Chapter 1

Introduction

This chapter depicts our related research introduction, such as motivation in section 1.1, goal in section 1.2, research architecture in section 1.3 and guide to this dissertation in section 1.4.

1.1 Motivation

First, as traditional tables are always used in the manufacturing industry to record quality-related problems raised by customers, it lacks for an effective way to make use of the most important information. It results in wasting time and production cost on investigations and analysis when the problems reoccur. Next, data are often large and complicated in the manufacturing process; the users in charge of quality-related problems can hardly identify the discrepant factors and generalize the characteristics rapidly and correctly. Then, the frequency of machine holding lot, cycle time and product yield during the manufacturing process need to be improved continuously. For these three main reasons, the challenge required to be settled in the study lies in the conclusion of the problems arising in connection with the semiconductor manufacturing industry.

1.2 Goal

According to the statistics of Kdnuggets [28] conducted on the users of data mining community, the statistics of frequently used data mining techniques are shown in Table 1. Furthermore, there are cases applying data mining in a number of literature reviews, such as manufacturing, financing and telecommunications. Among the available data mining tools, the common classification methods are statistics,

memory-based reasoning, link analysis, decision tree, neural network and genetic algorithm, shown in Table 2 [2].

First, the study is intended to use the Microsoft SQL Server 2005 classification data mining methods, i.e. decision tree, neural network, Bayesian, logistic regression and association rule, to analyze and identify which data mining method is better in application for the semiconductor packaging industry, based on implementation outcomes. Next, we apply the data mining and data warehouse to assist the user in charge of quality-related problems to analyze large data and identify the major discrepant causes. Then, a quality improvement system is constructed to lower the frequency of machine holding lot, cycle time to decrease the chip defects and finally to increase the product yield during the manufacturing process.

Table 1 Most used data mining/analytic methods

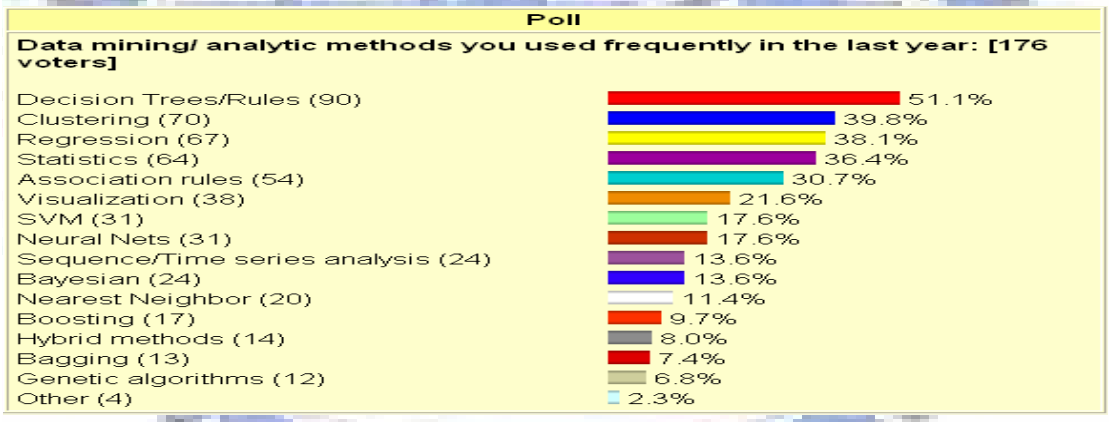


Table 2 Classification of data mining technology

Technology	Classification	Estimation	Prediction	Affinity grouping	Clustering	Description
Statistics	◎	◎	◎	◎	◎	◎
Market Basket Analysis			◎	◎	◎	◎
Memory-based Reasoning	◎		◎	◎	◎	
Clustering					◎	
Link Analysis	◎		◎	◎		
Decision Tree	◎		◎		◎	◎
Neural Network	◎	◎	◎		◎	
Genetic Algorithm	◎		◎			

1.3 Research Architecture

To overcome the product yield mentioned in the previous sections and quickly meet customers' competitive advantages in the semiconductor packaging industry, this study presented a comparison and evaluation of the five classification algorithms to achieve the goal mentioned above. We used the clustering algorithm to classify the categories of problem from the data warehouse. Hence, a product quality improvement system adapting a better algorithm as the core engine is established to discover the major causes of product problems and find out the countermeasures for resolving them. The tape carrier package process analyzed by this research scope is illustrated in Figure 1 and the research architecture is shown in Figure 2.

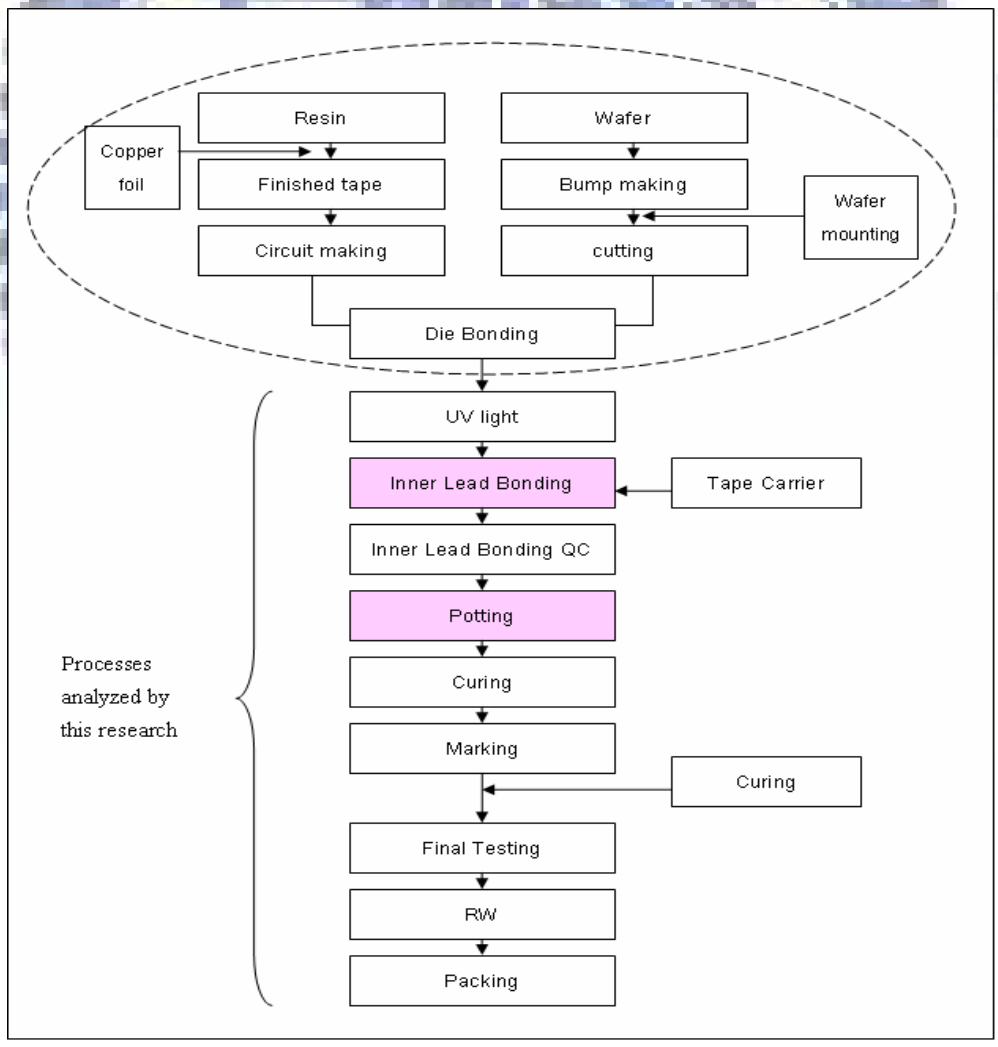


Figure 1. Tape carrier package processes analyzed by this research.

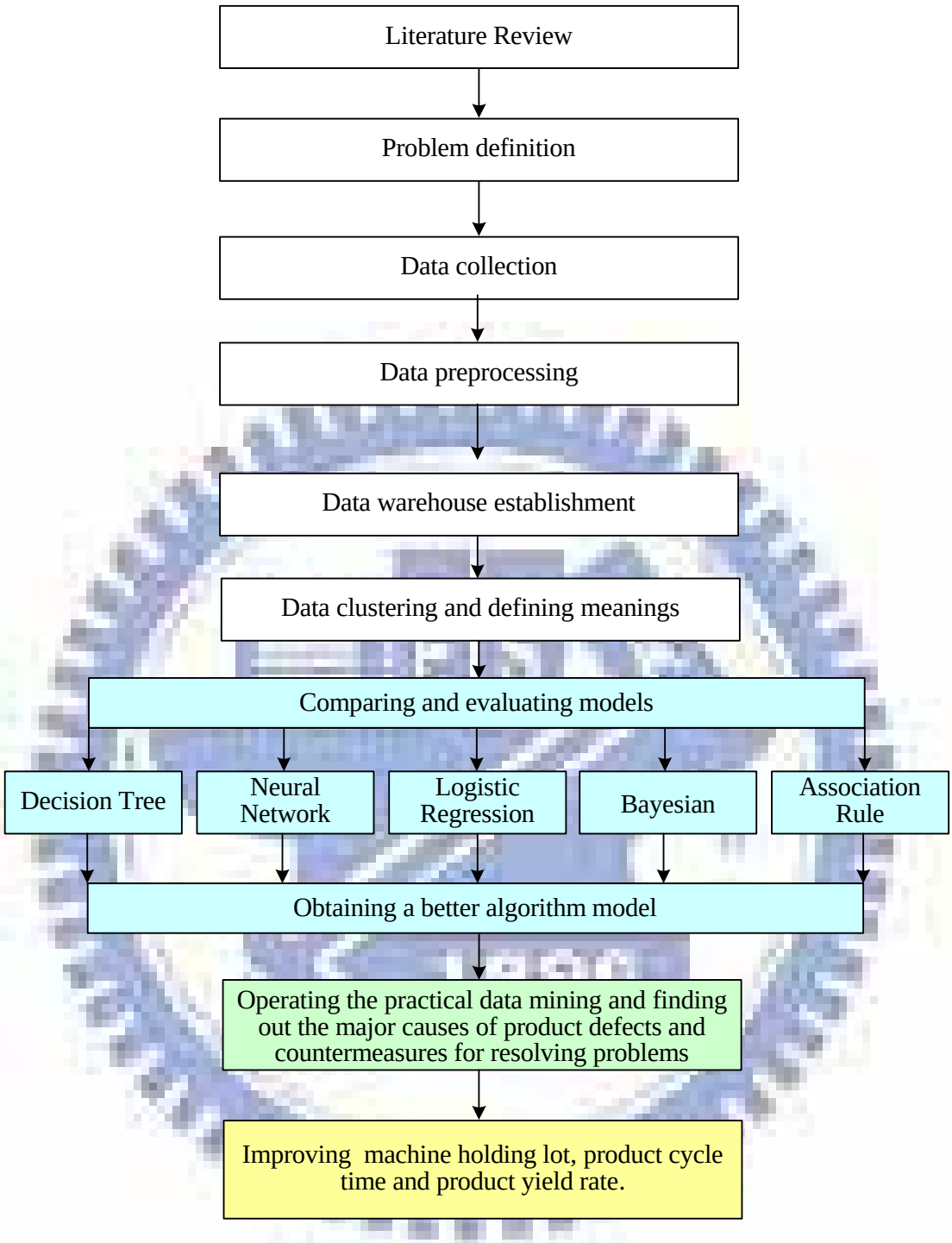


Figure 2. The research architecture.

1.4 A Guide to this Dissertation

The rest of the chapters of this paper are organized as follows. Chapter 2 introduces the background knowledge regarding data warehouse, data mining, and related studies of semiconductor packaging industry. Chapter 3 introduces the main trends in the Liquid Crystal Display driver IC process nowadays, and describes the

process flow in tape carrier package technology. Chapter 4 depicts the system design method and framework process flow proposed in this paper to explain how to apply data warehouse and data mining technology in this research. Chapter 5 engages in the system implementation environment and analysis of implementation results in order to identify causes for the defects and countermeasures to solve these problems. Evaluating a better algorithm and adapting the algorithm to bring about the benefits including machine holding lot, product cycle time and product yield rate. Chapter 6 concludes the result of the analysis in the hope of contributing to the packaging process improvement and product yield increase.

Chapter 2

Related Works

This chapter states the background knowledge required and technology involved in the exploration of data such as data warehouse in section 2.1, data mining in Section 2.2, and traditional semiconductor packaging processes in sections 2.3.

2.1 Data Warehouse

A data warehouse involves not only all data required but applications essential to data processing. These applications include the data transformation into the data warehouse applications from external media. The data can be categorized as (1) fact data, (2) metadata, (3) dimension data, and (4) aggregation data, and the application software can be categorized as (1) load manager, (2) data warehouse manager, and (3) query manager [19].

Data warehouse is the summation of all decision-making support techniques. It assists knowledge workers in making decisions better and faster [14] and it is the core of a decision-making support system. It is believed that, data warehouse should not only have a database function, but it should also have the following four features: (1) Integrated – a data warehouse is usually constructed by integrating multiple heterogeneous sources, such as relational databases, flat files and on-line transaction records. Data cleaning and data integration techniques are applied to ensure consistency in naming conventions, encoding structures, attribute measures, and so on. in naming conventions. (2) Subject-oriented – a data warehouse is organized around major subjects, such as customer, supplier, product and sales. Rather than concentrating on the day-to-day operations and transaction processing of an organization, a data warehouse focuses on modeling and analysis of data for decision makers. Hence, data warehouses typically provide a simple and concise view around

particular subject issues by excluding data that are not useful in the decision support process. (3) Time-variant – data are stored to provide information from a historical perspective. Every key structure in the data warehouse contains, either implicitly or explicitly, an element of time. (4) Nonvolatile – a data warehouse is always a physically separate store of data transformed from the application data found in the operational environment. Due to this separation, a data warehouse does not require transaction processing, recovery, and concurrency control mechanisms. It usually requires only two operations in data accessing: initial loading of data and access of data. In other words, new data increase with time and thus they are continually added to a data warehouse to be used by decision-makers. In short, by creating a centralized data warehouse, using appropriate data analysis tools, and quickly developing software that supports decision-making, data warehouse enables decision-makers to acquire intended information at any time and use the acquired information as important references for supporting their decision-making [5].

2.2 Data Mining

According to Cabena, Hadjinian, Stadler, Verhees and Zamasi [4], data mining is a process extracting effective information, which is unknown previously, from a large database for executives to make critical decisions. Besides, Frawley, Paitetsky-Shapiro, and Matheus [9] define data mining as a process exploring in database unobvious, implicit, unprecedented information which may be useful. Thus, data mining is to use specific techniques, generalize and organize data from database and then excavate unknown, hidden information for executives to make decisions.

Data mining is the process whereby knowledge is discovered in a database and then implicit, previously unknown and potentially useful information is extracted from the database [9]. It enables the discovery of potentially useful information in

voluminous information in order to provide references for decision-makers. The whole process of data mining comprises data selection, preprocessing, conversion, data analysis, and interpretation and evaluation [10].

After understanding the definition of data mining and the objective thereof, we have to look into the steps leading to the knowledge discovery. Kleissner [15] suggests that a knowledge discovery cycle should comprise the following four steps: data selection, (2) data cleaning, (3) data conversion and meaning-giving, and (4) data mining. The aforesaid steps lead to knowledge discovery wherein the essence lies in mining target data in order to discover knowledge. Brachman et al. [3] believe that all the activities and processes in connection to exploration of knowledge are intended to find out useful patterns in those data, and then important causes of problems are identified in order to solve them, using the data mining algorithm as well as subsequent processing or re-processing of knowledge. After the discovery of knowledge, related experts have to evaluate and explain the extracted knowledge so as to ensure that the discovered knowledge will have genuine efficacy. Knowledge discovery processes are shown in Figure 3 [9].

