Fan Gongxiu Honors College

Analysis, Matching, and Prediction of Data with Time or Space Characteristics

according to Association Rule

of Data Mining

A Keystone Project

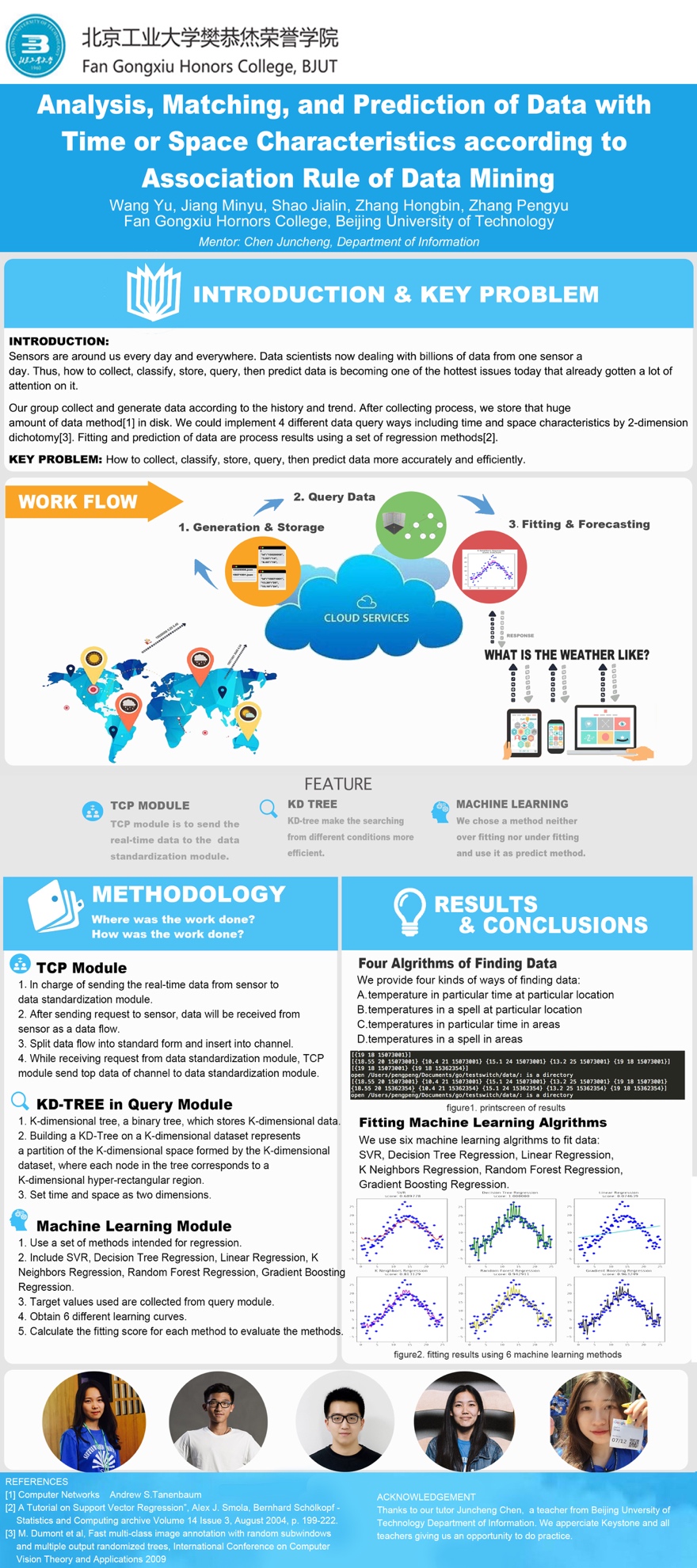
Final Report

Authors: Wang Yu, Jiang Minyu, Shao Jialin,

Zhang Hongbin, Zhang Pengyu

Supervised by: Chen Juncheng

Date:2017/12/22



# Executive Summary

The Executive Summary should be a maximum of 1 page. 1/2 page is ideal.

Considering about the huge amount of data from sensors we get every moment in our daily life, we develop a whole system to collect, transfer, store, query, and predict data with time and space characteristics. We separate the whole system into four main modules which are data generation module, TCP module, storage and index module, and machine learning module.

In this project, we run our system both on real-time data collected from temperature sensor DHT11 in Raspberry Pi and simulated temperature data according to data trend of reality with random perturbation points.

With the huge amount of data from sensors, we use our own Transfer Control Protocol (TCP) to build up connection between clients and servers then send data among them. While transferring data, different modules requires different types of data, so we transfer data between string and bytes. Along with the request message from server, TCP get the data in channel to send it to server.

While dealing with huge amount of data with time and space characteristics, we use KD-tree as the storage structure which offers us an effective way to store data with order and type. Then we set up index for them to make the data structure clear and easy to query.

We realize four kinds of query in data with combination of time and space characteristics, showing our advantages in searching speed and accuracy.

Fitting data is also a key problem which we solved in our project. We use a set of regression methods to fit data and use fitting score to evaluate the fitting result. Thus, we can choose the best ones for each query to improve the accuracy of prediction and further analysis.

As we have built up the whole system, storage, query, and prediction of data with time and space characteristics is no longer a hard question. But further research still requires us to achieve higher and higher accuracy and less searching time.

As a further wish, we hope this system can be used in many fields like smart city, medical treatment, mobile application and crime analysis.

# Background

**Key Problem:**

How to collect, classify, store, query, then predict data.

1. How to use specific methods to store and query data from parallel database which contains a great deal of data from varied sensors.
2. How to improve searching time-space data efficiency?
3. How to make data both visible and small-sized?
4. How to send the real-time data from sensor to data standardization module.
5. How to fit the discrete data to a continuously curve or a hyper plane.

