doing any comparisons. Since it has no comparisons between different models and equations, its processing time is less and is more efficient computation wise.

A. Executing Accuracy Oriented Strategy

In accuracy oriented technique, the data models are required to be as accurate as possible. So, first polynomial models are created for all independent variables using Equation 3.1 and then compare their respective mean errors. Then the model which has the minimum mean error is chosen and is used along with dependent variable to generate accurate models for other independent variables using the bivariate polynomial equations (Equations 4.2, 4.3). Again the model with minimum mean error is selected and is used along with previous selected variables to generate model for the remaining independent variable using three variable polynomial equation (Equation 4.4). In this way more accurate polynomial models can be generated for the given data.

Now to execute this strategy, a sample sensor dataset is used which contains readings of temperature, pressure and humidity for respective time slots. Here, there are three dependent variables (temperature, pressure and humidity) and one independent variable (time). Following algorithm is used to execute accuracy oriented strategy.

Algorithm 2 Implementing accuracy oriented strategy



```
procedure let_call(indep_var, dep_var1, dep_var2, dep_var3, max_deg, max_err)
loop
    process_start(indep_var, dep_var, max_deg, max_err)
    compare models for min mean error
    store model with min mean error. //say z1
end loop
process_start1(indep_var, dep_var, z1, max_deg, max_err)
process_start1(indep_var, dep_var, z1, max_deg, max_err)
compare models for min mean error
store model with min mean error. //say z2
process_start2(indep_var, dep_var, z1, z2, max_err)
store model.
end procedure let_call
```

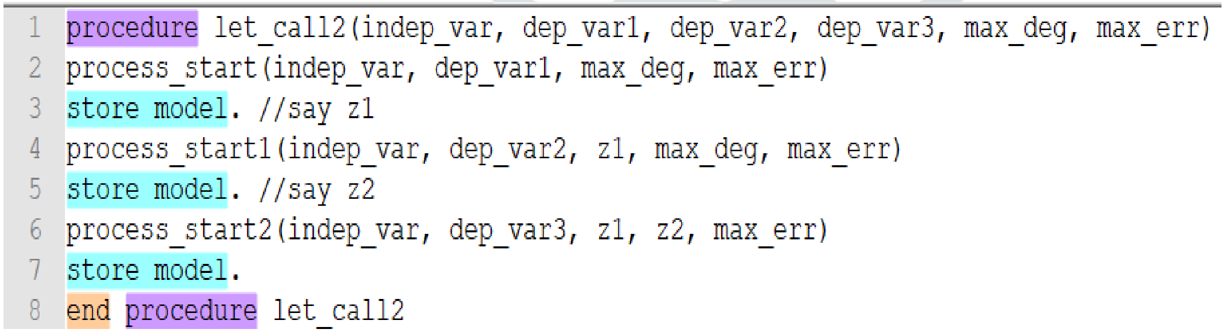
In the above algorithm's procedure *let_call*, time is considered as *indep_var*, pressure as *dep_var1*, humidity as *dep_var2*, temperature as *dep_var3*, 2 as *max_deg* and 0.1 as *max_err*. In this procedure, there is a loop which iterates for three time. This loop contains call to procedure *process_start* which generates model for all the three dependent variables and compare the respective mean error of each model. The one with the minimum mean error is selected, stored and is designated by z1. Then outside the loop procedure *process_start1* is called with time as *indep_var* and one of the remaining two independent variables.

Again *process_start1* is called but this time with the other remaining independent variable. This procedure uses the two variables equations to generated models. Both these models' mean error is compared and the one with minimum mean error is selected, stored and is designated by z2. Now procedure *process_start2* is called with the remaining independent variable. This model uses three variable equation to generate polynomial model. The model thus obtained is stored for the used independent variable.