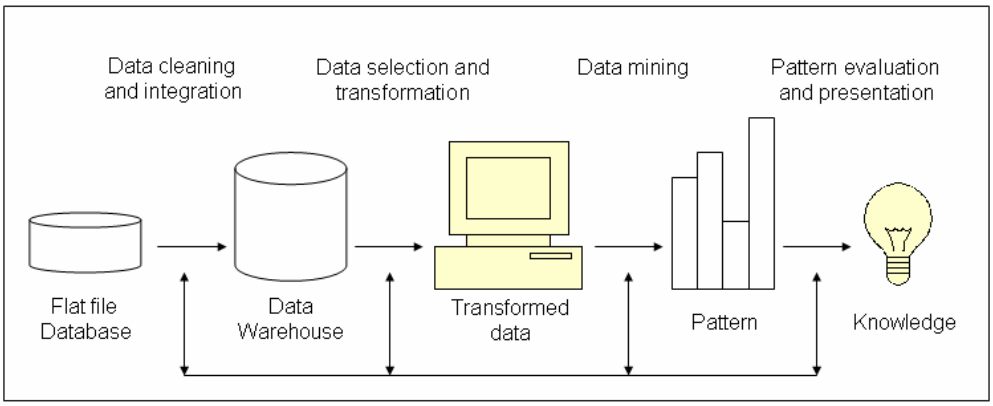


Figure 3. Knowledge discovery processes.

In the past, some researchers applied data mining technology to improve the quality for semiconductor manufacturing process. The brief descriptions are listed as follows.

1. Perry Lee [16] proposed data mining procedures for analyzing semiconductor manufacturing data for the purpose of manufacturing process monitoring and defect diagnosis. In particular, SOM is applied for clustering and decision tree is applied for feature extraction to analyze multi-dimensional semiconductor manufacturing data. We used real data from a fabrication to conduct two case studies for validation and found that this approach can effectively limit the scope for defect diagnosis and summarize the findings in specific decision rules. In addition, the researcher developed a new decision tree algorithm focused on target class while classification and implement the algorithm on windows platform. This prototype can be referenced while constructing completed data mining system for semiconductor manufacturing.
2. Cang-Ren Fan [8] proposed this system framework which uses data warehouse, on line analytical processing technique and the data mining engine with association rule algorithm. This data mining system is applied to discover machines combinations in DRAM semiconductor packaging factories. The research result had shown that the enhancement of yield is 4.43%.
3. Wei-Lin Tseng [22] proposed the research focusing on printed circuit board (PCB) industry to bring up the internal process quality diagnosis information system. The research utilizes two data mining defective analyzed models (See5 and PolyAnalyst) to build the diagnosis knowledge database to address the major causes of defective by using data from internal CAR. The research evaluated two data mining defective analyzed models and the results had shown the performance of See5 (accuracy rate: 91.66%) is better than that of PolyAnalyst (accuracy rate: 89.28%) in the case. The results from the PCB manufacturers demonstrate that the rule is a useful tool for diagnosis defect mode.
4. Cheng-Lung Huang [13] proposed the genetic programming (GP) and artificial

neural networks (ANN) systems models to predict the future production rate based on the historical production performance data in the DRAM semiconductor plants. After evaluating the two methods, the accuracy rates of GP and ANN are 79.25% and 76.68% respectively. Moreover, an application for the prediction of daily production rate for a DRAM wafer fabrication demonstrated the effectiveness of our approach in predicting production performance.

2.3 Traditional Semiconductor Packaging Processes

A semiconductor manufacturing process consists of IC (integrated circuit) design, mask production, wafer fabrication, wafer packaging and final testing. The whole manufacturing process can also be divided into front-end process and back-end process; wafer packaging and wafer testing are included in back-end process.

A semiconductor packaging process consists of four stages, shown in Figure 4. Different products undergo different processes and packaging patterns, which will then influence the processing methods. A processing method may be designated based on customer special requests [20].

1. Stage of wafer cleaning/mounting/saw: wafer income inspection, wafer mounting, wafer sawing/cleaning, post saw inspection, and sampling inspection by the QC (quality control) division.
2. Stage of die bonding/wire bonding/curing: die bonding, i.e. to glue dies on the lead frame one after one; epoxy curing, which places the semi-finished goods in an oven for curing; wire bonding and post bond inspection, and QC inspection.
3. Stage of molding/marking: molding, backside marking, which means to have a mark on the bottom of an IC; post mold cure, that sends an IC semi-finished goods for curing once more; trimming/dejunking, trimming the pins on leads; solder plating, soldering external pins on ICs; and QC inspection.

4. Stage of forming/packing/storage: top marking, which marks on the front side of ICs; post mold cure, which sends IC semi-finished goods to oven for curing again; forming/singulation; final visual inspection; QC inspection and then should be inspected by the QC division for packing compliance and conformity with customer requirements prior to storage.

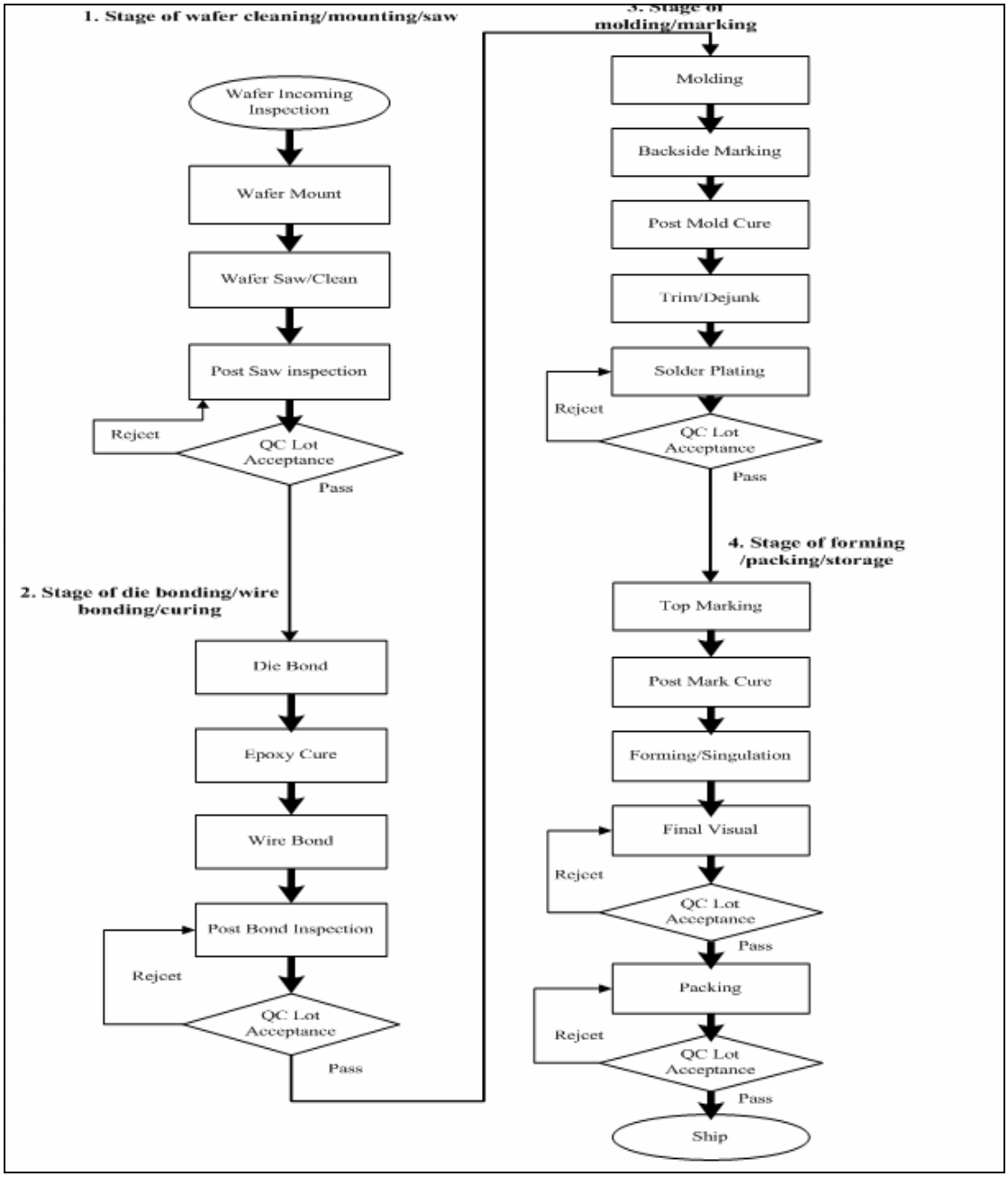


Figure 4. Traditional semiconductor packaging processes.

Chapter 3

Tape Carrier Package Technology

This chapter depicts the related semiconductor packaging processes. Section 3.1 is LCD driver IC packaging processes and section 3.2 is tape carrier package processes.

3.1 LCD driver IC Packaging Processes

To categorize different Liquid Crystal Display (LCD) driver IC back-end manufacturing process, it can be done by IC packaging types, and three types can be identified: Tape Carrier Package (TCP), Chip on Film (COF) and Chip on Glass (COG). Currently LCD driver IC mostly use TCP package, mobile phone LCD panel modules' driver IC mostly use COG package, and COF package is the future trend. Figure 5 [20] shows LCD driver IC back-end main process [12], and Table 3 [27] illustrates the advantages and disadvantages comparisons among TCP, COG and COF.

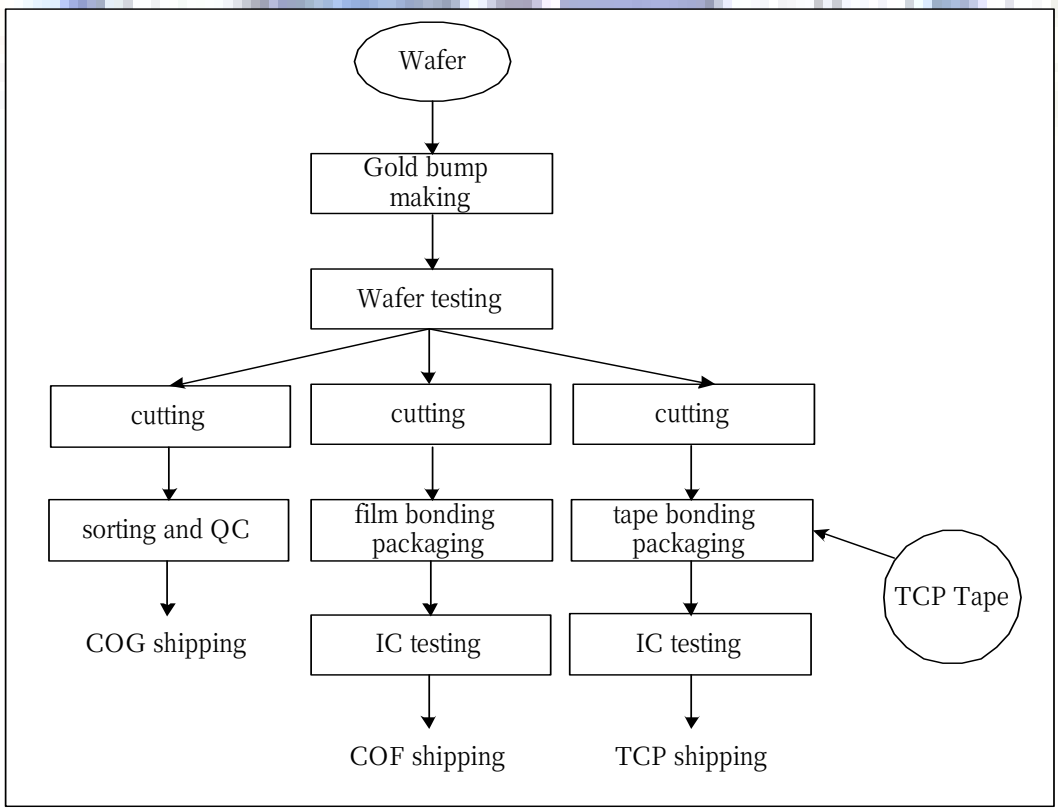


Figure 5. LCD driver IC back-end packaging processes.

Table 3 Advantages and disadvantages comparisons among TCP, COG and COF

	Technique	Required space	Packaging application	Major advantage	Major disadvantage
TCP	mature	medium	LCD driver IC for all sizes of displays	mature technology / low cost	line width limit at 40μm
COG	mature	large	LCD driver IC for small sized displays	mature technology / low cost	inability to decrease package volume
COF	immature	small	LCD driver IC for large sized displays	Fine pitch combined with active component	high cost

3.2 Tape Carrier Package Processes

The packaging process of semiconductor product is a very complicated process. Section 3.1 has already described the packaging process of LCD driver IC. This research is to investigate the defective factors of product quality by focusing on ultra-violet light (UV light), inner lead bonding (ILB), ILB QC, potting, curing, and marking processes, etc.

The original purpose of TCP's tape packaging technology was to replace the wire bonding packaging, along with the increase of IC's I/O numbers, and the trend of automated production. TCP technology became ever more mature, and is currently the main steam technology for large sized LCD driver IC packaging [27]. The overall TCP's process is shown in Figure 6. Moreover, the TCP's tape can be divided into several parts, such as sprocket hole, inner lead, outer lead, outer hole and polyimide tape. The sprocket hole is used for rewinding on the polyimide tape, the outer hole is used to fix outer leads, and inner/outer lead is used for chip bonding, illustrated in Figure 7 [6].

Figure 6. Tape carrier package processes.

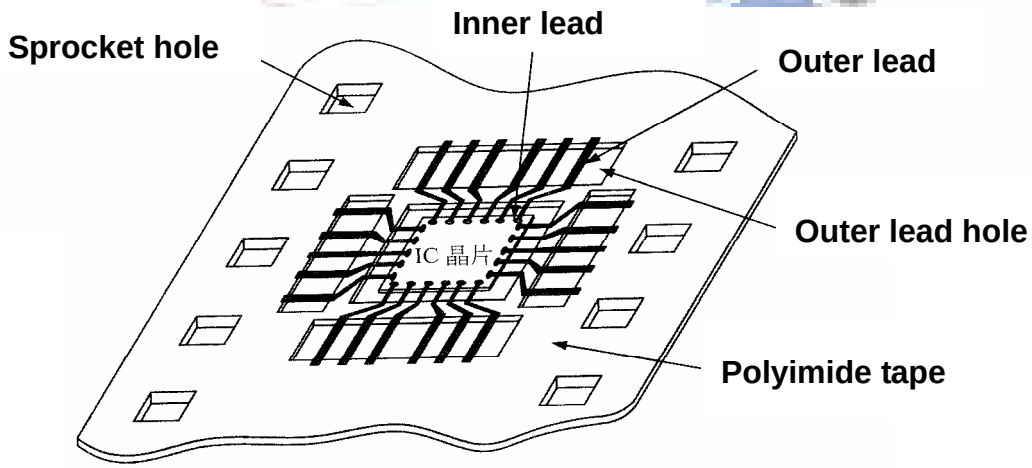


Figure 7. Tape carrier package tape.

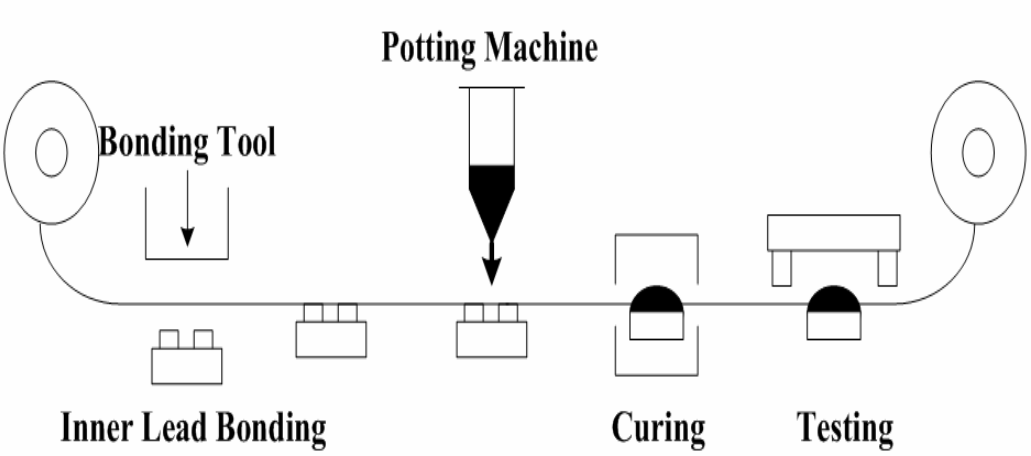


Figure 8. Illustration of tape carrier package processes technology.

The following is the LCD driver IC's packaging procedure covered in our study. The illustration of TCP's processes technology is shown in Figure 8 [25].

1. Ultra-violet light (UV light)

After the wafers are cut, they become individual separate chips, and still attached to the original UV tape film. Therefore, chips are following irradiated by UV light, to soften the tape film and the attached chips can be conveniently detached.