**State of Art**

Because of the amount of data that we are sending and receiving every day is increasing dramatically, information explosion is becoming the core of this era. Thus we could say that we are living in the real big data era. In order to reduce the running load of data processing, we need to sort, process, and manage data, thus data mining is becoming a hot issue.

The main value of data mining is its capability of transferring huge amount of data into useful information and knowledge, which have been used in various applications. Nowadays, data mining is using ideas from the following fields: statistical sampling, estimate, hypothetical test, search algorithm and modeling in machine learning. [1]

Generally speaking, engineers around the world usually utilize the following analytical method to do data mining [2]: classification, estimation, prediction, affinity grouping or association rules, clustering, and complex type data mining. The core is the research and improvement of association rules, which could be divided into simple association, time association, and reason-result association. The objectives of association analysis is to figure out the hidden network of relationship, which is likely to be hidden label, then group or layer data. This could simplify the following search process during retrieve in order to avoid time and resources waste resulted by data traverse.

However, due to the characteristic of association rules data mining, it shows better effect in handling discrete data. Thus the rationality of dividing continuous data of raw database before data mining would influence result of association rule data mining.

Along with these ideas, database system should provide effective support in storage, index, and query. Another point is that the parallel computing (high performance computing) also plays vital role in processing massive amount of data, especially when huge amount of data are separated.

While talking about parallel computing, MapReduce [3] is an effective tool. Map operation means it could process each element separately but not generally. Aside from that, reduce operation could do recursive operation in highly parallel environment. Hadoop [4] Distributed File System (HDFS) is one of the most influential practice using MapReduce, which is a distributed file system and do well in distributed processing of massive data. Hadoop have the following advantages compared to traditional relational database: higher reliability, higher expansibility, higher efficiency, higher fault tolerance, and lower cost. The appearance of Hadoop brings a new possibility for parallel computing, which makes MapReduce not only a result in lab but also a practical and useful system.

In this section, briefly provide the context and the rational for the project. It should answer the questions:

After considerable discussion, we take temperature as an example to show the whole process of collecting data, transferring data, storing data in index, querying data, fitting data, and prediction.

Not only in bicycle, car-pooling is becoming a hot issue. People spend most of their daily time at home and working place, so the daily commute is the most frequent travel trajectory. In order to protect the environment and save energy, car-pooling is a great choice for people. But car-pooling has a disadvantage of efficiency. If someone need to find his companion after work, he may need to wait from 5 minutes to 30 minutes, even not find the proper person to go with. This may annoy him then lead to reduce his trust of car-pooling. But according to the data mining of people’s travel trajectory and their trip mode, we could predict their possible travel trajectory. Thus people could just set the travel time then get the recommended companion whenever they need before finishing work. This could effectively avoid rushing hours and increase the success rate.

After the consideration of status quo, we proposed a novel solution to solve the problem that we are facing in this big data era.

# Objectives

**Question1：How to store the huge amount of data the sensors collect?**

1. Classifying the data according to specific characteristics.

2. Store relative data together and separate different kinds data.

**Question2: How to search the required data fragments from the big amount of data?**

1. Using the idea of index to store data in order.

2. Use better algorithm to avoid useless search in huge amount of data.

3. Find a way to reduce the searching time.

**Question3: How to fit the huge amount data for further prediction?**

1. Classifying the data while storage, and get them sorted.

2. Handling the data with different fitting methods.

3. Avoiding traversal algorithm.

4. Using parallel computing to optimize the algorithm.

**Question4: How to get the whole system run with accuracy and efficiency?**

1. Find a proper protocol for hosts.

2. Build up connection for clients and servers.

3. Transfer data with accuracy and efficiency.

# Methodology

Considering about the time and space characteristics of data, we need to simulate test data and collect real-time data, then build up a protocol to realize the communication of different hosts, then find a more effective way to store data and set up index for it. Besides, using regression methods in machine learning also help us fit all the data thus we could do some prediction work based on the collected or generated data. Thus, we divided the whole system from data collection to data prediction into 4 main modules: data module (sensor), TCP module, storage and index module, and machine learning module.

1. **Data Generation and Collection Module**

In the first stage, we see through the temperature data for the last decade, finding that the temperature per day is likely to be in normal distribution. Then we use Python to simulate data for a day with interval of 25 minutes.

In the next stage, we use temperature sensor DHT11 from Raspberry Pi [9]in Python to collect real-time data for a period or a whole day. Collection interval is 5 minutes in the live demonstration, and we could set a maximum number of collected data as a limit or just let it do the cycle all around the year.

Data structure: [sensorid, timestamp, temperature]

1. **TCP Module**

TCP (Transfer Control Protocol) module acts as a client of sensor. TCP module establishes the connection with the sensor by dial the IP address through the TCP protocol, then send a byte slice to the sensor to ask for data.[7]

TCP module creates an empty slice buffer to save the data sent from the sensor. In order to make the data readable, we use strings.

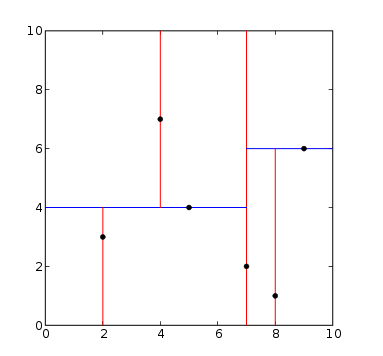
TCP module ranges on slice provides both the index and value for each entry. With the index, we find the corresponding value and transfer the type into byte. We create a new channel with make(chan [], 1024) syntax, then send value into the channel.[8]

TCP module acts as a server of the standardization module. After connection, it will listen the port and get ready for the coming request. The moment a request has been accepted, TCP module will take the first data from the channel and send it to the data standardization module.