B. Implementing Performance Oriented Strategy

In this performance oriented strategy, our focus is to generate data models for the independent variables in the fastest way i.e. in a way that uses less computation power and less time. Here, Models for different variables are directly generated without comparing their respective mean errors. So, take the first independent variable and generate its data models using the basic polynomial equation (Equation 4.1) and store it. Then use the two variable polynomial equations (Equation 4.2, 4.3) to generate data models for the second independent variable. And then use the three variable equation (Equation 4.4) to generate data models for the third independent variable. In this way, models are generated without doing comparisons and decreasing the time needed for computation.
Following is the algorithm used to execute this strategy.

Algorithm 3 Implementing performance oriented strategy



```
procedure let_call2(indep_var, dep_var1, dep_var2, dep_var3, max_deg, max_err)
process_start(indep_var, dep_var1, max_deg, max_err)
store model. //say z1
process_start1(indep_var, dep_var2, z1, max_deg, max_err)
store model. //say z2
process_start2(indep_var, dep_var3, z1, z2, max_err)
store model.
end procedure let_call2
```

In the above algorithm's procedure *let_call2*, supply time as *indep_var*, temperature as *dep_var1*, humidity as *dep_var2,* pressure as *dep_var3*, 2 as *max_deg* and 0.1 as *max_err*. In this procedure, first call the procedure *process_start* which time as *indep_var* and temperature as *dep_var1* and uses the general polynomial equation to generated models for this independent variable. Store this model and designate it with z1 for further References.

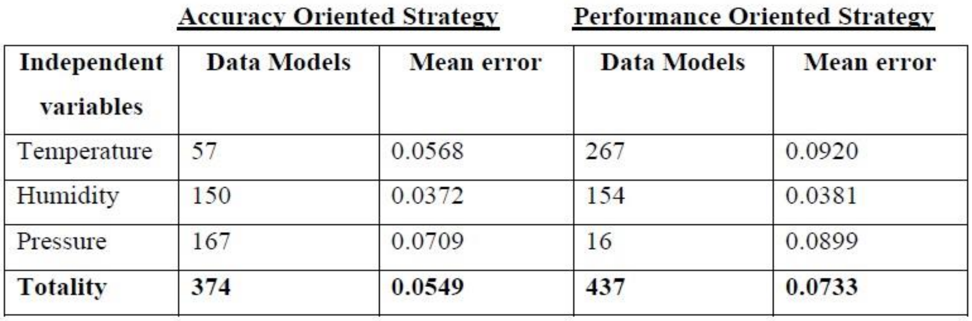
Then call procedure *process_start1* which takes humidity as the *dep_var2* and generate models for it using the two variable (time, temperature) equations. Store this model and designate it with z2 for further references. Now call procedure *process_start2* which takes pressure as *dep_var3* and generate data models for it using the three variable (time, temperature, humidity) polynomial equations. Finally store this this model.

VI. COMPARING THE TWO STRATEGIES

Although the main purpose of these two strategies is to generate the data models by using different polynomial equations for the sensor data set points but how they use these equations is the main point of difference between them. The accuracy oriented strategy uses these equations number of times to achieve the maximum accuracy whereas the performance oriented strategy uses these equations only once as its focus is on to use less time and less computation. Now on applying the accuracy oriented strategy on a sample sensor data set which contains 1440 points. These points contain temperature, humidity and pressure readings for a complete day starting from 00:00 hours to 23:59 hours. Data models were generated for each variable i.e. temperature, humidity and pressure. So for pressure, there were 1440 points but on applying the above algorithm on it, only 167 models (or points) were obtained. That is, the data set is reduced by 88% (approx.) with an error of 0.0709. For humidity, there were 1440 points but on applying our algorithm on it, only 150 models (or points) were obtained. That is, the data set was reduced by 90% (approx.) with an error of 0.0372. For temperature, there were1440 points but when on applying the algorithm on it, only 57 models (or points) were obtained. That is, the data set was minimized by 96% (approx.) with an error of 0.0568. So this shows how accuracy oriented strategy reduced the size of data set with a great accuracy to the models generated. Now on the same data set the performance oriented strategy is used. Here also data models were generated for each variable i.e. temperature, humidity and pressure. So for temperature, there were 1440 points but when on applying the above algorithm on it, only 267 models (or points) were obtained. That is, the data set was reduced by 81% (approx.) with an error of 0.0920. For humidity, there1440 points but on applying the above algorithm on it, only 154 models (or points) were obtained. That is, the data set was reduced by 88% (approx.) with an error of 0.0381. For pressure, there were 1440 points but on applying the above algorithm on it, only 16 models (or points) were obtained. That is, the data set was reduced by 98% (approx.) with an error of 0.0899.