2. Inner lead bonding (ILB)

The inner lead bonding process involves in taking the inner lead of the tape and the gold bumps on the chips. The heated press was used to attach them together on the tape, and make connection points. Thus the process allows the chip to be connected to the circuit on the tape. The process is shown in Figure 9 [25].

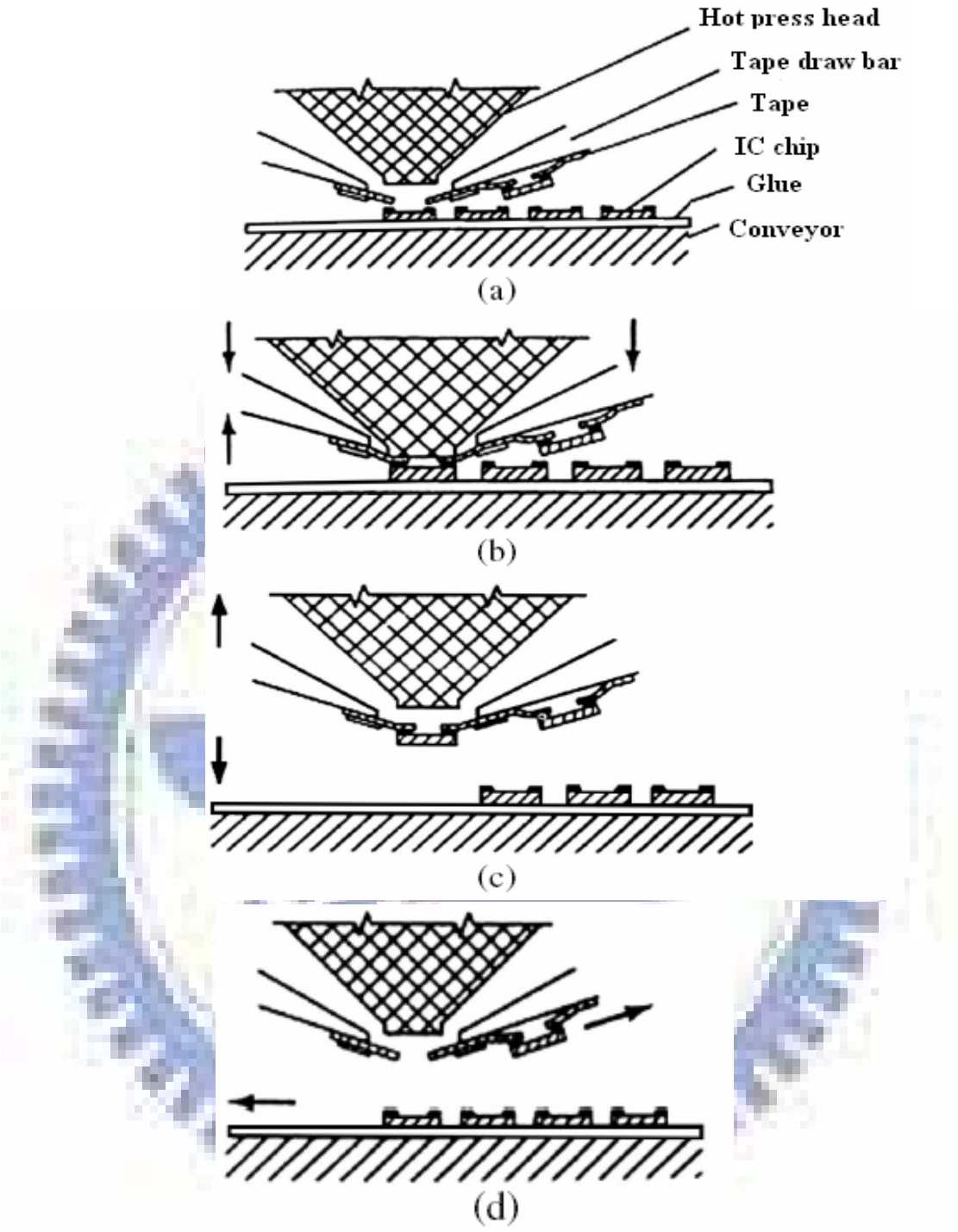


Figure 9. Illustration of inner lead bonding processes flow.

3. Inner lead bonding quality control

After making the inner lead bonding process, follows the quality control of the inner lead's completeness, pitch, lead and tape bond, etc.

4. Potting

The process applies a coating using resin sealing to provide protection for the chips to prevent damage by moisture, increase support of lead frames, and help to exchange heat etc.

TCP packaging utilizes liquid sealing material which is carried out on the potting machine, shown in Figure 10 [25]. This packaging uses glob-top method, but COF and COG can be divided, based on the requirement of packaging function, into dam and fill processes.

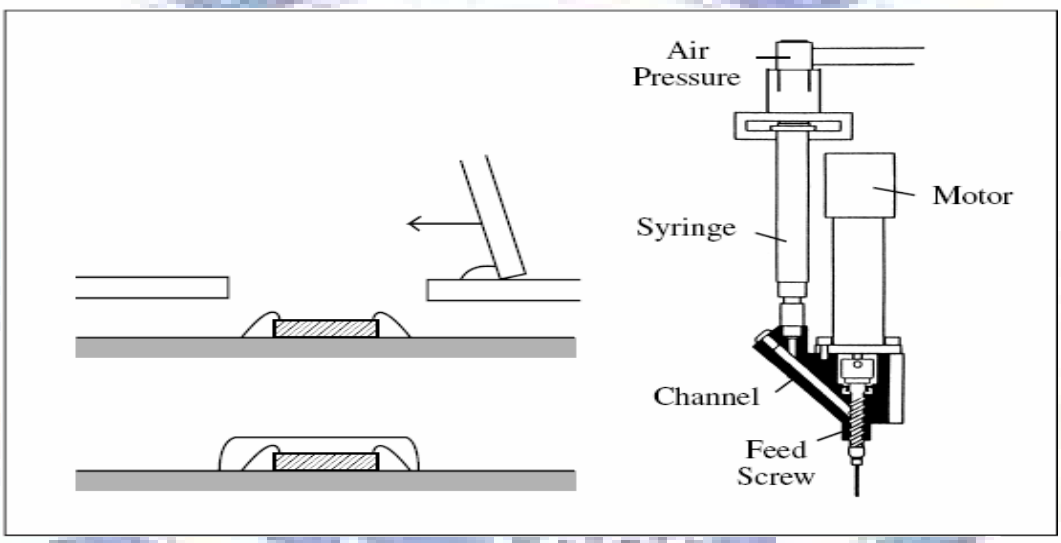


Figure 10. Major sealing processes using liquid sealing material.

5. Curing

When a product finishes with potting, in addition to the brief heating of the resin on the machine, the finished product needs to be put in a oven for further heating, and making the resin on the product completely moisture free and harden.

6. Marking

The marking process mainly deals with printing text onto the IC product's packaging, and the product specifications to be identified and the originality product process. The marking method is determined by the client needs.

7. Final testing

Final testing is done after the completion of the packaging process, using probes to connect with the product's outer leads; using electronic detection products to ensure the TCP packaged product reaches specification and sift defects.

8. Rewind (RW)

Tape bonding machine can be used to wind and bond the inner leads. The method of transport using fixed conveyers to transport soft chip type bearer, when the inner leads attached to the tape, reach position, the bonding head will lower down to press and bond the leads, thus completing a chip's RW process.

9. Packing

Classifying the finished products and packing them into client specified packing containers, and affix label and logos etc.

Chapter 4

System Design

With the understanding and observation of the semiconductor packaging process, we started the system design. This design procedure includes establishing system process flow, construction of data warehouse, building up data warehouse schema, and setting up kinds of data mining algorithms, etc.

4.1 Design a Framework for the Quality Improvement System

Based upon the data mining system, the framework for semiconductor product quality improvement system is illustrated in Figure 11. The function of each design element is shown in the following steps.

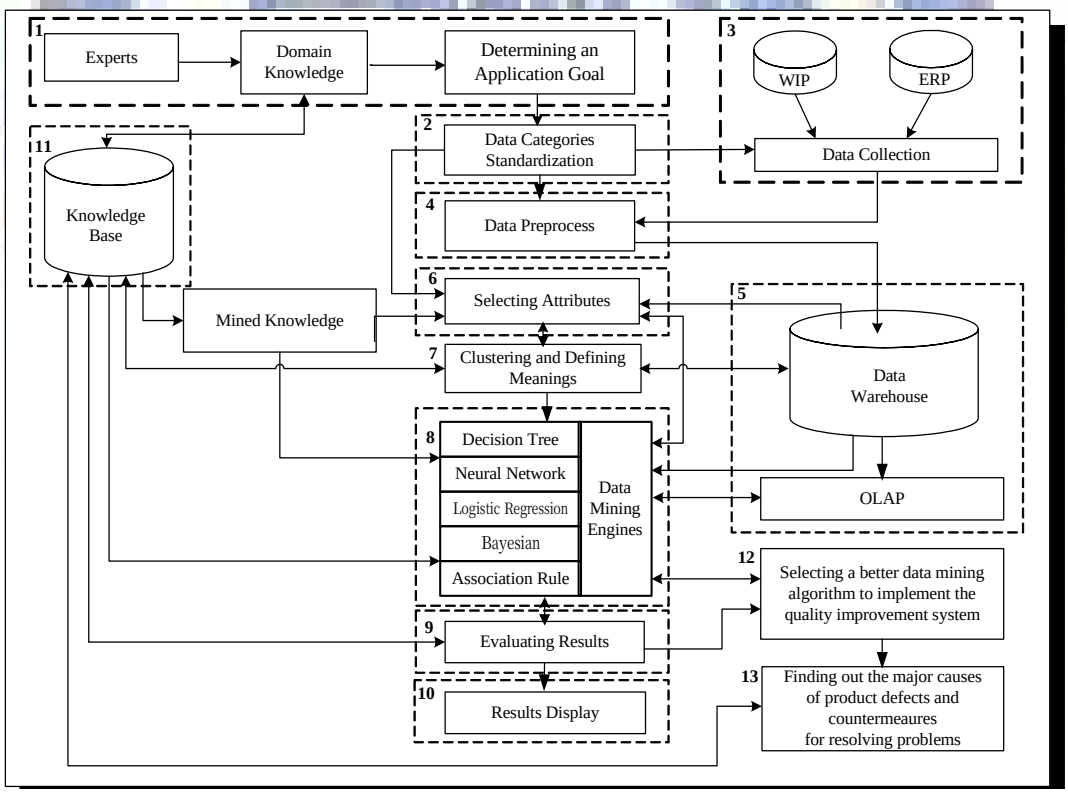


Figure 11. A framework of the quality improvement system for semiconductor packaging industry.

Step 1: A definite application goal is determined by domain knowledge provided by experts from various fields.

Step 2: The standardization of data categories allows for consistency in the subsequent gathering and management of data.

Step 3: In the stage of producing and gathering data, all on hand data from each category must be integrated.

Step 4: The preparatory management of data involves the processes of integration, transformation, refinement and filtration.

Step 5: When establishing the database and data warehouse, the time spent in the data mining search process is minimized.

Step 6: Analysis is carried out with using appropriate collections of data which is carefully selected by specialists in each particular field.

Step 7: Use clustering analysis to determine number of clusters from the data collected, and defines classification meaning after discussion with experts in the field.

Step 8: The design of the data mining engine is crucial and it serves as the core of the data mining engine in the entire framework.

Step 9: The patterns and rules that are discovered are evaluated and explained.

Step 10: The results of the mining process are displayed visually. This allows users to understand the results more easily.

Step 11: A knowledge base is used to store specialized knowledge and the knowledge obtained through the mining process. Thus, this knowledge base acts as a point of reference for making decisions.

Step 12: Comparing and evaluating the proposed algorithms, we apply a better algorithm as the core engine for data mining and establish quality improvement system.

Step 13: Use this better algorithm to analyze historical data and find out key product defect factors. The mined rules will be stored into database after analyzing the optimization of process parameters and countermeasures for resolving problems to provide related engineers for references.

4.2 Establishing a Data Warehouse System

We select WIP and ERP databases for this research, and filter out the production data needed for this study to build up data warehouse. The description of the procedures is as follows.

4.2.1 Establishing Data Warehouse Architecture

The data warehouse was viewed as the queryable source of data in the enterprise. However, a data mart is a logical subset of the complete data warehouse. This research utilizes the top to bottom model to establish a data warehouse system. The database of product quality improvement system serves as the origin of data extracted and transformed data, an integral and unified data warehouse system may be established. Data marts and data warehouses have a one sided relationship, in which data from a data warehouse flows into a data mart. This process can be divided into three levels from top to bottom: operational data, data warehouse and intelligent application, shown in Figure 12.

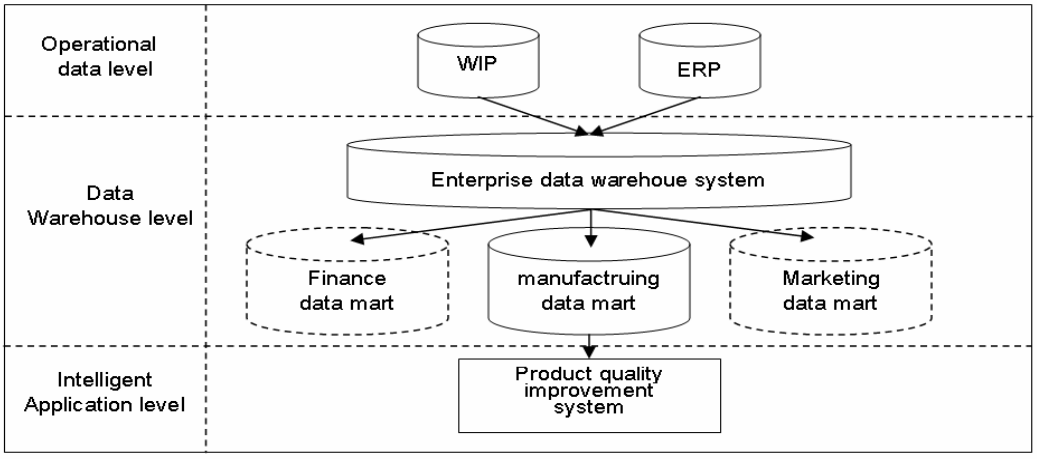


Figure 12. The data warehouse system established through the top-down model.

4.2.2 Establishing Data Warehouse Procedures

When integrating a product quality improvement system and a data warehouse, it is common to encounter the problems of inconsistent, incomplete and duplicate data. Therefore, the integration of product quality improvement system and data

warehouses involves the collection of different types of data from their original sources. This data are then placed in a data staging area where it undergoes such processes as the collection, selection, cleaning, transformation, combination, removal of duplicates indexing, etc. Next, the data is stored within a presentation server. Organized data stored in the presentation server can be used directly for the user to query data. At this point, users can carry out search tasks. The framework for integrating product quality improvement system and data warehouses is shown in Figure 13 [14]. The procedures for integrating product quality improvement systems and data warehouses include several distinct steps. The steps include extraction, cleaning, integration, transformation, loading into data warehouse, refresh the data warehouse, and export and store the out-of-date data at a fixed time.

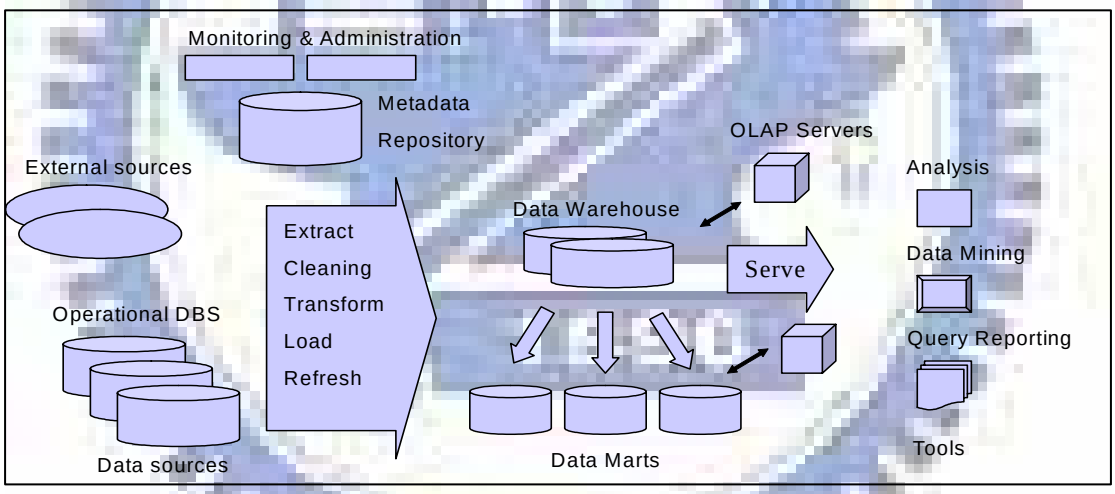


Figure 13. The framework for data warehouse procedures.