1. **Storage and Index Module**

Balancing the visualization and the size of data is a tough work. Using numbers only is good to reduce the size of data, but hard for human to recognize. So, combining the numbers with separator will not only minimize the size but also make it easier to read. Therefore, we choose JSON (JavaScript Object Notation) to save data with time and space characteristics. For each type of data, we give it a new JSON file to save. And for each single line in this type of data, we only save the key value of the data and show the two characteristics on the JSON file.

Dichotomy method is a common method in search. Traditional dichotomy method can only work well with one-dimension data, like S = {1, 6, 8, 10, 2}, but our data with time and space characteristics is 2-dimension data. In order to search efficiently, KD-tree (K-dimension tree), a query index tree structure which is widely used in database index is a good choice for us to deal with fast query and storage of high-dimension data. [5]

Building a KD-Tree on a K-dimensional dataset represents a partition of the K-dimensional space formed by the K-dimensional dataset, that is, each node in the tree corresponds to a K-dimensional hyper-rectangular region. Time and space are working as two dimensions here.[6] KD-tree costs less time when searching data rely on both time and space.

Core of KD-tree is to put smaller value than the current node in the left subtree while putting larger value in the right subtree. Then the hyperplane would be separated by the current node’s value.

Figure 1 K-Dimensional hyper-rectangular region

1. **Machine Learning Module**

In order to fit data into a curve then combine them into a plane or 3D plot, we use the following six regression methods in machine learning and calculate fitting score to evaluate various methods to find the best one for each query and situation.

**Linear Regression** isa linear approach for modeling relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X, which is under fitting in our research.

**SVR (Support Vector Regression)** [11]is a method of Support Vector Classification, whose basic principle is to find a regression plane which makes the minimum distance for each one in the data set to the plane.

This model only depends on a subset of training data, because the cost function for building the model doesn’t care about training points that lie beyond the margin.

**Nearest Neighbors Regression** is mostly used in cases where the data labels are continuous rather than discrete variables. The label assigned to a query point is computed based the mean of the labels of its nearest neighbors.

K Neighbors Regression is one of the Nearest Neighbors Regression which implements learning based on the k nearest neighbors of each query point, where k is an integer value specified by the user.

**Decision Tree Regression** is a basic classification and regression method. Three main steps are as follows:

1. Feature selection: using information gain criterion algorithm select feature；

2. Decision tree generation: using classical ID3 algorithm to generate tree;

3. The pruning of the decision tree: in order to prevent over fitting phenomenon.

The generation of the decision graph corresponds to the local selection of the model, while the pruning of the decision tree takes into account the global minimum selection.

**Random Forest Regression** is the expanding of the decision tree regression which establishes a forest with many decision trees in it in random way. There is no correlation between every decision tree in the random forest. The predicted value is the weighted average value of the target variable of the leaf node.

**Gradient Boosting Regression** is the expanding of the decision tree regression.

Compared with the traditional gradient learning, Gradient Boosting Regression chooses the direction of gradient descent to ensure that the final results are best at the time of iteration.

**Method Evaluation:**

To evaluate these methods, we calculate the fitting score for each of them. Estimator in sklearn has a score method, which provides a default evaluation rule to solve the problem. Each score calculate the residual sum of squares ((y\_true - y\_pred) \*\* 2).sum() and the total sum of squares ((y\_true - y\_true.mean()) \*\* 2).sum().

Through these fitting scores we could figure out the best method through comparison. Under fitting means that the complexity of the model is too low to fit all the data well, and the training error is large, while over fitting shows that the model complexity is too high, the training data are too little, the training error is small, and the test error is big.

# Key results and achievements

## Key results and discussions

1. **Sensor and TCP Module**

Collect data from DHT11 in Raspberry Pi with time interval of 5 minutes. Data testing lasts for a whole day or even longer. The accuracy and reliability could be guaranteed with the support of TCP module, and we could also reduce the possibility of losing data into an extremely low extent.

Build up connection between sensor and TCP module, where TCP module send request message to sensor and sensor send back data flow to TCP module.

Split() syntax to split the data flow into standard form, then save then into slice sendArray.

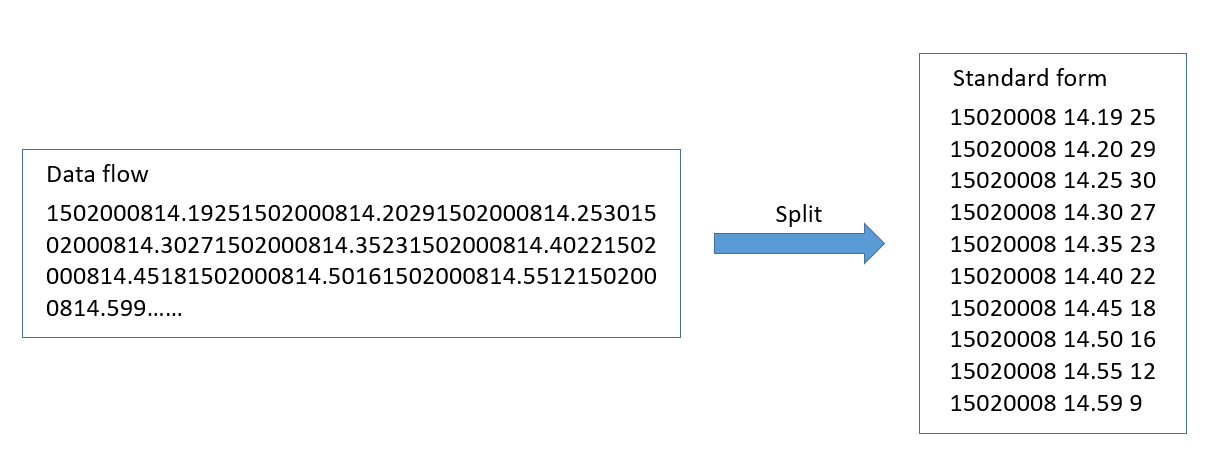


Figure 2 Split data flow into standard form

Find the corresponding value using index then transfer its type into byte. Then create a new channel and send value into the channel.