Thus, it can be observed how these two strategies work differently by keeping focus on their respective functionalities. The following table depicts the summarized comparison between the two strategies on 1440 points.

Table 1: Comparison of both the techniques in terms of number of models and mean error



	Accuracy Oriented Strategy		Performance Oriented Strategy	
Independent variables	**Data Models**	**Mean error**	**Data Models**	**Mean error**
Temperature	57	0.0568	267	0.0920
Humidity	150	0.0372	154	0.0381
Pressure	167	0.0709	16	0.0899
Totality	**374**	**0.0549**	**437**	**0.0733**

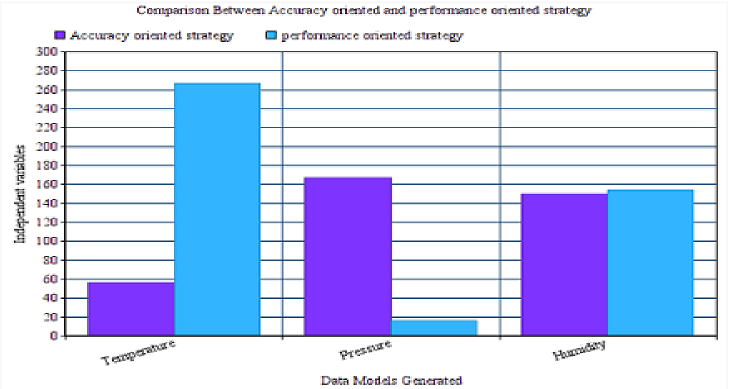


Fig 3: Comparison of both the techniques in terms of number of model

VII. RESULTS OBTAINED

To generate the models, previously existing data was used, collected from a MEMSIC sensor kit which consists of number of sensor nodes and a gateway. The sensor nodes took readings of temperature, pressure and humidity after every Two seconds. The algorithms were applied on single day of data which consisted of 43,200 points. On applying the accuracy oriented strategy on this data set, 12,920 data models for pressure, 1,284 data models for temperature and 58 data models for humidity were obtained. That is, for pressure readings the data set was reduced by 70% (approximately), for temperature readings the data set was reduced by 97% (approximately) and for humidity readings the data set was reduced by 99% (approximately).