4.2.3 Establishing a Schema for Data Warehouse

When establishing a data warehouse system, the following frameworks can be utilized: star schema, snowflake schema and starflake schema [2]. These three types of schema are all based upon a fact table. The difference between them is their mutual relationship with external dimension tables. The dimension table in the star schema merely creates a connection with the fact table, while different dimension tables have no relationship with each other. This research project utilizes the starflake schema in

designing the schema for data warehouse. This schema is based upon the manufacturing fact table, a time dimension table, a lot dimension table, a product dimension table and a quality dimension table. The quality dimension table is composed of a man dimension table, a machine dimension table, a material dimension table and a method dimension table, illustrated in Figure 14.

4.2.3.1 Establishing a Fact Table

The real data that we need is placed in the fact table. The data in this table cannot be altered; we may only add new information. Moreover, this table includes an index key related to a dimension table. When designing a fact table, several factors must be taken into consideration, shown as the following seven steps.

1. Determining which data is in the fact table and which data is in the dimension table.
2. Deciding the data warehouse period for all functions to achieve a balance point between high-speed search capacity and data storage capacity. A long period of time is not necessarily positive attributes. In fact, the more precise the real data are better the data warehouse can be. The time periods established for the data warehouse in this research project include a two year period to measure trends in quality data, a six month period to analyze quality data and a one year period to analyze the positioning of quality data.
3. Determining a principle to be used in statistical sampling for all functions. Only part of the real data should be placed in the data warehouse. Next, collected data is calculated according to the determined sampling principle.
4. Determining which fields are included in the fact table and eliminate unneeded data occupying these fields. For example: status display fields, storage result fields and certain fields used as internal references.

5. To save space for significant data effectively, the size of fields included in the fact table should be minimized.
6. Determining whether or not to use an intelligent key to speed up the data search process.
7. To include time in the fact table, there are three points should be taken into account: the actual storage time, storage interval (fixed time points), and the storage period of time, illustrated as Figure 15.

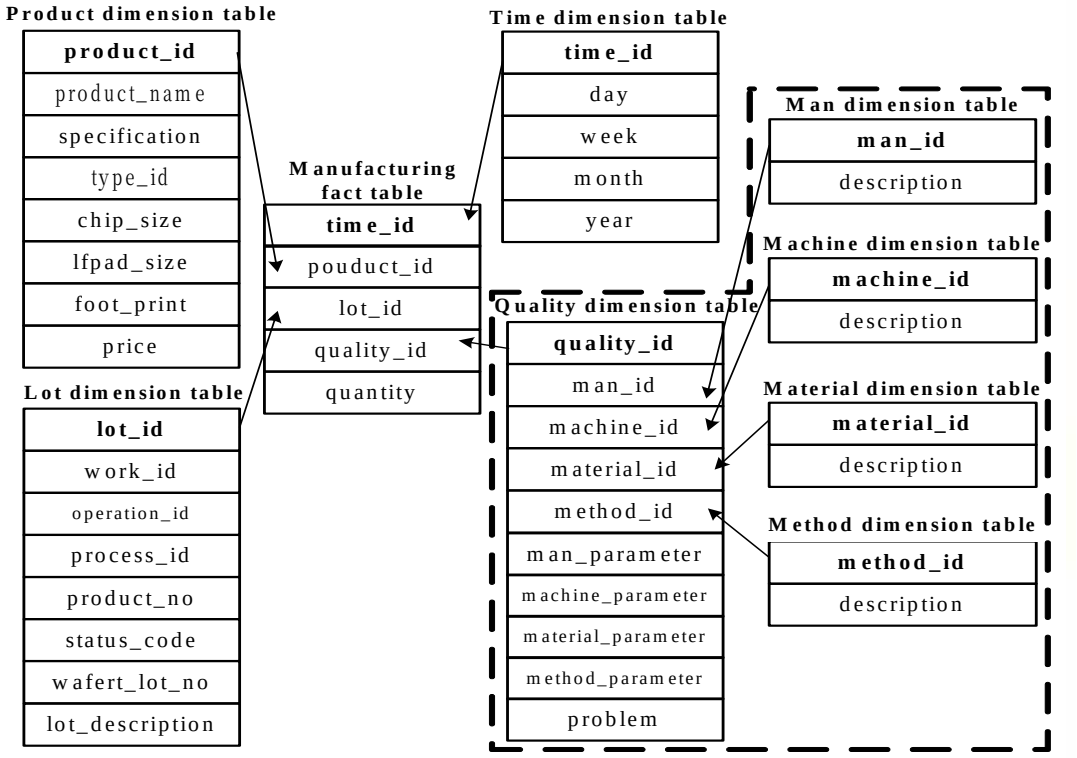


Figure 14. A starflake schema for the data warehouse.

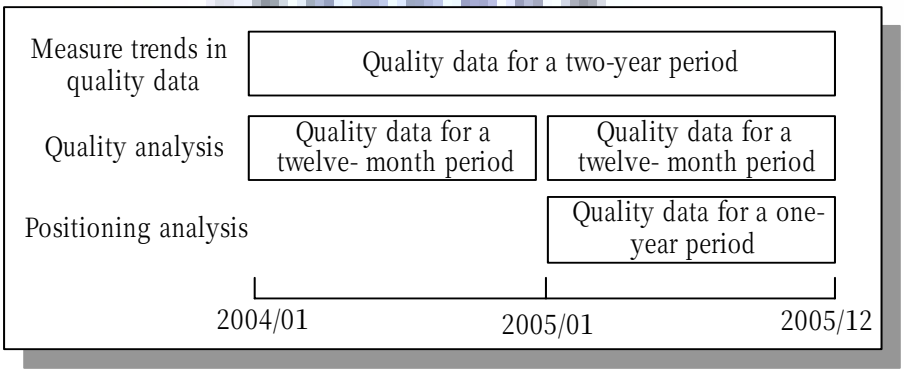


Figure 15. Time periods for the data warehouse.

4.2.3.2 Establishing Dimension Tables

Dimension table is used as a reference to fact table, when necessary, complex descriptions can be divided into several small parts, illustrated in Figure 14. During the initial setup stage, it is essential to assure that the dimension table's primary key will not be changed in any way.

If the primary key changes, the fact table will also changes. The dimension table is set up through a process of de-normalization. For example, when the quality of factory product is analyzed, new dimension tables can be established and used within tables to address related quality considerations. Therefore, the quality dimension table is related to man, machine, material, and method dimension tables. A dimension table will store a lot of duplicate data; when using the dimension table, it is not necessary to combine such data. Thus, space is sacrificed in the interest of time.

4.2.4 Establishing a Multidimensional Model

When analyzing data, multiple dimensions are brought together as one point of consideration. This process is called a multidimensional data model. Data warehouse systems may include many data cubes. Each data cube may be the product of different dimension and fact tables. The OLAP operations in data cubes include rollup, drilldown, slice, dice, and pivot. A data cube may be an N-dimensional data model. In order to provide an even wider range of search capabilities, this research uses the three dimensions of month, problem and quantity to construct a three-dimensional data cube model, shown in Figure 16.

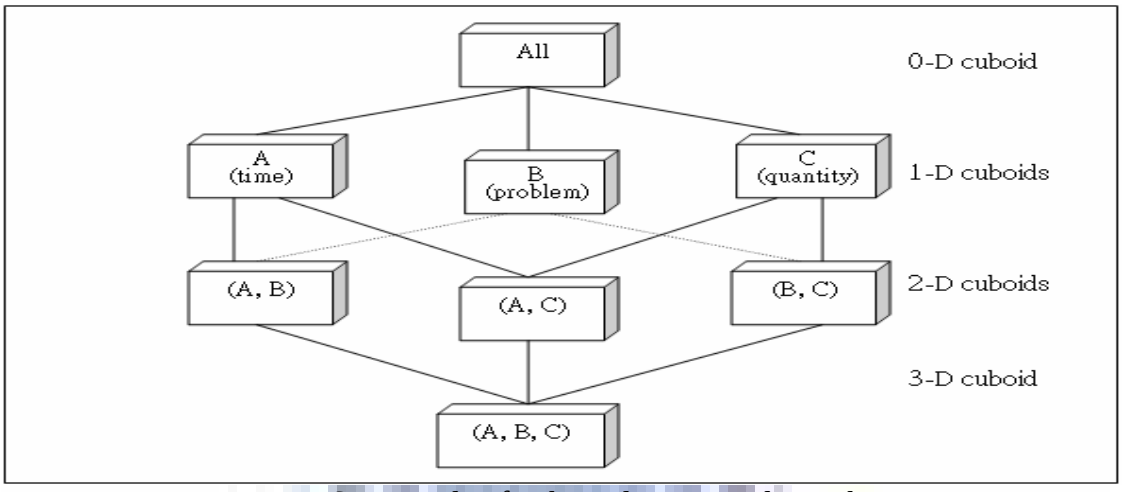


Figure 16. Example of a three-dimension data cube.

4.3 Integrating Decision Analysis and Data Mining Systems

After completing the construction of data cubes, it is possible to integrate decision-making analysis and the data mining system shown in Figure 17 [7]. The goals of integration are to allow OLAP analysis results to supply the knowledge base within the data mining system, thus providing analysis information to the data mining system and creating a point of reference for data mining tasks. OLAP technology is able to blend together people's observations and intelligence within the data mining system, thus improving the speed and depth at which data is excavated. Furthermore, the intelligence discovered by the data mining system acts as a guide in OLAP analysis tasks, and increases the depth of analysis. As a result, information left unearthed by the OLAP is extremely complex and delicate in nature.

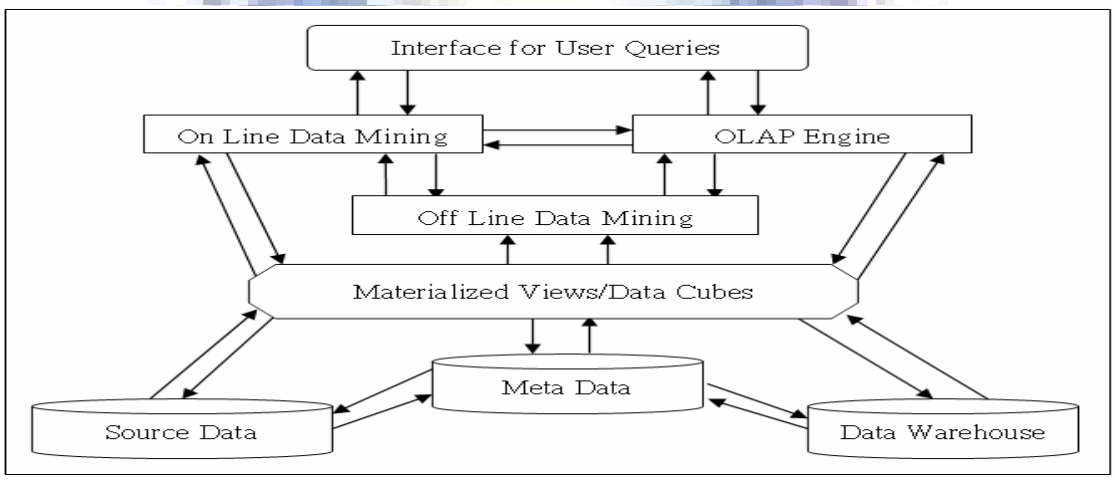


Figure 17. Integration of decision-making analysis and data mining systems.

4.4 Establishing Data Mining Systems

Clustering algorithm is an unsupervised learning technology which we can uncover the internal structures hidden in the data and categorize the data according to the similarities. Classification algorithms, through the use of external variables, examine the properties of the cases for predicting the occurrence of the future events.

In the data mining aspect, this study adopted the suitable clustering and five classification algorithms contained in Microsoft SQL Server 2005 that suitable for this study to conduct the experimental analysis and comparison so as to find out a better classification algorithm. This is in turn used to predict the cause of abnormal product and find out the key factors thereof. The algorithms [10] [17] [26] [34] are described from section 4.4.1 to 4.4.2.5.

4.4.1 Set Up of the Clustering Data Mining Engine

Clustering algorithm theory

In the processing of large amount of data, if n variables are involved, we will have n dimensions. And each case can be treated as a point projected onto the n-dimensional space. Thus, similarity can be described using the concept of distance. Cases with similar geometrical distance can be treated as having higher similarity. The distribution density of cases will then form a cluster that is similar to an archipelago.

Although K-means are well-known and popular, there are quite a few limitations in its application. K-means is using distance as the basis of similarity. As there is only one closest cluster, these clusters are adjacent and not overlapping. When this characteristic faces two very similar clusters at boundary situation, a slight deviation could change cluster A to cluster B. To solve these problems, many improved cluster algorithms exist. The traditional K-means algorithm is called “rigid cluster” as there is no overlapping between clusters. On the contrary, each case will belong to all clusters

at the same time if the concept of absolute distance is changed to attribute probability. The only thing depends on the probability. Therefore, there is no such concept of cluster boundary. We call such cluster algorithm as “soft cluster”.

The expectation maximum algorithm, usually called E-M algorithm, is an example of soft clustering. The calculation flow of E-M algorithm is very similar to K-means algorithm. The only difference is that E-M used Gaussian distribution as the distance function.

Gaussian distribution is also called normal distribution and its functional value can be represented as follows.

$f_X(x) = \frac{1}{\sqrt{2\pi}\delta} e^{\frac{(x-\mu)^2}{2\delta^2}}, -\infty < x < \infty$ where μ is the average of X and δ is the standard deviation of X.

Clustering algorithm

Presently, the cluster quantity parameter of Microsoft clustering algorithm, and the default value is 1. If it is set to 0, the system will automatically detect possible cluster amount. We used Figure 18 [30] as an example to explain the processing steps of E-M algorithm.

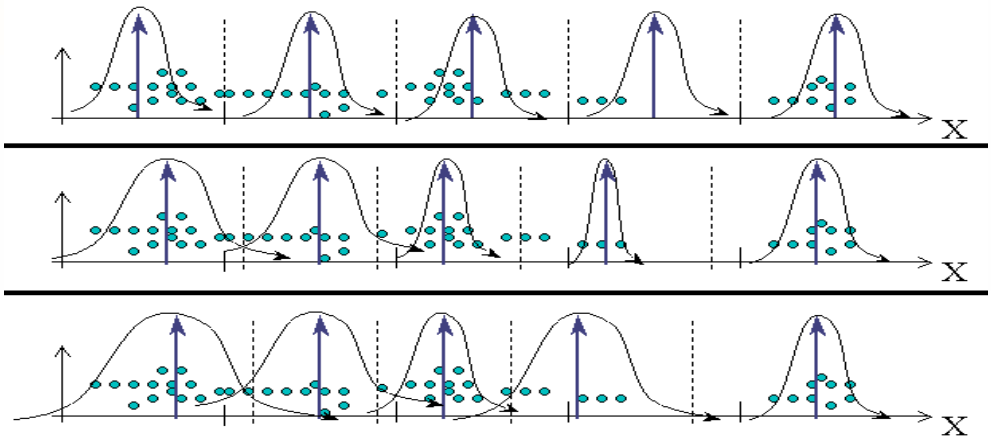


Figure 18. Calculation flow of E-M algorithm.

Step1. After setting the cluster number as *k*, system will first randomly select *k* cases as random seeds.

Step2. Each random seed has its own “identical” default Gaussian distribution. In each case, the probability will be calculated based on the Gaussian distribution of each random seed. And based on this probability, a random seed is assigned the nearest (highest probability value). Therefore, the broken line cluster boundary of the first portion is generated shown in Figure 18.

Step3. Calculate the mass center based on the case of the same cluster. Use this mass center to replace the previous random seed as the cluster center. At the same time, recalculate the new Gaussian distribution probability based on the surrounding case density of the mass center. It can be discovered that clusters with more widely spread case numbers has wider Gaussian distribution, whereas the more clusters with surrounding cases are, the more scattered curve has. It will become more concentrated if the situation is reversed.

Step4. Similarly, recalculate the probability value of each case away from the new cluster center based on the new Gaussian distribution, and then determine each case belongs to which cluster based on the value of its probability. Hence, the member of each cluster will have a drastic reshuffle and re-demarcate cluster boundary.

Step5. This is repeated until the members of each cluster do not change any more.