TCP module take the first data from channel and send it to the data standardization module as soon as it accepts a request while listening the port.

1. **Storage and Index Module**

Using K-Dimension tree to store data from sensors is really suitable.[12] As we all know, the static sensor's address is constant, so we build up a Time-Value index assuming time as T and temperature as V. As for numbers of sensor, we gather different kd-trees together and regard them as a forest.

Sensor1 Sensor2 Sensor3

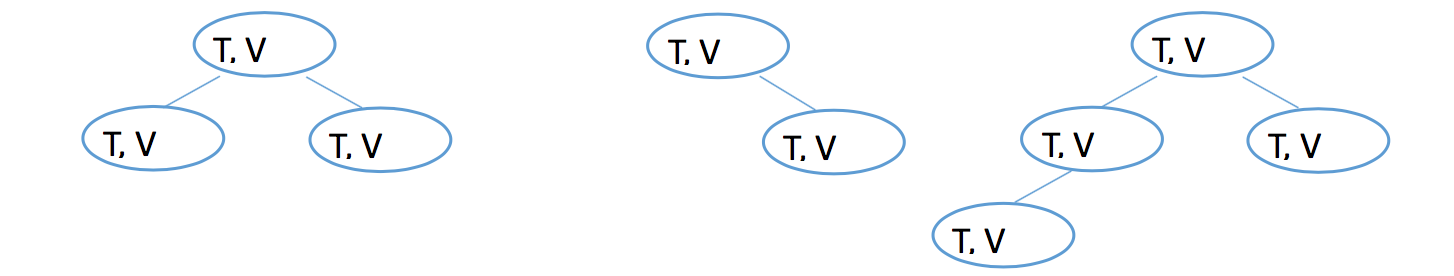
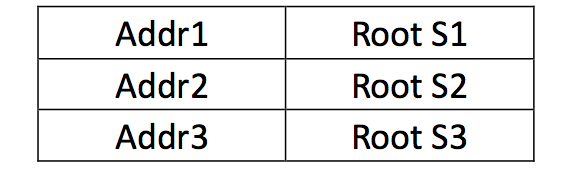


Figure 3 KD-tree forest Figure 4 Map of forest

At the same time, root node should be connected with sensor ID, so a map is necessary to store the node information.

KD-Tree structure provides 2-dimension searching method. A parameter *Split* should be added. *Split* means which dimension we should consider. For example, since we have 2-dimension data (x,y). split = {0,1}. When split = 0, we should calculate the first parameter x; When split = 1, we should calculate the second parameter y.

As for the following example, we want to find the temperature between time 10.30 and 13.30 and temperature between 15℃ and 24℃.

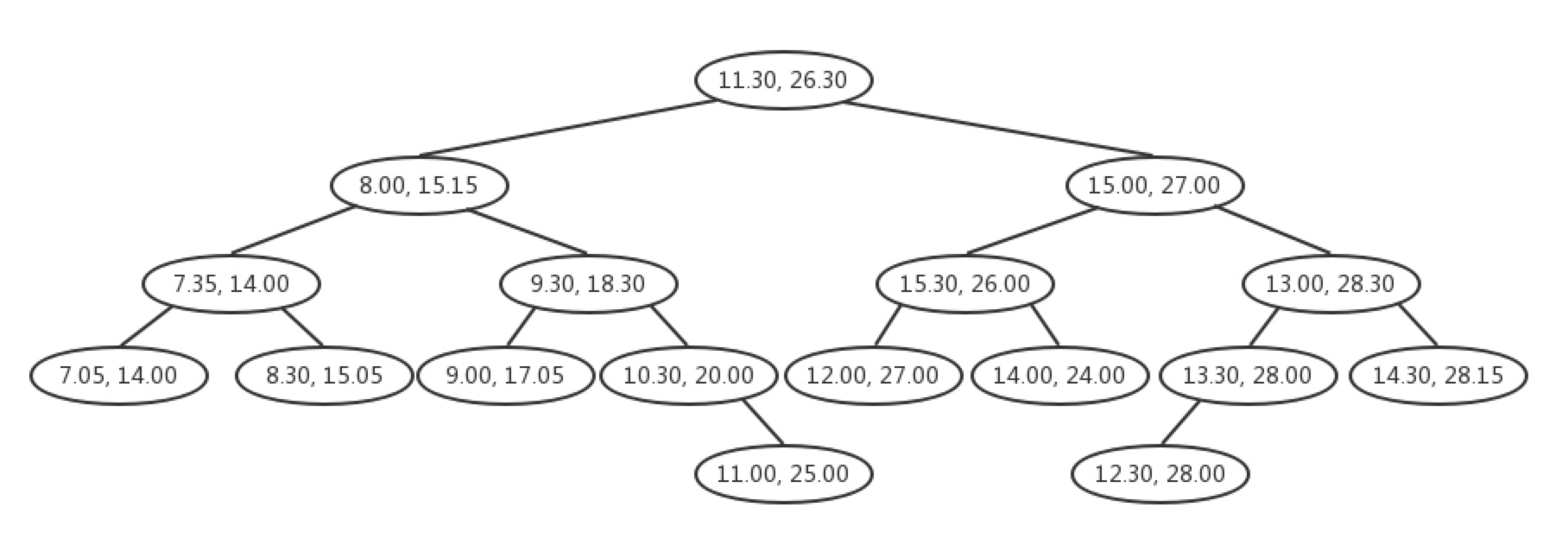


Figure 5 Query example-searching tree

S1: Inite splite =0. Since the splite = 0, compare the time 10.30 with the initial node point(root: 11.30, 26.30). 10.30<11.30. So the current node should move to the left child. splite = splite +1.