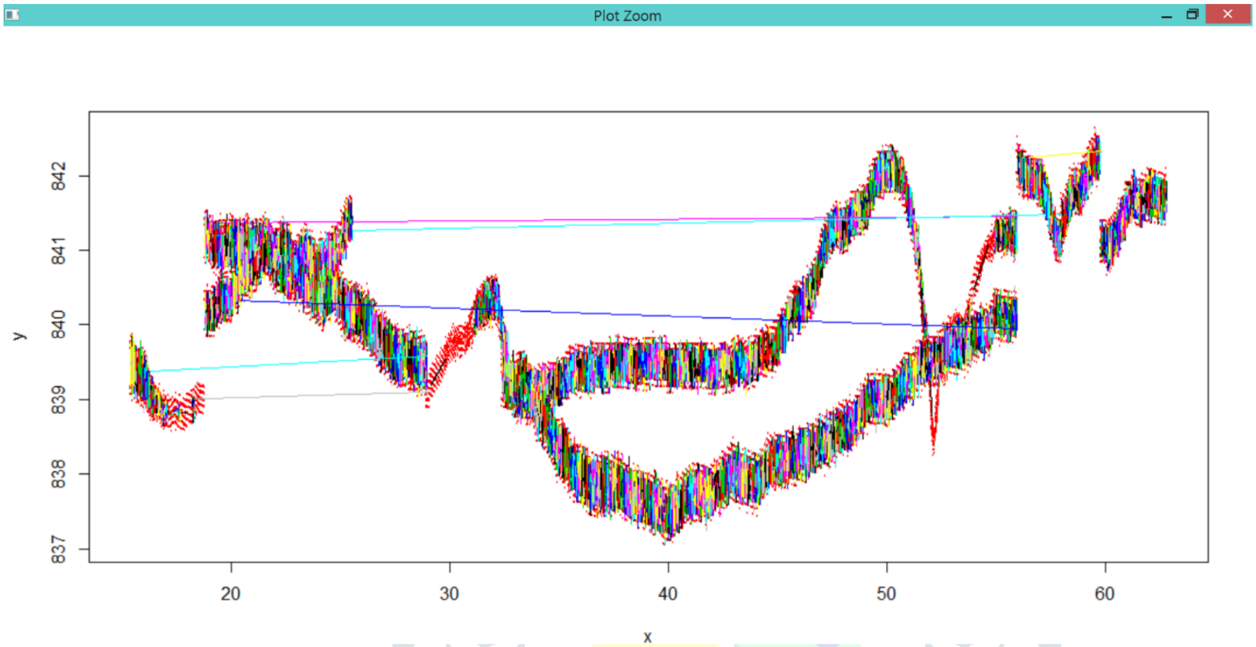


Fig 4: Plot depicting models generated for pressure

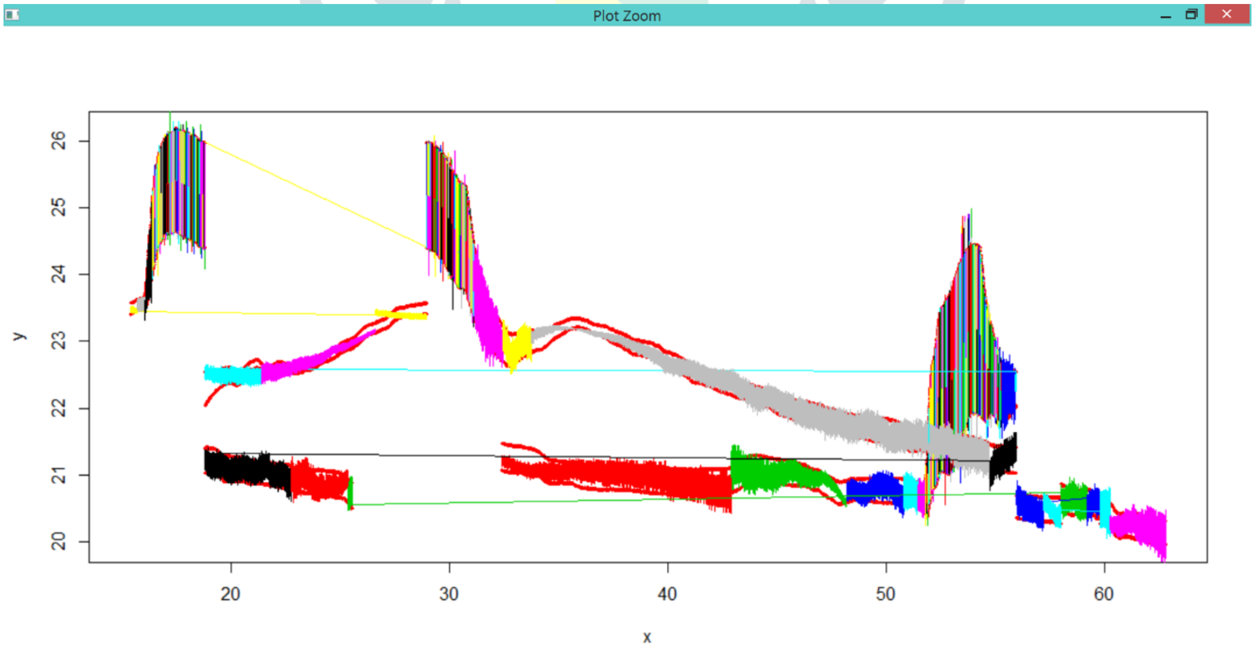


Fig 5: Plot depicting models generated for temperature

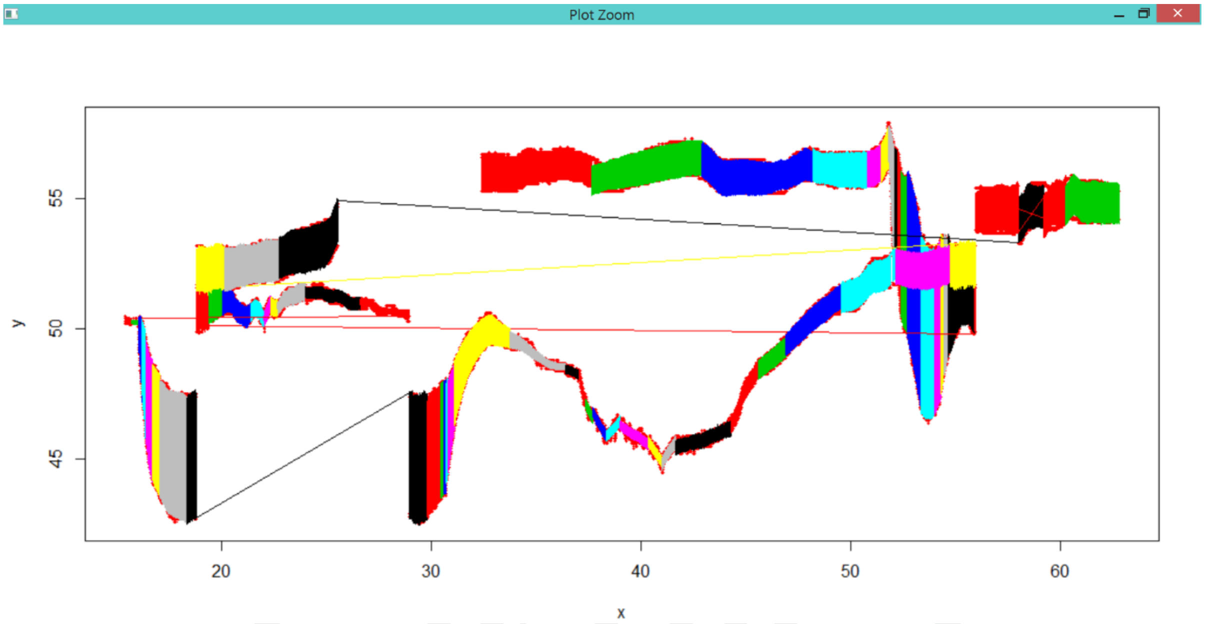
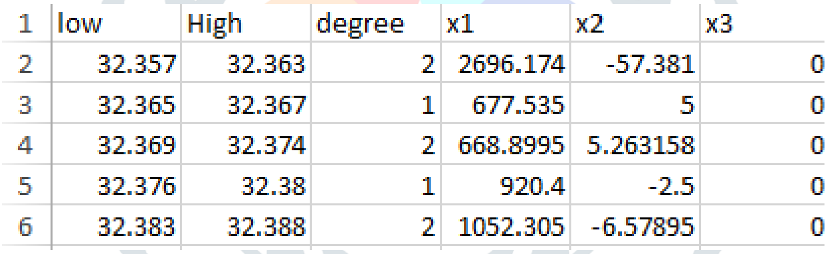


Fig 6: Plot depicting models generated for humidity

Thus, it can be seen how greatly the algorithm reduced the data set by generating the polynomial models for the recorded sensor data points. Also, these models are stored in tables corresponding to the equations used to generate these models. These tables contain the time stamp for which the model is stored, Degree of the polynomial and the coefficients which are used to predict the any reading when required.