4.4.2 Classification and Predictions Procedures

During the construction of prediction procedures, we can divide the data set into three parts composed of training set, validation set and testing set. The operational procedures are listed as follows.

1. First, the training set is used to establish the rough model and discover the patterns of predicted values.
2. Next, owing to using training set is not accurate enough, so we will use validation set to conform to the generalization of the model and predict the undiscovered

patterns.

3. The so-called testing set, another data set which is different from the training set and validation set, can be used to predict the efficiency of the target model.
4. Therefore, a better model will generated from above three steps. It can be used to predict the other data to produce results for the decision makers.

The overall procedures are illustrated as Figure 19.

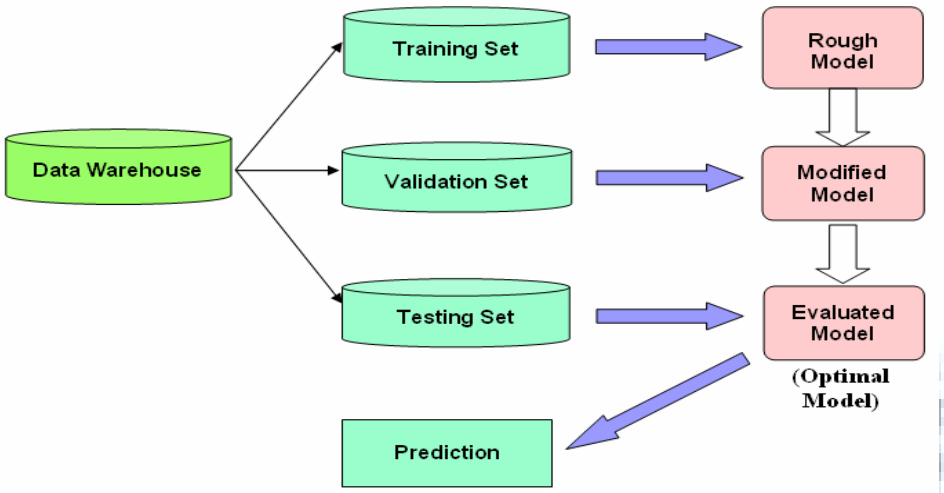


Figure 19. Setting up model and applying procedures.

4.4.2.1 Set Up of the Decision Tree Data Mining Engine

Decision tree algorithm theory

Decision tree is using a tree branch structure to generate classification rules. We explain the process of growth of a decision tree, illustrated as Figure 20.

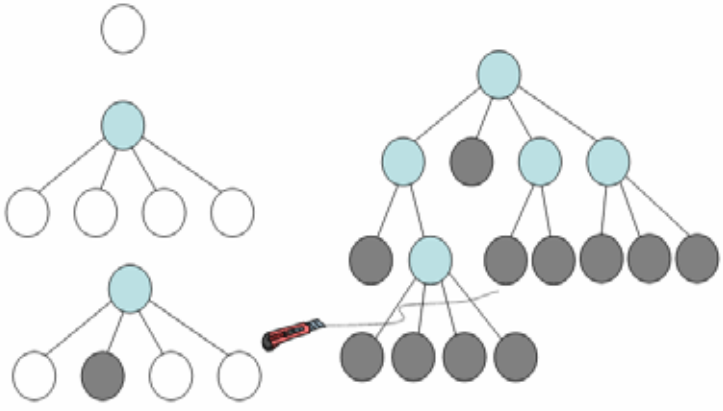


Figure 20. Process of growth of decision tree rules.

Step1. Use the whole data as the root node.

Step2. The algorithm calculates each variable at each branch, and then generates branch and child-node based on better variable.

Step3. Assign classification results based on the case distribution probability of each child-node and generate classification probability.

Step4. Decision tree continues to grow and finally pruning technique is used to prune off unneeded rules.

Decision tree algorithm

The computation steps for decision tree are provided as follows [10] [17].

Step 1. Prepare previously classified training data.

Step 2. Establish a decision tree node. Determine whether or not this node is a leaf node, or calculate information gain for the test attribute. The calculation method is shown as follows in steps 3-5.

Step 3. The expected information of the classified data samples selected for calculation: Let S be a set consisting of s data samples. Suppose the class label attribute has m distinct values defining m distinct classes, C_i (for $i = 1,\ldots,m$). Let s_i be the number of samples of S in class C_i. The expected information needed to classify is given by

$$I(S_1,S_2,\ldots,S_m) = -\sum_{i=1}^{m} P_i \log_2(P_i)$$

Where P_i is the probability that an arbitrary sample belongs to class Ci and is estimated by s_i/s.

Step 4. The expected information of the test attribute selected for calculation: Let attribute A have v distinct values,$\{a_1,a_2,\ldots,a_v\}$. Attribute A can be used to partition S into v subsets, $\{S_1,S_2,\ldots,S_v\}$,where S_j contains those samples in S that have value a_j of A. Let s_{ij} be the number of samples of class C_i in a subset

S_j. The entropy, or expected information based on the partitioning into subsets by A, is given by

$$E(A)=\sum_{j=1}^{v}\frac{S_{1j}+....+S_{mj}}{S}I(S_{1j},S_{2j},...,S_{mj})$$

$$I(S_{1j},S_{2j},....,S_{mj})=\sum_{i=1}^{m}P_{ij}\log_2(P_{ij}),P_{ij}=s_{ij}/|s_j|$$

Where P_{ij} is the probability that a sample in S_j belongs to class C_i.

Step 5. Calculate the information gain of the selected test attribute: The encoding information that would be gained by branch on A is

Gain Ratio (A) = ($I(s_1,s_2,\ldots,s_m)$-$E(A)$)/*Split Gains*.

Split Gains are segmented subsets; the computation method is given by

$$I(S_{1j},S_{2j},.....S_{mj})=-\sum_{i=1}^{m}P_{ij}\log 2(P_{ij}),P_{ij}=s_{ij}/|s_j|$$

Step 6. Repeat steps 2-5 until the information gain of the test attributes are completely calculated.

Step 7. Select the test attribute with the highest information gain to act as the node of partition for the decision tree.

Step 8. To complete set up of the decision tree, follow this sequence of steps to find test attribute nodes at each level.

4.4.2.2 Set Up of the Neural Network Data Mining Engine

Neural network algorithm theory

Microsoft Neural Network algorithm calculates probabilities for each possible state of the input attribute when given each state of the predictable attribute to predict an outcome of the predicted attribute, based on the input attributes. The Microsoft neural network algorithm uses a back-propagated delta rule network, composed of three layers of neurons. These layers are an input layer, a hidden layer, and an output layer. The hidden layer is one. Each neuron has a simple non-linear function assigned

to it, called the activation function, which describes the relevance or importance of a particular neuron to the layer of a neural network. Hidden neurons use a hypertangent function for their activation function, whereas output neurons use a sigmoid function for their activation function. Both functions are nonlinear, continuous functions that allow the neural network to model nonlinear relationships between input and output neurons.

Parameter setting by neural network algorithm

Neural network is a complicated algorithm than the other algorithms we proposed. Therefore, we explain it in a detail way. The steps are heavily influenced by the values that you specify for the algorithm parameters. The algorithm first evaluates and extracts training data from the data source. A percentage of the training data, called the validation data, is reserved for use in measuring the accuracy structure of the resulting model. During the training process, the model is evaluated against the holdout data after each iteration over the training data. When the accuracy of the model no longer increases, the training process is stopped. The values of the *SAMPLE_SIZE* and *HOLDOUT_PERCENTAGE* parameters are used to determine the number of cases to sample from the training data and the number of cases to be put aside for the holdout data. The value of the *HOLDOUT_SEED* parameter is used to randomly determine the individual cases to be put aside for the holdout data.

The algorithm creates a single network that represents all such attributes. If the mining model contains one or more attributes that are used for both input and prediction, the algorithm provider constructs a network for each such attribute. If the number of input or predictable attributes is greater than the value of the *MAXIMUM_INPUT_ATTRIBUTES* parameter or the *MAXIMUM_OUTPUT_ ATTRIBUTES* parameter, respectively, a feature selection algorithm is used to reduce the complexity of the networks that are included in the mining model. Feature

selection reduces the number of input or predictable attributes to those that are most statistically relevant to the model.

The maximum number of states that is supported in either case depends on the value of the *MAXIMUM_STATES* algorithm parameter. If the number of states for a specific attribute exceeds the value of the *MAXIMUM_STATES* algorithm parameter, the most popular or relevant states for that attribute are chosen, up to the maximum, and the remaining states are grouped as missing values for the purposes of analysis. The algorithm then uses the value of the *HIDDEN_NODE_RATIO* parameter when determining the initial number of neurons to create for the hidden layer.

The Microsoft neural network algorithm supports continuous and discrete types, and also supports several parameters that affect the performance and accuracy of the resulting mining model. The descriptions of each parameter and the better parameters setting by the study are shown in Table 4 [34].

Table 4 Parameters setting by neural network algorithm

Parameter	Description
HIDDEN_NODE_RATIO	Specifies the ratio of hidden neurons to input and output neurons. The following formula determines the initial number of neurons in the hidden layer:HIDDEN_NODE_RATIO * SQRT(Total input neurons * Total output neurons). The default value is 4.0.
HOLDOUT_PERCENTAGE	Specifies the percentage of cases within the training data used to calculate the holdout error, which is used as part of the stopping criteria while training the mining model. The default value is 30.
HOLDOUT_SEED	Specifies a number that is used to seed the pseudo-random generator when the algorithm randomly determines the holdout data. If this parameter is set to 0, the algorithm generates the seed based on the name of the mining model, to guarantee that the model content remains the same during reprocessing. The default value is 0.
MAXIMUM_INPUT_ ATTRIBUTES	Determines the maximum number of input attributes that can be supplied to the algorithm before feature selection is employed. Setting this value to 0 disables feature selection for input attributes. The default value is 255.
MAXIMUM_OUTPUT_ ATTRIBUTES	Determines the maximum number of output attributes that can be supplied to the algorithm before feature selection is employed. Setting this value to 0 disables feature selection for output attributes. The default value is 255.
MAXIMUM_STATES	Specifies the maximum number of discrete states per attribute that is supported by the algorithm. The default value is 100.
SAMPLE_SIZE	HOLDOUT_PERCENTAGE is set to 30, the algorithm will use either the value of this parameter, or a value equal to 70 percent of the total number of cases, and whichever is smaller. The default value is 10000.

Neural network algorithm

The neural network algorithm uses the back-propagation method to learn for classification, works as follows [26].

Step 1. Exist the training samples, the learning rate *l*, and a multilayer feed-forward network, shown in Figure 21 [26].

Step 2. The algorithm works as the following.

(1) Initiates all weights and biases of network;

(2) while the terminating condition is not reached {

(3) for each training sample *X* in samples {

(4) // Propagate the inputs forward

(5) for each hidden or output layer unit *j* {

(6) $I_j = \sum_i w_{ij} O_i + \theta_j ;$// compute the net input of unit j with respect to the previous layer, *i*

(7) $O_j = \frac{1}{1+e^{-I_j}} ;\}$// compute the output of each unit *j*

(8) // Back-propagate the errors:

(9) for each unit j in the output layer

(10) $Err_j = O_j (1-O_j)(T_j - O_j)$; //compute the error

(11) for each unit j in the hidden layers, from the last to the first hidden layer

(12) $Err_j = O_j (1-O_j) \sum_k Err_k w_{jk} ;$// compute the error with respect to the next higher layer, *k*

(13) for each weight w_{ij} in network{

(14) $\Delta w_{ij} = (l) Err_j O_i ;$// weight increment

(15) $w_{ij} = w_{ij} + \Delta w_{ij} ;\}$// weight update

(16) for each bias θ_j in network{

(17) $\theta_j = (l)Err_j;$// bias increment

(18) $\theta_j = \theta j + \Delta\theta j;\}$// bias update

(19) }}

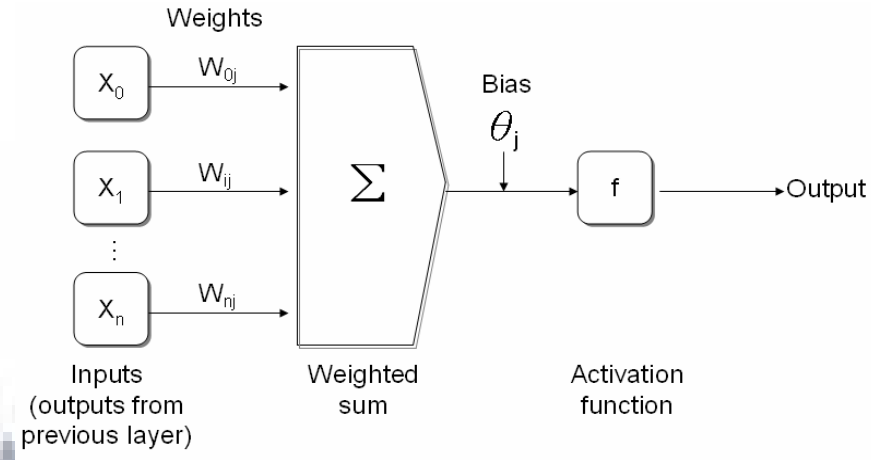


Figure 21. Feed-forward network flow chart.

4.4.2.3 Set Up of the Logistic Regression Data Mining Engine

Logistic regression algorithm theory

As the traditional linear model does not process probability distribution, we must select other nonlinear function as the approximation value of probability distribution. The selected different probability distribution function represents different regression algorithm. Taking logistic regression as an example, the function can use Logit function. Logit function can be described using $Y = 1/(1+e^{-x})$, shown in Figure 22. When X approaches infinite, e^x will approach infinite and Logit function will approach 1; when X approaches 0, Logit function will approach 0.

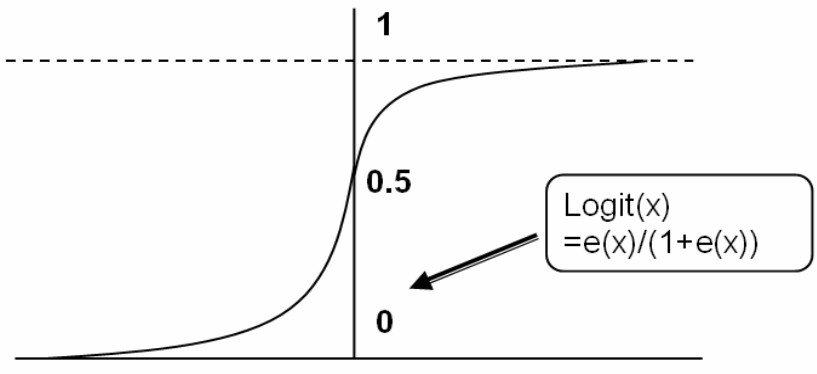


Figure 22. Logit function.

As we want to predict is the probability of occurrence of events, the probability *P* is used to represent predicted variable. The logistic regression is expressed by the following equation [26]:

$$P = 1/(1+e^{-z}) \quad Z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

Exchanging the numerator and denominator we obtain:

$$e^{-z} = 1/P - 1 = (1-P)/P$$

Using logarithm on both sides of the equation we have:

$$In(P/(1-P)) = Z \quad Z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

Through this transformation, the original equation of nonlinear model is converted into linear equation and most importantly is to calculate *P/(1-P)*. The equation *P/(1-P)* is exactly the probability of the event occurring divided by the probability of the event not occurring, which we called "Odd Ratio". As logistic regression is a nonlinear function, it is not possible to find the solution using traditional elimination method but must use iteration method to find approximated solution. In the Microsoft logistic regression, the method of estimating equation parameters is to use the mechanism of reiteration learning of neural network. Through back-propagation network model, reiterative modification method is used to obtain the coefficients of the equation.

Microsoft logistic regression algorithm is a kind of variation of Microsoft Neural Network algorithm and when the neural network completely does not have any hidden layer neuron (neural network parameter HIDDEN_NODE_RATIO is 0) the algorithm will become logistic regression. Hence, Microsoft logistic regression can handle classification variables different from the traditional statistical algorithm. It also can handle the prediction of continuous values.

Logistic regression algorithm

Logistic regression assumes that the Logits, logarithm of the odds ratios, are

linear with regard to the features [24].