S2: Since the splite =1, compare the temperature 15.15℃ with 15℃, 15 < 15.15, move the current node to left child. splite = splite +1.

S3: Since the splite =0, compare the time 10.30 with the time 9.30. 10.30 > 9.30. So the current node should move to the right child. splite = splite +1.

S4: Repeat steps S2 and S3, until 10.30 > nodeA.time and 15 > nodeA.temperature. Output all node which satisfy that the time 13.30 < nodeB.time and 24 < nodeB.temperature.

In this way, we provide 4 query methods:

1. Temperature in particular time at particular location
2. Temperatures in a spell at particular location
3. Temperatures in particular time in areas
4. Temperature in a spell in areas

KD-tree can identify 2-dimension data with dichotomy by using 2-dimension dichotomy while other method can only deal with 1-dimension data.

1. **Machine Learning Module**

After data collection, we use a set of 6 regression methods to fit the dataset in order to solve the query mission in the KD module.

The regression methods set includes SVR, Decision Tree Regression, Linear Regression, K Neighbors Regression, Random Forest Regression, Gradient Boosting Regression.

**Query B:** temperature in a period at a spot

The data set in this query is 2-dimension (temperature, time).

The fitting result is shown below, and we could conclude that SVR and K Neighbors Regression fit the data set better than other methods.

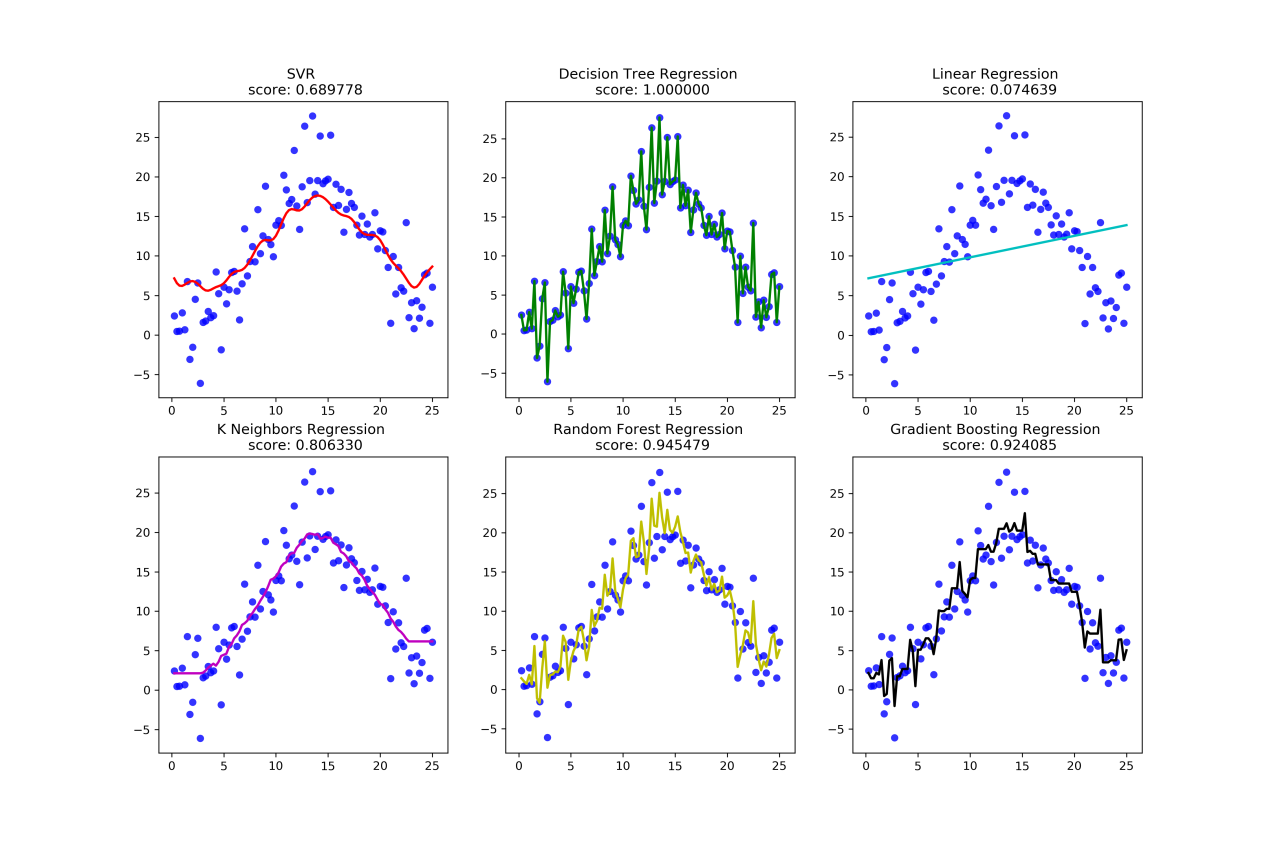


Figure 6 Query B

**Query C:** temperature at a moment in particular area

The data set which is 2-dimension (temperature, sensorid). Here we regard a location as a sensorid.