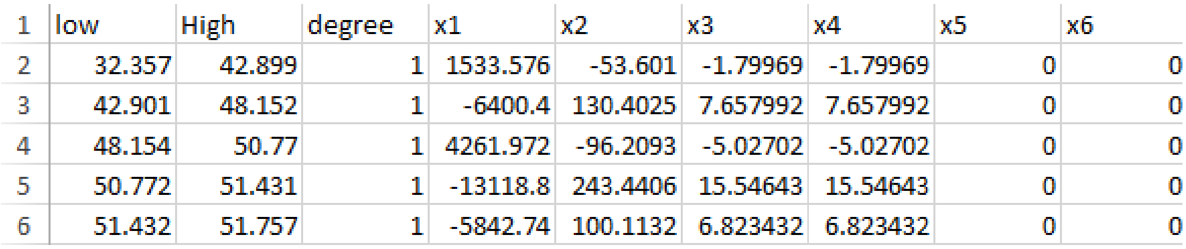
Following are the snapshots of the portion of the tables in which these data models are stored.



1	low	High	degree	x1	x2	x3
2	32.357	32.363	2	2696.174	-57.381	0
3	32.365	32.367	1	677.535	5	0
4	32.369	32.374	2	668.8995	5.263158	0
5	32.376	32.38	1	920.4	-2.5	0
6	32.383	32.388	2	1052.305	-6.57895	0

Fig 7: Portion of table storing data models for the pressure

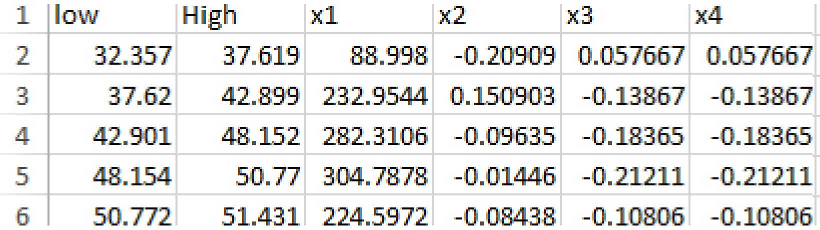
Here, the low and high fields depict the time stamp for which the models is stored, degree is the polynomial degree of the equation used and x1, x2 and x3 are the coefficients.



1	low	High	degree	x1	x2	x3	x4	x5	x6
2	32.357	42.899	1	1533.576	-53.601	-1.79969	-1.79969	0	0
3	42.901	48.152	1	-6400.4	130.4025	7.657992	7.657992	0	0
4	48.154	50.77	1	4261.972	-96.2093	-5.02702	-5.02702	0	0
5	50.772	51.431	1	-13118.8	243.4406	15.54643	15.54643	0	0
6	51.432	51.757	1	-5842.74	100.1132	6.823432	6.823432	0	0

Fig 8: Portion of table storing data models for the temperature

Here, the low and high fields depict the time stamp for which the models is stored, degree is the polynomial degree of the equation used And x1, x2, x3, x4, x5 and x6 are the coefficients.