Step 1. Until the stopping criteria are reached, for each pattern x_k in the training set, compute the logistic output y_k:

$$y_k = \frac{1}{(1+e^{-\Sigma})}$$

where $\Sigma_k = w_0 + \sum_{i=1}^{Ninputs} w_i x_{ki}, \quad -\infty < \Sigma < +\infty, \quad w_i = \text{free parameters}$

Step 2. Compute the gradient of the entropy error with respect to each w_i due to x_{ki}:

$$E_k = d_k \cdot \ln(1/y_k) + (1-d_k) \cdot \ln(1/(1-y_k))$$

$$\frac{\delta E_k}{\delta wi} = \frac{\delta \Sigma}{\delta wi} \cdot \frac{\delta_{yk}}{\delta \Sigma} \cdot \frac{\delta E_k}{\delta_{yk}}$$

$$= x_{ki} \cdot yk(1-yk)\left(\frac{y_k - d_k}{y_k(1-y_k)}\right)$$

$$= x_{ki}(y_k - d_k)$$

Step 3. Compute the change in weights:

$$\Delta W_i = (-\eta)\frac{\delta E_k}{\delta w_i} = \eta x_{ki}(d_k - j_k)$$

4.4.2.4 Set Up of the Bayesian Data Mining Engine

Bayesian algorithm theory

Bayesian theory is derived from conditional probability. The so-called conditional probability refers to “the conditional probability of occurrence of event Y under the condition of X equals to the probability of X and Y occurring at the same time divided by the probability of X occurring”. Described in mathematical equation becomes:

P(Y | X)=P(XUY)/P(X)

In other words, the probability of X and Y occurring at the same time equals to the occurrence of Y multiplied by the conditional probability of X divided by Y.

Described in mathematical equation becomes:

$$P(X \cap Y)=P(Y)*P(X \mid Y)$$

Combining the two equations we have:

$$P(Y \mid X)=P(XUY)/P(A)=P(Y)*P(X \mid Y)/P(X)$$

And how to calculate when there are more than one input variables? Here we want to assume that all inputs are independent event. Therefore, the probability of meeting all conditions are to multiply the probability of each condition. In the real world, input variables usually are not independent of each other. However, this assumption will not seriously affect the accuracy of Bayesian probability classification.

Bayesian algorithm

The naïve Bayesian classifier, or simple Bayesian Classifier, works as follows [21].

Step 1. Each data sample is represented by an n-dimensional feature vector, $X=(x_1,x_2,\ldots,x_n)$, depicting n measurements made on the sample from n attributes, respectively, $A_1,A_2,\ldots,A_n$.

Step 2. Suppose that there are m classes, $C_1,C_2\ldots,C_m$. Given an unknown data sample, X（i.e., having no class label）, the classifier will predict that X. That is, the naïve Bayesian classifier assigns an unknown sample X to the class C_i if and only if

$P(C_i|X) > P(C_j|X)$ for $1 \le j \le m, j \ne i$.

Thus we maximize $P(C_i|X)$. The class C_i for which $P(C_i|X)$ is maximized is called the maximum posterior hypothesis. By Bayesian theorem,

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{P(X)}$$

Step 3. As $P(X)$ is constant for all classes, only $P(X \mid C_i)P(C_i)$ need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, htat is, $P(C_1)=P(C_2)=\ldots=P(C_m)$, and we

would therefore maximize $P(X \mid C_i)$. Otherwise, we maximize $P(X \mid C_i)\, P(C_i)$. Note that the class prior probabilities may be estimated by $P(C_i)=s_i/s$, where s_i is the number of training samples of class C_i, and s is the total number of training samples.

Step 4. Given data sets with many attributes, it would be extremely computationally expensive to compute $P(X|C_i)$. In order to reduce computation in evaluation $P(X|C_i)$, the naïve assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the sample, that is, there are no dependence relationships among the attributes. Thus,

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

The probabilities $P(x_1|C_i)$, $P(x_2|C_i)$…, $P(x_n|C_i)$,…,$P(x_n|C_i)$ can be estimated from the training samples, where

If A_k is categorical, then $P(x_k|C_i)=s_{ik}/s_i$, where s_{ik} is the number of training samples of class C_i having the value x_k for A_k, and si is the number of training samples belonging to C_i.

Step 5. In order to classify an unknown sample X, $P(X|C_i)P(C_i)$ is evaluated for each class C_i. Sample X is then assigned to the class C_i if and only if

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \quad for\ 1 \le j \le m, j \ne i.$$

In other words, it is assigned to the class C_i for which $P(X|C_i)P(C_i)$ is the maximum.

4.4.2.5 Set Up of the Association Rule Data Mining Engine

Association rule algorithm theory

The definition of association rule by Agrawal [1] is as follows.

Assume that $I = \{I_1, I_2, \cdots, I_m\}$: I can be treated as the assembly of m items, $D = \{t_1, t_2, \cdots, t_n\}$: D is the overall assembly of the transaction of n clients,

where $t_i = \{I_{i1}, I_{i2}, \cdots, I_{ik}\}$: t_i represents the transaction data of the ith client.

The representing equation of the Association Rule is

"If condition then result", or

"X⇒Y" where X and Y are called itemsets.

There are two important parameters in the association rule, or support and confidence, where support refers to numbers of X itemset and Y itemset appearing in the D transaction assembly at the same time. From the viewpoint of probability, support is the probability of X and Y events happening at the same time. Below is the representing equation:

Support(X⇒Y)=P(XUY)

Confidence refers to the number of times of X itemset and Y itemset appearing in the D transaction assembly at the same time divided by the number of times of X itemset appearing in the D transaction assembly. Looking from viewpoint of probability, Confidence is the probability of occurrence of Y under the condition of X event occurring. Below is the representing equation:

Confidence(X⇒Y)=P(Y | X)

Apriori algorithm

The association rule works as follows.

Step 1. First, it is necessary to define the Minimum Support and Minimum Confidence.

Step 2. Apriori Algorithm makes use of the concept of candidate itemset. The first generated itemset is called candidate itemset. If the Support of candidate itemset is larger or equals to the Minimum Support, the candidate item set is a large itemset.

Step 3. In the process of Apriori algorithm, all transactions are first read from database to obtain the support of candidate 1-itemset. Then, find out the

support of large 1-itemset and use the combination of these large 1-itemset to generate candidate 2-itemset.

Step 4. Rescan the database to obtain the support of candidate 2-temset, then find out the assembly of large 2-itemset and use the combination of these Large 2-itemset to generate candidate 3-itemset.

Step 5. Repeat scanning of database, and compare with minimum support to generate the large itemset. Then combining to generate the next step of candidate itemset until no new candidate itemset can be generated.

4.5 Establishing Visual Figures of Data Mining

The operational process and knowledge rules excavated by data mining can be displayed through visual figures. This allows users to operate data mining more easily and to understand the meaning represented by excavated data. A graphical user interface for a data mining engine is composed of the functional elements [10]:

1. Data collection and data mining query composition.
2. Presentation of discovered patterns.
3. Manipulation of data mining primitives.
4. Interactive multilevel mining.
5. On-line help manuals, indexed search, debugging, and other interactive graphical facilities.

Chapter 5
System Implementation

With the understanding the related theory of classification algorithms in the previous chapter, we begin the system implementation. The whole system framework involved in data preprocessing, transformation in to data warehouse, and practical operations of all algorithms. The implementation details are listed as follows.

5.1 System Implementation Environment

The system's environment and framework shown in Figure 23 include Microsoft SQL Server 2005 Data Warehouse Server and Data Mining tools, Web Server, PC, and front-end PC.

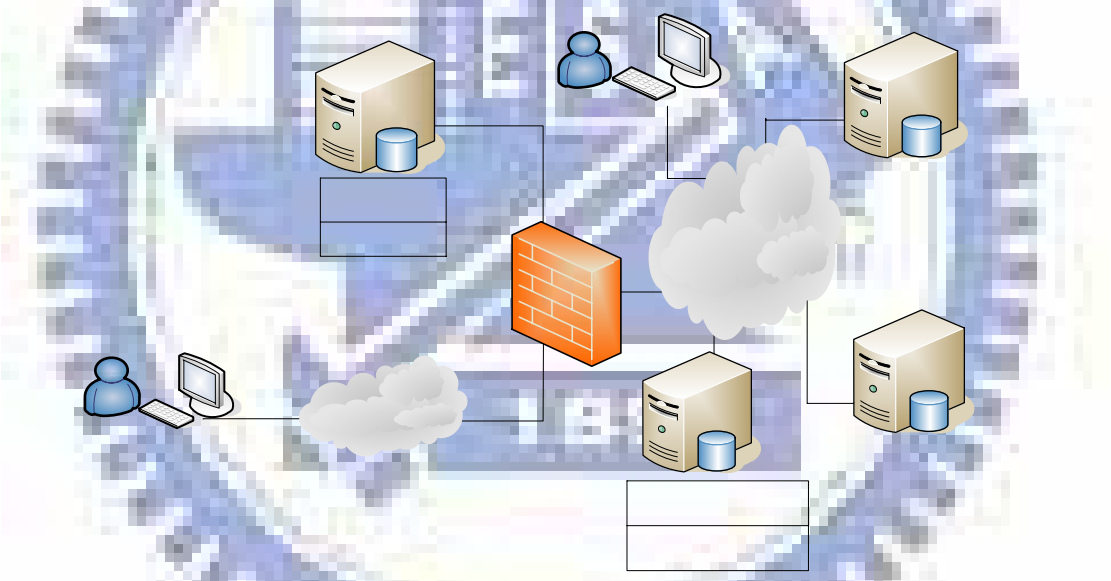


Figure 23. The environment and framework of product quality improvement system.

5.2 Data Collection

In the semiconductor packaging process, defective types often appear. There are seven categories, such as die saw, die attach, wire bonding, sealing, trimming/forming, marking, and plating. The detail items [18] are as shown in Table 5.

Table 5 Defective types in the manufacturing process

	Type of defect		Type of defect
Die saw	Scratches	Trimming/ forming	Missing or broken lead
	Damage		Bent lead
	Foreign material		Lead coplanarity
	Oxidation		Lead protruding, depression
	Void		Foreign material
	Dirty wafer		Scratches
Die attach	Die missing		Bur
	Cracked die	Marking	No marking
	Broken die		Poor alignment
	Wafer wrong orientation		Overlap marking
	Excessive epoxy		Dirty marking
	Insufficient epoxy		Reverse marking
Wire bonding	Wire missing		Positioning error
	Poor arc		Slanting words
	Poor wire pitch		Missing words
	Displaced wire		Serial number
	Excessive wire, long tail		Incorrect serial number
	Depressed wire		Word missing sides
	Damaged wire		Local misalignment
	Short circuited		Broken
Sealing	Void/insufficient filling		Scratches
	Shifting	Plating	Plating too thick or too thin
	Scratches		Foreign material
	Foreign material		Lead frame discoloration
	Crack/Broken		Insufficient or excessive solder
	Missing epoxy		Base material exposed
	Excessive epoxy		
	Mold alignment		
	Insufficient body epoxy		

In this stage, we collect the data of WIP and ERP from January to December in 2004 and diagnose and analyze quality data. After understanding the problem

definition and purpose, exploring the industry's characteristics, we proceed to collect data and select data items from the desired analysis scope for follow-up analysis. Data attributes are defined as man, machine, material and method, also known as 4Ms [23] [31], which are described as follows:

1. Man factors: Refer to associated man operation factors, such as failure to follow procedures and methods, failure to inspect, and failure to set parameters, etc.
2. Machine factors: Refer to associated machine operation problems, such as excessive pressure, extremely high tool head position, and no sensor reaction, etc.
3. Material factors: Refer to associated material problems, such as excessively strong glue, excessive thin inner lead, etc.
4. Method factors: Such as poor programming, indefinite instructions, and failure to use dust-free cloth, etc.

After completing attributes and their corresponding values, it requires preparing the data on the corrective action form. During the analysis process, we should summarize the data and convert them into a consistent format for data mining. And then we assign numbers (e.g. 0, 1, 2…) to attribute variables. "0" represents attribute variables with "no error"; rank them in a proper order. Please refer to Table 6, the contrast table of numbers of attribute variables and brief descriptions. In accordance with the four attributes of data mining, the numbers of individual variables are: man (9 variables), machine (13 variables), material (11 variables), and method (10 variables). Furthermore, we can define the major causes of each cluster of products defects shown in Figure 24.

Table 6 Four attributes contrast table

No	Man	Machine	Material	Method
0.	N/A	N/A	N/A	N/A
1.	ILB Maladjustment of parameters: Need to adjust physical control parameters such as temperature, pressure, etc during ILB.	CURING Temp. Error: Need to set up physical control parameters such as timing, temperature, etc. during curing.	Stronger glue: due to too slow flowing rate possibly will result in void, resin overflow, etc.	ILB No Positioning: may cause inner lead fracture, lift, etc.
2.	ILB station personal injury caused by impact: ILB machine operator bumped into machine, affecting the operation of machine.	ILB Pressure Error: abnormal ILB Pressure parameter may cause lead and bump wetting damage.	Glue material Void: Bubbles in the glue material, possibly result in bad roughness.	MARKING No positioning: may cause overlap marking, IC shrinkage, etc.
3.	MARKING Maladjustment of Parameters: Physical control parameter such as marking pressure, ink flow rate, etc.	POTTING Pressure Error: possible result in resin overflow, resin adhesion to tape, etc.	Excessive thin inner lead: ILB leads too thin on the tape, possibly result in lead bending, fracture, delamination, etc.	POTTING No positioning: may cause resin overflow, resin adhesion to tape, etc.
4.	PACKING station personal injury caused by impact: Operator on packing station bumped into machine, affecting the operation of machine.	ILB Sensor Error: resulting in unable to complete bonding action, and may cause lead fracture.	Tool head scratch: may cause IC shrinkage.	RW No Positioning: may cause IC shrinkage, lead fracture, etc.
5.	POTTING Maladjustment of Parameters: Need to adjust physical control parameters such as temp., pressure, and amount of glue, etc. during gluing.	MARKING Sensor Error: resulting in overlap marking.	Buckled inner lead of Tape: Inner leads on tape are slanting, may cause lead fracture.	Excessive Glue: may result in resin overflow, resin adhesion to tape, etc.
6.	POTTING station personal injury caused by impact: Gluing machine operator bumped into machine, affecting the operation of machine.	POTTING Sensor Error: Sensor recognition error on potting station, resulting in resin overflow.	Tape recession: Poor tape quality, resulting in lead damage.	Glue misuse: Error in selection of glue, may result in bad roughness, void, etc.
7.	RW Maladjustment of parameters: Need to adjust physical control parameters such as conveyor speed, alignment of sprocket holes, etc.	MARKING Pressure Error: Abnormal pressure parameter on MARKING machine may cause overlap marking.	Tape expansion: Poor tape quality, result in lead fracture.	Improper glue clearing: may result in void, filler settling, etc.
8.	RW station personal injury caused by impact: Operator on RW station bumped into machine, affecting the operation of manufacturing processes.	ILB Material jam on rail: Material dropped on the track, resulting in lead damage, fracture, etc.	Insufficient solder for tape: possibly result in bad wetting between lead and bumping.	Failure to use dust-free cloth: Not wiped with dust-free cloth, possibly lead to surface roughness.
9.		MARKING Material jam on rail: Material dropped on the track, resulting in overlap marking.	Excessive IC hardness: may cause IC shrinkage.	Temperature Condition Error: Error in lead bonding temperature, possibly resulting in void, lead delamination, etc.
10.		POTTING Material jam on rail: Material dropped on track, resulting in unable to complete potting action.	Error Program: Program for testing product possibly having error, resulting in unable to pass electronic testing.	
11.		ILB Tool head error: may cause lead fracture, lift, etc.		
12.		ILB Pattern Identification Error: Sensor recognition error on POTTING station, resulting in unable to complete POTTING action.		