The fitting result is shown below and it shows clearly that SVR and K Neighbors Regression fit the data set better than other methods.

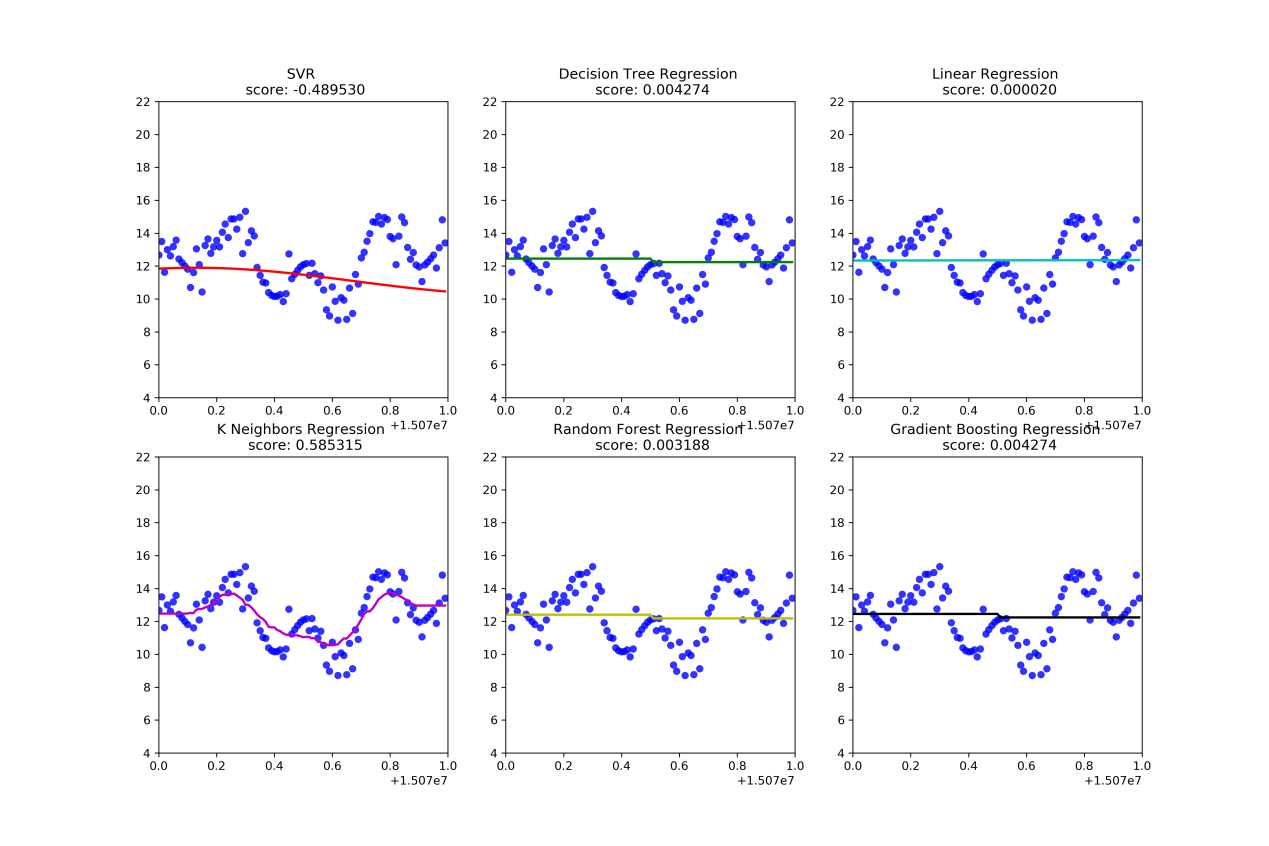


Figure 7 Query C

**Query D:** temperatures in a period in specific area

The data set in this query is 3-dimension (temperature, sensorid, time).

The fitting result is shown below, we can find out that SVR fit the data set better than other methods.