1	low	High	x1	x2	x3	x4
2	32.357	37.619	88.998	-0.20909	0.057667	0.057667
3	37.62	42.899	232.9544	0.150903	-0.13867	-0.13867
4	42.901	48.152	282.3106	-0.09635	-0.18365	-0.18365
5	48.154	50.77	304.7878	-0.01446	-0.21211	-0.21211
6	50.772	51.431	224.5972	-0.08438	-0.10806	-0.10806

Fig 9: Portion of table storing data models for the humidity

Here, the low and high fields depict the time stamp for which the models is stored, degree is the polynomial degree of the equation used and x1, x2, x3 and x4 are the coefficients.Thus, the entire result can be summarized in the following table

Table 2: Result analysis of models generated for live data

Independent variable	No. of data models	Percentage reduction	Mean error
Pressure	12,920	71%	0.0902
Temperature	1,284	97%	0.0720
Humidity	58	99%	0.0660
Totality	**14,262**	**69%**	**0.0761**

VIII. CONCLUSION AND SUMMARY

With the increasing trend of information communication technologies, data is being created at enormous rates. It is becoming very hard to manage data and an efficient way to organize data in databases is an important issue. IoT model databases is becoming an important aspect in alleviating data generation by decreasing the space that data uses while also maintaining the same information. Data models also provide data with minor error that can fit many raw data points from sensors. These models are created by fitting a function to the data points. In this research, polynomials with different order were used, for example, first order, second order, etc. to fit the data points. Our algorithm finds a polynomial curve whose parameters are the coefficients of the polynomial equations. These parameters now enclose many raw data points within a time range. In other words, with data models enormous amount of data points can be represented without having to overfill databases or sacrifice data utility.

REFERENCES

[1] Moawad, T. Hartmann, F. Fouquet, G. Nain, J. Klein, and Y. Le Traon. Beyond discrete modeling: A continuous and efficient model for IoT. In Proc. of 18th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS).

[2] Xiaomin Xu, Sheng Huang, Yaoliang Chen, K. Browny, I. Halilovicy, and Wei Lu. TSAaaS: Time Series analytics as a service on IoT. In Web Services (ICWS), 2014 IEEE International Conference.

[3] Nguyen Quoc Viet Hung, HoyoungJeung, and K. Aberer. An evaluation of model-based techniques to sensor data compression. Knowledge and Data Engineering, IEEE Transactions.

[4] Mira Yun, Danielle Bragg, Amrinder Arora, and Hyeong-Ah Choi. Battle event detection using sensor networks and distributed query processing. Computer Communications Workshops, INFOCOM WKSHPS, IEEE Conference.

[5] Balakrishnan, Vasudevan, 'IoT Environment', 2016. [Online]. Available: https://www.quora.com/What-are-some-examples-of-Internet-of-Things-IoT-used-in-telecom-industry/ .

[6] A model-based sensor database for IOT, by Paul Maheshwari, University of miami Available: https://scholarlyrepository.miami.edu/oa_theses/628/

[7] Wikipedia, 'wireless sensor network architecture', 2015. [Online].Available: https://en.wikipedia.org/wiki/Wireless_sensor_network/

[8] Sensor | Types of Sensor. http://www.electrical4u.com/sensor-types-of-sensor/, 2011.

[9] Ankur Jain, Edward Y. Chang, and Yuan-Fang Wang. Adaptive stream resource management using kalman filters. In Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data.

[10] A. Moawad, T. Hartmann, F. Fouquet, G. Nain, J. Klein, and Y. Le Traon. Beyond discrete modeling: A continuous and efficient model for IoT. In Model Driven Engineering Languages and Systems (MODELS), 2015 ACM/IEEE 18th International Conference.

[11] W. Elmenreich and R. Leidenfrost. Fusion of heterogeneous sensors data. In Intelligent Solutions in Embedded Systems, 2008 International Workshop.

[12] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey.

[13] A.Ghaddar, T. Razafindralambo, I. Simplot-Ryl, S. Tawbi, and A. Hijazi. Algorithm for data similarity measurements to reduce data redundancy in wireless sensor networks. In World of Wireless Mobile and Multimedia Networks, 2010 IEEE International Symposium.