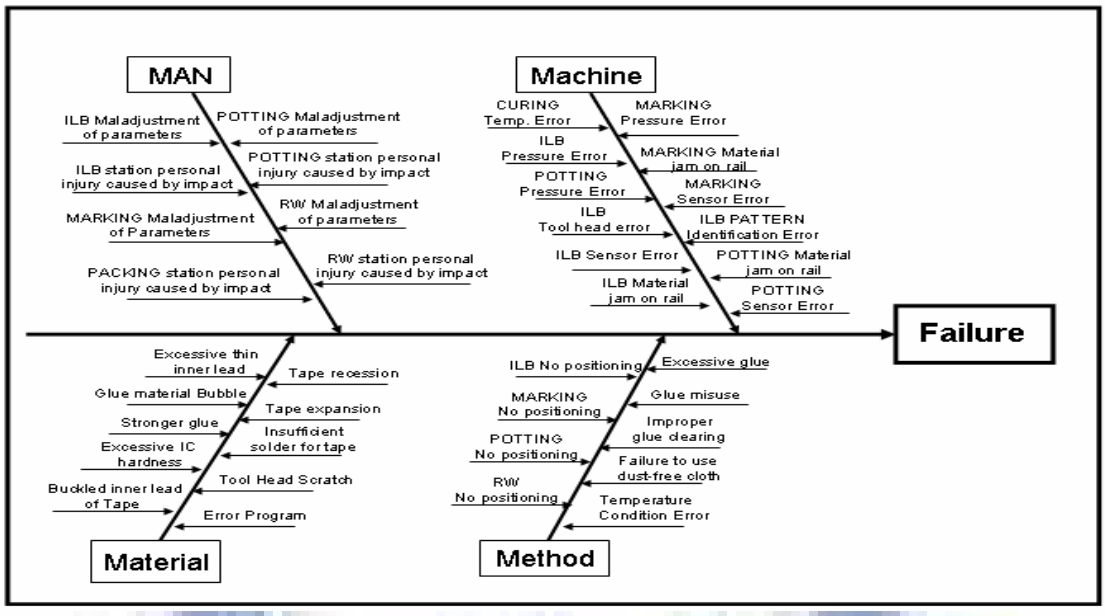


Figure 24. Major causes of product defects.

5.3 Data Cleaning

The four major parts in this stage are as follows.

1. Disposition of data loss: Such as null value, inexistent value, and incomplete data adjustment and settlement.
2. Disposition of deviated value inspection: Delete or retain the scattered data beyond normal distribution.
3. Clarification of fuzzy definition: In case of any value in different fields represents the same meaning, it is necessary to make the data consistent to clarify the definition.
4. Disposition of error values: If a field value does not conform to the effective value, there may be a problem with input errors or program errors. Determine the disposition method according its effectiveness.

5.4 New Data Generation

After a series of data filtering and cleaning up, the step follows the generation of new data. The original data of the whole quality system is extracted from WIP and ERP systems. The ETL is also known as the combination of extraction, transformation, and loading. The processing procedures include extracting data from the system, loading data to the staging area, purifying and transforming the data, and

transferring it to the data warehouse. We use SSIS (SQL Server Integration Service of Microsoft SQL Server 2005) to transform WIP and ERP database into data warehouse, shown in Figure 25 [26].

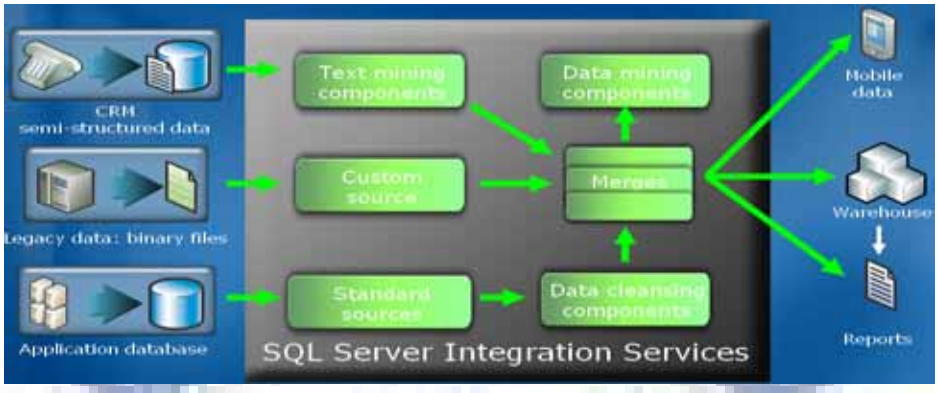


Figure 25. Microsoft SQL Server 2005 SSIS diagram.

5.5 Establishing a Data Warehouse

The data warehouse system is a starflake framework centered the manufacturing fact table with the association time, lot, product, and quality dimension table. In the system, the quality problem dimension table includes multiple dimensions and measurements; a dimension represents a specific view diagram of observation data, and a measurement represents an actual data indicator, shown in Figure 26. The schema of each table attribute is shown in Tables 7, 8, 9, 10, 11 and 12.

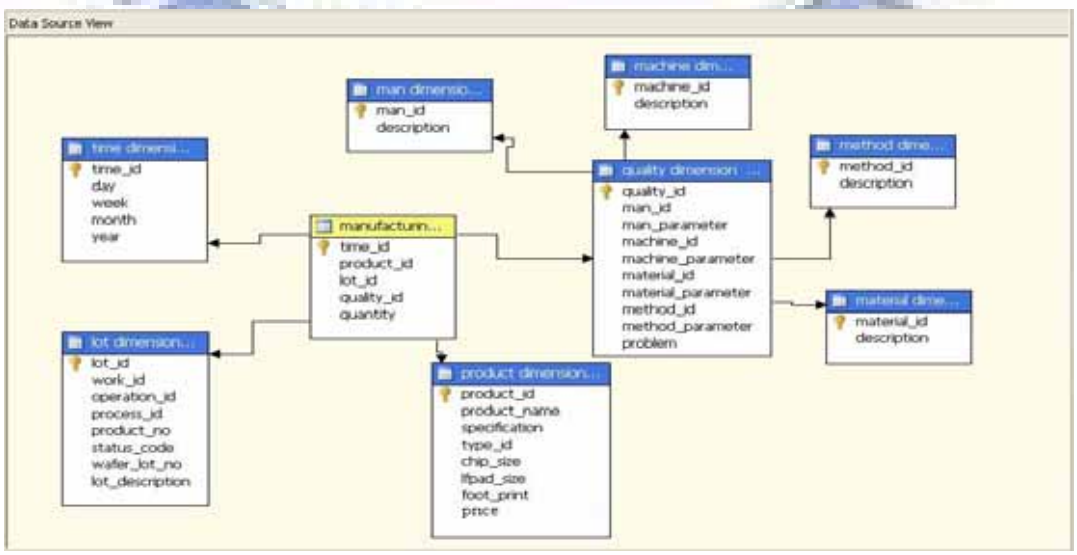


Figure 26. The quality improvement system starflake schema.

Table 7 Manufacturing fact table

Column Name	*Type*	*Length*	*Description*
time_id	int	14	time identification
product_id	char	16	product identification
lot_id	char	16	lot identification
quality_id	char	16	quality identification
quantity	int	16	quantity of production
Primary key: time_id	Foreign key: product_id, lot_id and quality_id		

Table 8 Quality dimension table

Column Name	*Type*	*Length*	*Description*
quality_id	char	16	quality identification
man_id	char	2	man identification
machine_id	char	2	machine identification
material_id	char	2	material identification
method_id	char	2	method identification
man_parameter	char	20	man parameter description
machine_parameter	char	20	machine parameter description
material_parameter	char	20	material parameter description
method_parameter	char	20	method parameter description
quantity	int	16	quantity of production
Primary key: quality_id			
Foreign key: man_id machine_id, material_id, and method_id			

Table 9 Man dimension table

Column Name	*Type*	*Length*	*Description*
man_id	char	2	man identification
description	char	20	man description
Primary key: man_id			

Table 10 Machine dimension table

Column Name	*Type*	*Length*	*Description*
machine_id	char	2	machine identification
description	char	20	machine description
Primary key: machine_id			

Table 11 Material dimension table

Column Name	*Type*	*Length*	*Description*
material_id	char	2	material identification
description	char	20	material description
Primary key: material_id			

Table 12 Method dimension table

Column Name	*Type*	*Length*	*Description*
method_id	char	2	method identification
description	char	20	method description
Primary key: method_id			

5.6 Clustering and Defining Meanings

Although K-means are well-known and so-called “rigid cluster”, there are quite a few limitations in its application. However, to solve these limitations, the expectation maximum algorithm came to existence, usually called E-M algorithm, and is an example of soft clustering. After randomly selecting some portions of data form data warehouse and discussing with experts, we can obtain the E-M algorithm is better than K-means algorithm. With the understanding the two algorithms, we can start to set the parameters. The parameters setting by the research needed to specify and some parameters of the tool we used are default values are listed as Table 13 [32]. Therefore, we use the E-M algorithm to cluster the quality data collected to obtain 9 clusters of defective types, such as tape resin adhesion, shrinkage/sprocket hole damage, resin overflow, inner lead broken/inner lead bending/inner lead delamination/outer lead damage, Short/Open, overlap marking, resin overflow, resin poor wrapping/riding, resin wrapping drawn-in object/tape indent, shown in Figure 27. The number of items and percentages are shown in Table 14. We can define the major causes of each cluster of products defects mentioned in the previous section shown in Figure 24.

Table 13 Parameters of clustering algorithm setting by this research

Parameter	Description	Parameter Setting by this research
CLUSTERING_ METHOD	The following clustering methods are available: scalable EM (1), non-scalable EM (2), scalable K-Means (3), and non-scalable K-Means (4). The default is (1).	Scalable EM (1): scalable model is based on sampling to find out the structure of clustering, and use the whole data to calculate the detail structure to save time. Besides, EM (Expectation Maximization) belongs to soft cluster without the problem of clustering boundaries.
CLUSTER_COUNT	Setting the CLUSTER_COUNT to 0 causes the algorithm to use heuristics to best determine the number of clusters to build. The default is 10.	Set to 0 causes the algorithm to use heuristics to best determine the number of clusters to build.
CLUSTER_SEED	Specifies the seed number that is used to randomly generate clusters for the initial stage of model building. The default is 0.	After we run the proper range of parameters in the study, the better value is 0.
MINIMUM_ SUPPORT	Specifies the minimum number of cases in each cluster. The default is 1.	After we run the proper range of parameters in the study, the better value is 1.
MODELLING_ CARDINALITY	Specifies the number of sample models that are constructed during the clustering process. The default is 10.	After we run the proper range of parameters in the study, the better value is 10.
STOPPING_ TOLERANCE	Specifies the value that is used to determine when convergence is reached and the algorithm is finished building the model. Convergence is reached when the overall change in cluster probabilities is less than the ratio of the STOPPING_TOLERANCE parameter divided by the size of the model. The default is 10.	After we run the proper range of parameters in the study, the better value is 10.
SAMPLE_SIZE	Specifies the number of cases that the algorithm uses on each pass if the CLUSTERING_METHOD parameter is set to one of the scalable clustering methods. Setting the SAMPLE_SIZE parameter to 0 will cause the whole dataset to be clustered in a single pass. This can cause memory and performance issues. The default is 50000.	There are 25270 cases in the study. The training data contain 70% within the whole data. The value is at least 18000.
MAXIMUM_INPUT_ ATTRIBUTES	Specifies the maximum number of input attributes that the algorithm can handle before it invokes feature selection. The default is 255.	There are 4 input attributes in the study. The value is at least 4.
MAXIMUM_STATES	Specifies the maximum number of attribute states that the algorithm supports. The default is 100.	The maximum number of discrete states of machine in the study is 13. The value is at least 13.

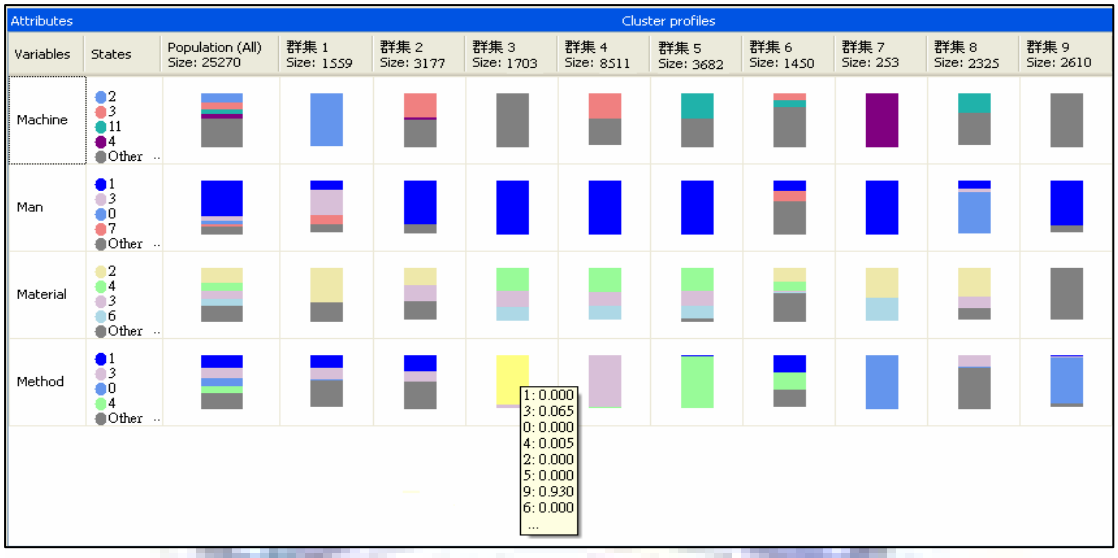


Figure 27. Illustration of clustering classification.

Table 14 Number of all defective clusters and their percentage

Cluster No.	Judgment (Problem)	No. of Defects	Percentage of total defects (%)
1	Tape Resin Adhesion	1559	6.17
2	Shrinkage/ Sprocket hole damage	3177	12.57
3	Resin overflow	1703	6.74
4	Inner lead Broken/ Inner lead Bending/ Inner lead delamination/ Outer Lead Damage	8511	33.68
5	Short/Open	3682	14.57
6	Overlap Marking	1450	5.74
7	Void	253	1.00
8	Resin Poor Wrapping/ Riding	2325	9.20
9	Resin wrapping Drawn-in object/ Tape indent	2610	10.33
	Total	**25270**	**100.00**

The 9 clusters of defective types and the major causes are shown as follows.

Cluster 1: Tape resin adhesion: resin overflows adhesion to tape.

Cluster 2: Shrinkage/sprocket hole damage: IC fracture because of temp. or pressure of ILB or potting error/maybe tape recession cause the defect.

Cluster 3: Resin overflow: stronger glue; temp. or pressure error of potting station.

Cluster 4: Inner lead broken/inner lead bending/inner lead delamination/outer lead damage: excessive thin inner lead/temp. and pressure of ILB or potting/insufficient solder for tape/temp. and pressure of ILB or potting may cause the defect.

Cluster 5: Short/Open: lead overlap/lead fracture may cause the defect.

Cluster 6: Overlap marking: improper marking parameter setting may cause the defect.

Cluster 7: Resin overflow: improper selection of glue material may cause the defect.

Cluster 8: Resin poor wrapping/riding: improper potting parameters setting/improper potting parameter setting may cause the defect.

Cluster 9: Resin wrapping drawn-in object/Tape indent: improper potting parameters setting/tape recession or expansion may cause the defect.

After completing attributes, we define the judgment field formats. Similarly, quality problem types are numbered 1, 2, 3…. There are 9 types of problems in total (or 10 types, if added the type of "Normal"). The contrast table of numbers of attributes variables and quality problems are shown in Table 15 and 16 respectively.