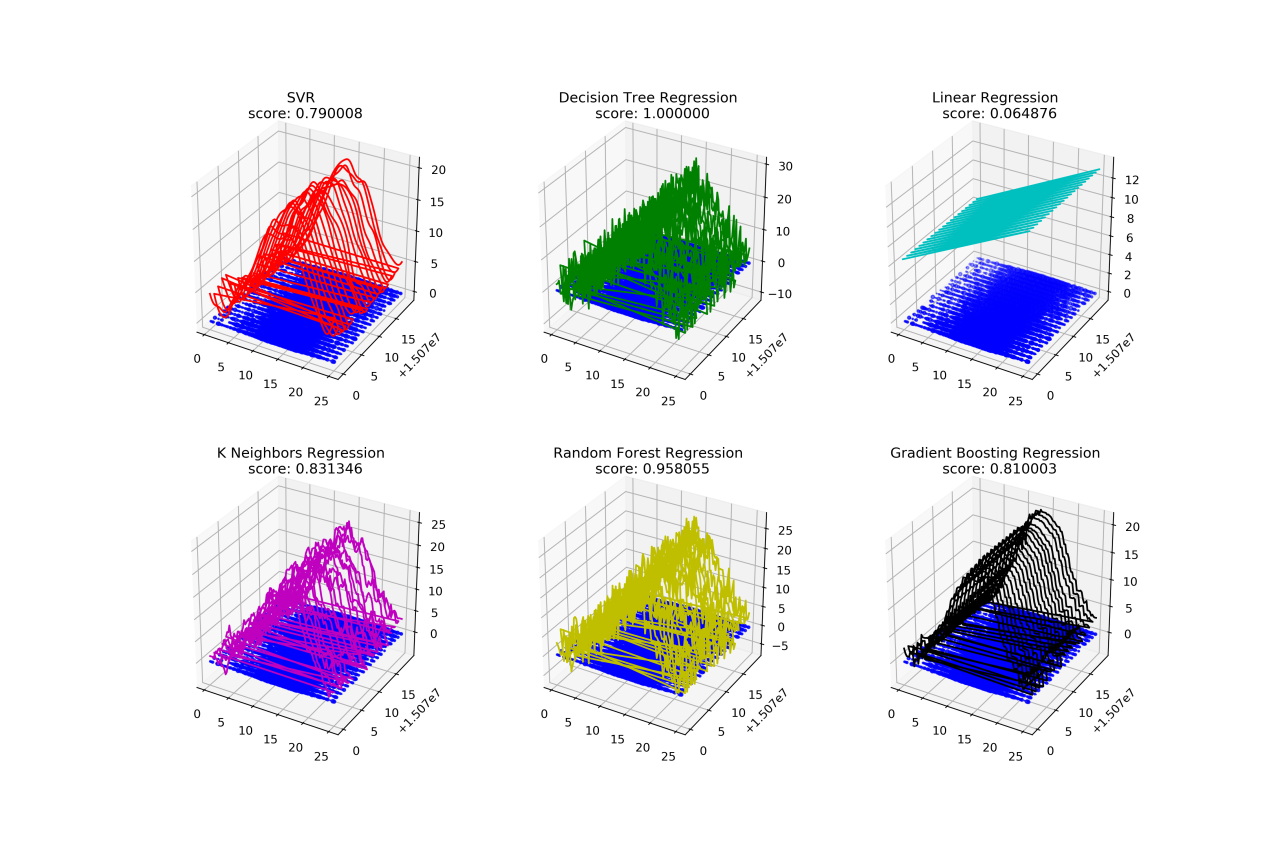


Figure 8 Query D

SVR fitting result is still in linear level and is still not continuous. This can’t present the overall trend of data. Thus, fitting the curve which we get from SVR into a hyper plane which is based on the data set can help us predict any time’s and sensorid’s temperature.

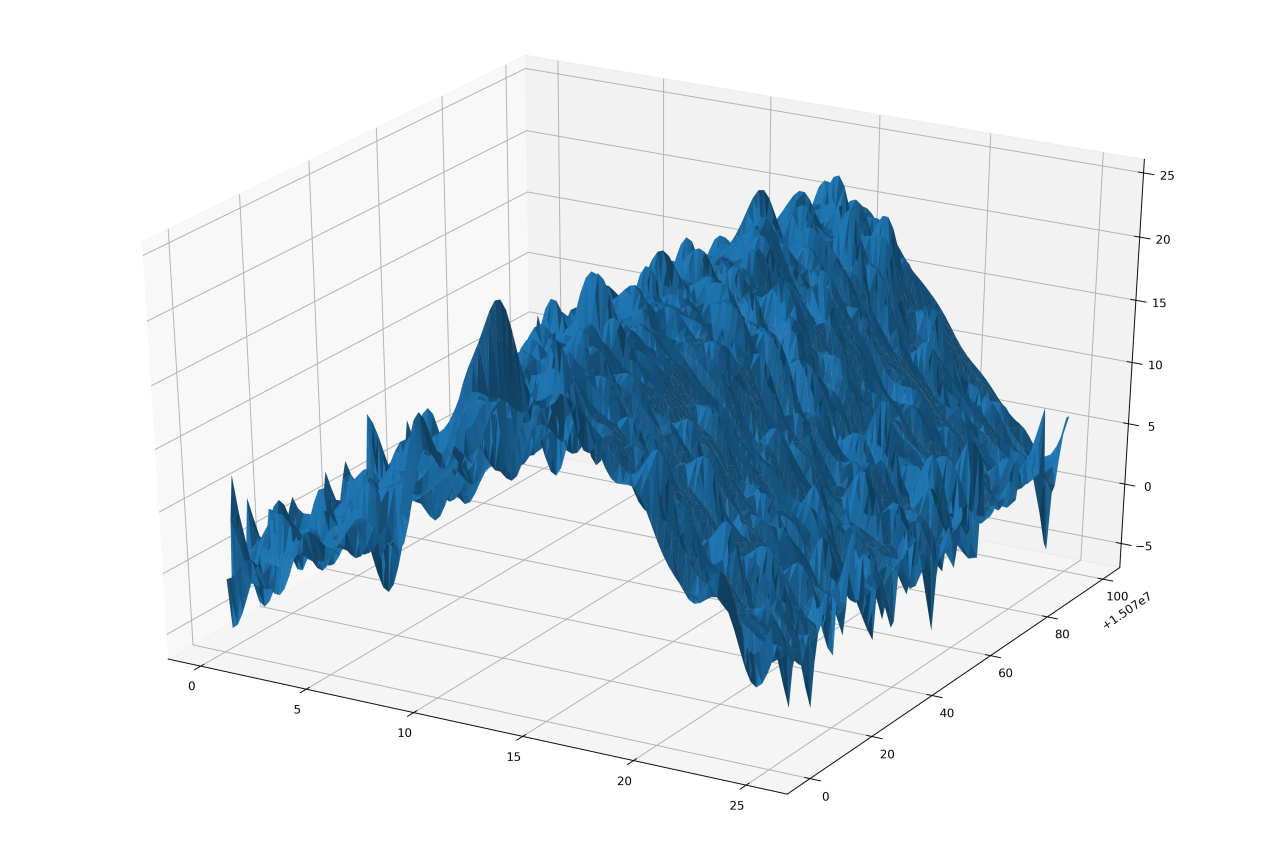


Figure 9 Hyperplane of SVR curve

## The achievement against activities and milestones

Achievements according to objectives are listed below:

|  |  |  |
| --- | --- | --- |
| Scheduled date | Expected results | Achieved milestones |
| 2017.2.5 | Get data set ready | Simulate data according to real data then add perturbation points. |
| Collect data from real temperature sensors. |
| 2017.3.25 | Store data in high efficiency | Save data in both visible and small-sized then build data in high-efficiency structure of K-Dimension tree. |
| 2017.4.20 | Store data in order with index | Handle data with high speed in memory which based on index. |
| 2017.6.10 | Transfer data and build up connection between clients and servers | Split data flow received from sensor then send data to standardization module in reliable way. |
| 2017.8.15 | Query data without useless search | Realize 4 different queries. |
| 2017.9.20 | Fit data well and do prediction work in high accuracy | Fit data set into a function curve by using regression methods in machine learning then evaluate quality of each method we use for the best one. |

# Impacts

**Scientific impacts:**

KD-Tree used in searching time-space data can reduce the time complexity to some extent.

KD-Tree and searching methods are written in Go language [10], which can be a reference of people following behind.

TCP provides reliable, ordered, and error-checked delivery of a stream of octets between applications running on hosts communicating by an IP network.

Channels and goroutines are used to handle the data so that data can be handled in memory and get operated at a high speed.

Based on the date set of the weather changed per day, machine learning module finds out a suitable machine learning methods for the weather liked data set in analysis and prediction.

**Community impacts:**

The learning curves can help the meteorological department to predict weather more accurate.

# Conclusions and recommendations

In order to develop a whole system of processing data with time and space characteristics, we build up our own TCP (Transfer Control Protocol) to guarantee the communication between different hosts and various client-server relationship.

In comparison with other store and index method, we find that KD-tree has the highest rate of accuracy and fastest speed in data query so we develop out storage and index module based on KD-tree.

In order to fit data better, we regard time and space as 2 dimensions that we take a sensorid as a symbol of one specific location. Thus, we could simplify the fitting process and get a more accurate fitting curve or plane.

In the tests of both real-time collected from temperature sensors and simulated data with perturbation points, we figure out the best methods for each query.

SVR and K Neighbors Regression fit best in querying temperature in a period at a spot and temperature at a moment in particular area. SVR fits best in querying temperatures in a period in specific area.

The result shows that we need different regression methods in dealing with different queries. We can also learn from the result that a high-speed storage and index method plays a quite important for a data processing system, which is the most apparent parameter for users. There is also great need for a reliable TCP in working among various hosts.

This system shows that it is possible to collect, transfer, store, index, query, and predict huge amount of data with time and space characteristics. But we also know that the query speed is extremely significant in this system, so the following will focus on how to optimize the storage structure and how to shorten searching time.

We truly hope that our work is worthy, and it could be used in many other fields besides temperature. Smart city, crime analysis, smart transportation, and many other fields need this kind of solution to massive data collected from sensors every day around us. We believe that dealing with data around us is a meaningful thing to do because it can truly improve everyone’s life quality and bring human beings more convenience and economic value.

# Acknowledgments

Thanks to our tutor Juncheng Chen, a teacher from Department of Information, Beijing University of Technology.

We also appreciate Keystone Project of Fan Gongxiu Honors College and all teachers giving us an opportunity to do practical exercise and test.

# References

In this section, list references used in the report as well as published reports about this project.

[1] S.N. John, C. Anele, O. Okokpujie Kennedy, F. Olajide, Chinyere Grace Kennedy. Realtime Fraud Detection in the Banking Sector Using Data Mining Techniques/Algorithm. Computational Science and Computational Intelligence (CSCI), 2016 International Conference on (2016).

[2] Leslie Mertz. What Can Big Data Tell Us About Health?: Finding Gold Through Data Mining. IEEE Pulse. (2016) 7(5): 40-44.

[3] Seema Maitrey, C.K. Jha. Handling Big Data Efficiently by Using Map Reduce Technique, Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on (2015).

[4] Trupti V. Kenekar, A.R. Dani. An efficient private FIM on hadoop MapReduce, Automatic Control and Dynamic Optimization Techniques (ICACDOT), International Conference on (2016).

[5] Bentley, J.L, Multidimensional binary search trees used for associative searching[C], Communications of the ACM , 1975

[6] M. Dumont et al, Fast multi-class image annotation with random subwindows and multiple output randomized trees[C], International Conference on Computer Vision Theory and Applications , 2009

[7] Andrew S.Tanenbaum, Computer Networks[M]

[8] W.Richard Stevens, The Protocols[C], TCP/IP Illustrated, 2012(1):64-66

[9] Matt Richardson, Shawn Wallace, Getting Started with Raspberry Pi[M]

[10] Brian W. Kernighan, Alan Donovan, The Go Programming Language[M]

[11] Alex J. Smola, Bernhard Schölkopf, A Tutorial on Support Vector Regression[J], Statistics and Computing archive, 2004, 14(3), 199-222.

[12] Mehdi Sookhak, Richard Yu, Albert Zomaya. Auditing Big Data Storage in Cloud Computing Using Divide and Conquer Tables. IEEE Transactions on Parallel and Distributed Systems （2017）PP（99）

# Appendixes

**Budget Use:**

|  |  |
| --- | --- |
| **Item** | **Cost** |
| Raspberry Pi | 2500 |
| Books（Machine Learning） | 400 |
| Books（SQL） | 210 |
| SD card | 260 |
| Sensors | 250 |
| Books（Artificial Intelligence） | 350 |
| Books（TCP/IP） | 150 |
| Books（goloang） | 350 |
| HDMI wire | 180 |
| **TOTAL** | **4650** |