Table 15 Contrast table of numbers of attribute variables and quality problems

No.	Man	Machine	Material	Method	Judgment(Problem)
0	N/A	N/A	N/A	N/A	Normal
1	ILB Maladjustment of parameters	CURING Temp. Error	Stronger Glue	ILB No positioning	**Tape Resin Adhesion:** Resin overflows adhesion to tape.
2	ILB station personal injury caused by impact	ILB Pressure Error	Glue matrerial Bubble	MARKING No positioning	**Shrinkage:** IC fracture because of temp. or pressure of ILB or potting error / **Sprocket hole damage:** maybe tape recession cause the defect.
3	MARKING Maladjustment of Parameters	POTTING Pressure Error	Excessively thin inner lead	POTTING No positioning	**Resin overflow:** stronger glue; temp. or pressure error of potting station.
4	PACKING station personal injury caused by impact	ILB Sensor Error	Tool Head Scratch	RW No positioning	**Inner lead Broken:** excessive thin inner lead may cause the defect. / **Inner lead Bending:** temperature and pressure of ILB or potting may cause the defect. / **Inner lead Delamination:** insufficient solder for tape may cause the defect. / **Outer Lead Damage:** temperature and pressure of ILB or potting may cause the defect.
5	POTTING Maladjustment of parameters	MARKING Sensor Error	Buckled inner lead of Tape	Excessive Glue	**Short:** lead overlap may cause the defect. / **Open:** lead fracture may cause the defect.
6	POTTING station personal injury caused by impact	POTTING Sensor Error	Tape recession	Glue misuse	**Overlap Marking:** improper marking parameter setting may cause the defect.
7	RW Maladjustment of parameters	MARKING Pressure Error	Tape expansion	Improper glue clearing	**Void:** improper selection of glue material may cause the defect.
8	RW station personal injury caused by impact	ILB Material jam on rail	Insufficient solder for tape	Failure to use dust-free cloth	**Resin Poor Wrapping:** improper potting parameters setting may cause the defect. / **Riding:** improper potting parameter setting defect.
9		MARKING Material jam on rail	Excessive IC hardness	Temperature Condition Error	**Resin wrapping Drawn-in object**: improper potting parameters setting may cause the defect. / **Tape indent:** tape recession or expansion, etc.
10		POTTING Material jam on rail	Error Program		
11		ILB Tool head error			
12		ILB PATTERN Identification Error			

Table 16 Independent mapping table for all attribute variables and judgment items

No.	Man	Machine	Material	Method	Problem
C010334-01	POTTING Maladjustment of Parameters (5)	ILB Pressure Error (2)	Excessively thin inner lead (3)	N/A (0)	Inner lead Broken/ Inner lead Bending/ Inner lead Delamination/ Outer Lead Damage (4)
C030046-02	Marking Maladjustment of parameters (3)	N/A (0)	Stronger glue (1)	Temp. Condition error (9)	Resin overflow (3)
C030122-05	N/A (0)	N/A (0)	Glue material bubble (2)	Temp. Condition error (9)	Resin overflow (3)
……	……	……	……	……	……
C310114-07	MARKING Maladjustment of parameters (3)	ILB Pressure Error (2)	N/A (0)	ILB No positioning (1)	Shrinkage/ Sprocket hole damage (2)
C310196-04	POTTING Maladjustment of parameters (5)	POTTING Pressure Error (3)	N/A (0)	Excessive glue (5)	Resin overflow (3)
C410201-09	MARKING Maladjustment of parameters (3)	MARKING Sensor error (5)	N/A (0)	N/A (0)	Overlap Marking (6)

5.7 Implementing Proposed Algorithms

Quality problem data (25,270 entries) are classified, predicted, and analyzed by the decision tree, neural network, logistic regression, Bayesian, and association rule algorithms. The data is randomized to partition into training set and testing set by 3:1 (training set: 18,952 entries, testing set: 6,318 entries). The classification matrix, lift chart, processing speed, robustness, and visualized display of proposed algorithms are represented as follows.

5.7.1 Decision tree

1. The parameters setting by the research needed to specify are listed as Table 17 [33].

Table 17 Parameters of decision tree algorithm setting by this research

Parameter	Description	Parameter Setting by this research
SCORE_METHOD	Determines the method that is used to calculate the split score. Available options: Entropy (1), Bayesian with K2 Prior (3), or Bayesian Dirichlet Equivalent (BDE) Prior (4). The default is (4).	Entropy (1): after practical operation and comparing the three methods. We obtain the entropy (1) is better.
SPLIT_METHOD	Determines the method that is used to split the node. Available options: Binary (1), Complete (2), or Both(3). The default is (3).	Both (3): Binary(1) generates the binary branches without the complete structure of decision tree, and Complete (2) generates the structure of decision tree is complex. The methods mentioned above are not proper in the study. Therefore, the selection of Both (3) depends on the algorithm to generate the structure of decision tree.
MINIMUM_SUPPORT	Determines the minimum number of leaf cases that is required to generate a split in the decision tree. The default is 10.	After we validate all range of the parameter, the expenditure of time is long with the number increase. We chose 1 as the value.
MAXIMUM_INPUT_ATTRIBUTES	Defines the number of input attributes that the algorithm can handle before it invokes feature selection. Set this value to 0 to turn off feature selection. The default is 255.	After we run the proper range of parameters in the study, the value is at least 4.
MAXIMUM_OUTPUT_ATTRIBUTES	Defines the number of output attributes that the algorithm can handle before it invokes feature selection. Set this value to 0 to turn off feature selection. The default is 255.	After we run the proper range of parameters in the study, the value is at least 9.
COMPLEXITY_PENALTY	Controls the growth of the decision tree. A low value increases the number of splits, and a high value decreases the number of splits. The default value is based on the number of attributes for a particular model, as described in the following list: • For 1 through 9 attributes, the default is 0.5. • For 10 through 99 attributes, the default is 0.9. • For 100 or more attributes, the default is 0.99.	After we run the proper range of parameters in the study, the better value is 0.5.

2. The visualized result with decision tree is represented in Figure 28. For example, if machine =”3” and man = “1” and material = “7” then problem = “9”.

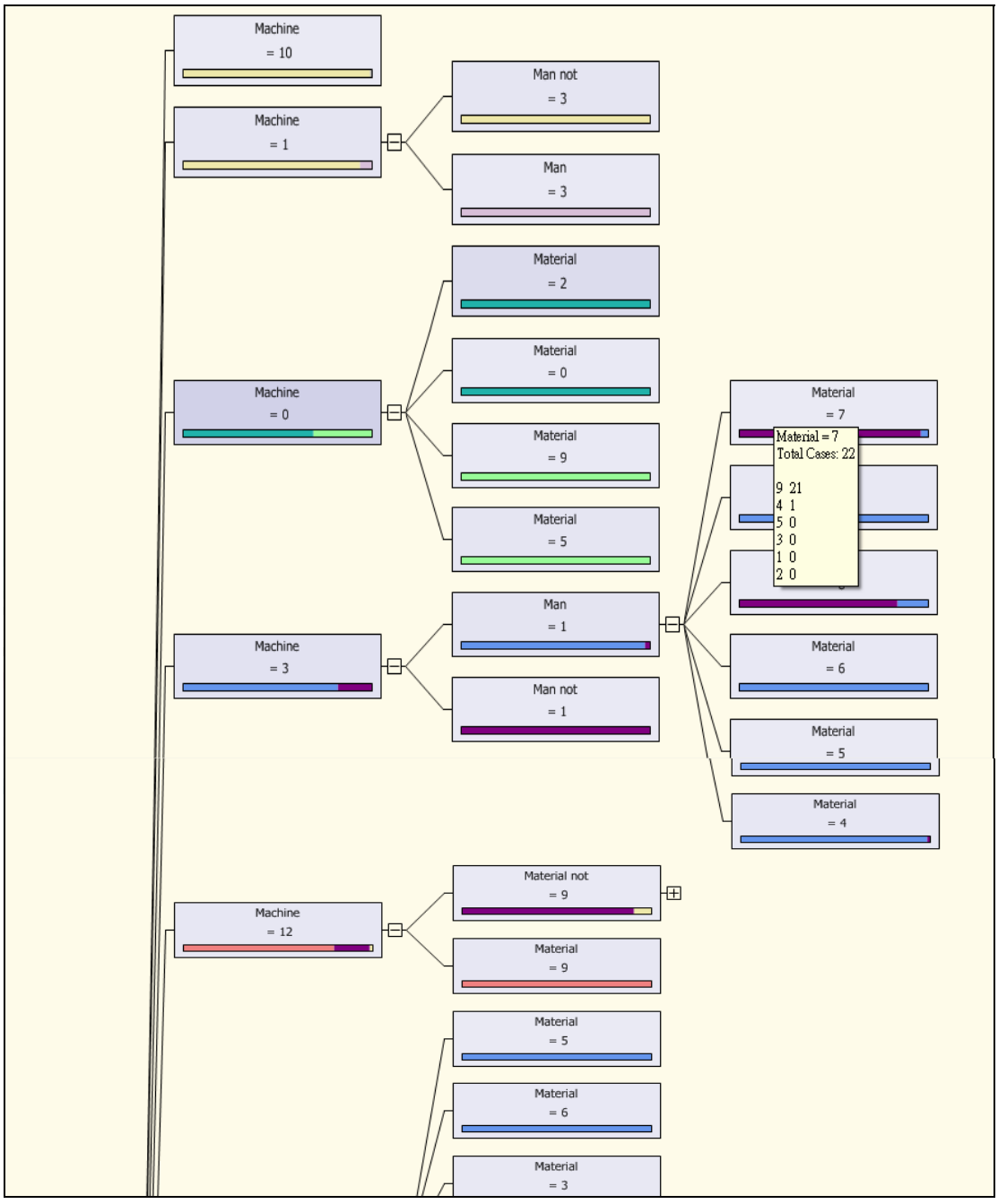


Figure 28. Display of decision tree model classification.

3. For the classification matrix of the decision tree, there are 5,680 entries of correct data, representing a success rate of 89.9%. The classification matrix diagram is shown in Figure 29, and Table 18.

		True Class									
		1	2	3	4	5	6	7	8	9	
Predicted Class	1	350	23		23	12	11	3	5		427
	2		714	14	15	12				21	776
	3	3		383	33				18		437
	4	12	6		1913	24	15	1	14	11	1996
	5	15	19	11	18	828					891
	6		3		41		326			25	395
	7	8	21	4	32	11		57		9	142
	8		3	3	22	13	11	2	523		577
	9	2	5	11	31	21			21	586	677
		390	794	426	2128	921	363	63	581	652	6318

Figure 29. Matrix classification of decision tree.

Table 18 Total and accurate entries of decision tree

Entries \ Problems	1	2	3	4	5	6	7	8	9	Total	accuracy rate
Accurate entries	350	714	383	1913	828	326	57	523	586	5680	89.9%
Total entries	390	794	426	2128	921	363	63	581	652	6318	

4. The x-axis and y-axis of lift chart are composed of percentage values. The x-axis percentage is representative of the percentage value calculated by dividing the number of odds of the data model, which are ranked from high to low, by the total number of data values. The y-axis is representative of the percentage calculated by dividing the number of incidents for this specific list by the total number of incidents. The 45-degree line in the figure represents optimal model. If the lift accuracy illustrated in the model is higher, it is indicative of a more validated and more effective model. The lift chart of decision tree is 91.0%, shown in Figure 30.

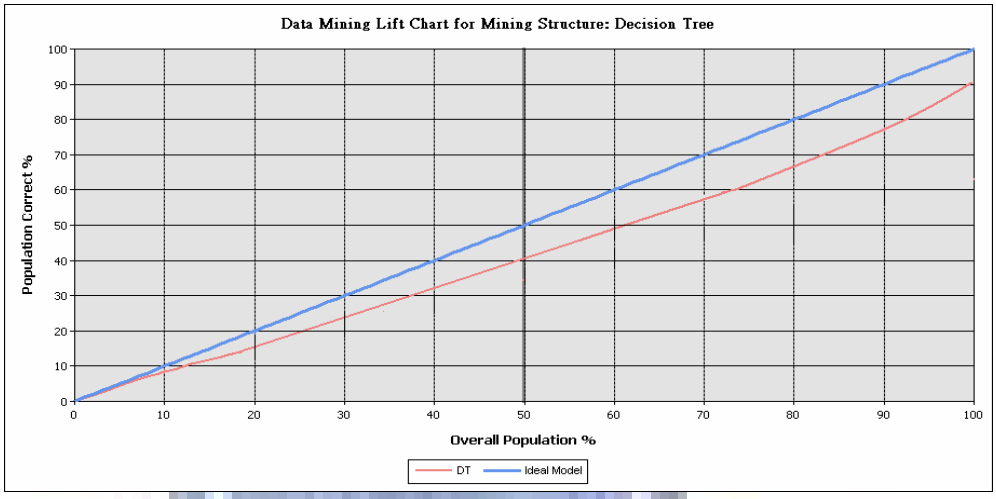


Figure 30. Lift chart of decision tree.

5. The processing speed of decision tree is illustrated in Figure 31. The duration of process is 23 seconds.

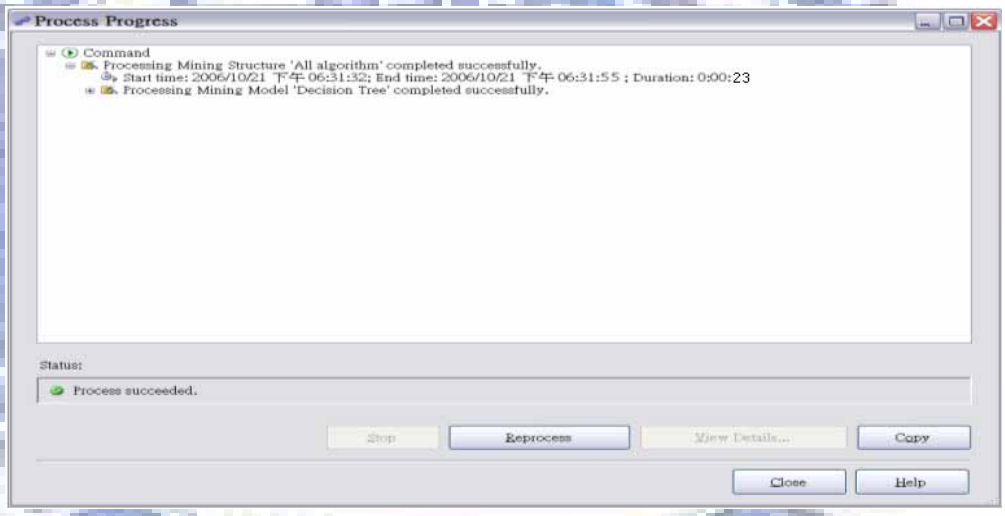


Figure 31. Processing time of decision tree.

6. The robustness of decision tree is shown in Figure 32. The missing value percentage is 0.00%.

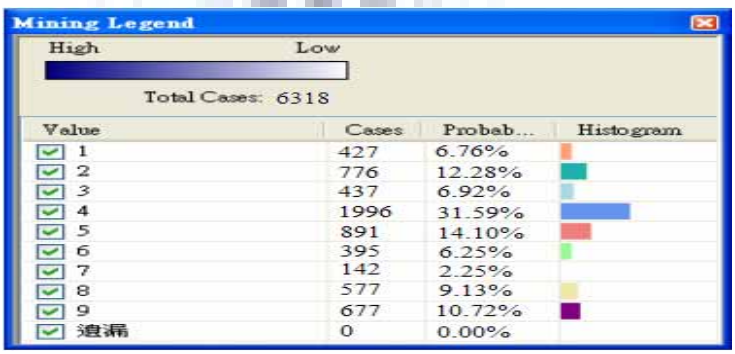


Figure 32. Missing value automatically filled in missing with decision tree.

5.7.2 Neural network

1. All parameters of the tool we used are default values, shown as Table 19.

Table 19 Parameters of neural network algorithm by this research

Parameter	Description	Parameter Setting by this research
HIDDEN_NODE_RATIO	Specifies the ratio of hidden neurons to input and output neurons. The following formula determines the initial number of neurons in the hidden layer: HIDDEN_NODE_RATIO * SQRT(Total input neurons * Total output neurons) The default value is 4.0.	There are 4 input attributes and 9 output attributes in the study. The hidden layer is only one and initial neueons are calculated in the fumula: HIDDEN_NODE_RATIO * SQRT(Total 4 * 9) equal to 24.
HOLDOUT_ PERCENTAGE	Specifies the percentage of cases within the training data used to calculate the holdout error, which is used as part of the stopping criteria while training the mining model. The default value is 30.	The validation set select 30% within whole data. The value set to 30.
HOLDOUT_SEED	Specifies a number that is used to seed the pseudo-random generator when the algorithm randomly determines the holdout data. If this parameter is set to 0, the algorithm generates the seed based on the name of the mining model, to guarantee that the model content remains the same during reprocessing. The default value is 0.	To guarantee that the model content remains the same during reprocessing. The value is 0.
MAXIMUM_INPUT_ ATTRIBUTES	Determines the maximum number of that can be supplied to the algorithm before feature selection is employed. Setting this value to 0 disables feature selection for input attributes. The default value is 255.	There are 4 input attributes in the study. The value is at least 4.
MAXIMUM_OUTPUT_ ATTRIBUTES	Determines the maximum number of output attributes that can be supplied to the algorithm before feature selection is employed. Setting this value to 0 disables feature selection for output attributes. The default value is 255.	There are 9 output attributes in the study. The value is at least 9.
MAXIMUM_STATES	Specifies the that is supported by the algorithm. If the number of states for a specific attribute is greater than the number that is specified for this parameter, the algorithm uses the most popular states for that attribute and treats the remaining states as missing. The default value is 100.	The maximum number of discrete states of machine in the study is 13. The value is at least 13.
SAMPLE_SIZE	HOLDOUT_PERCENTAGE is set to 30, the algorithm will use either the value of this parameter, or a value equal to 70 percent of the total number of cases, whichever is smaller. The default value is 10000.	There are 25270 cases in the study. The training data contain 70% within the whole data. The value is at least 18000.

The architecture of neural network by the research, shown in Figure 33, is composed of three layers, such as an input layer, a hidden layer and an output layer. There are 4 inputs in the input layer, 24 neurons in a hidden layer and 9 outputs in the output layer.

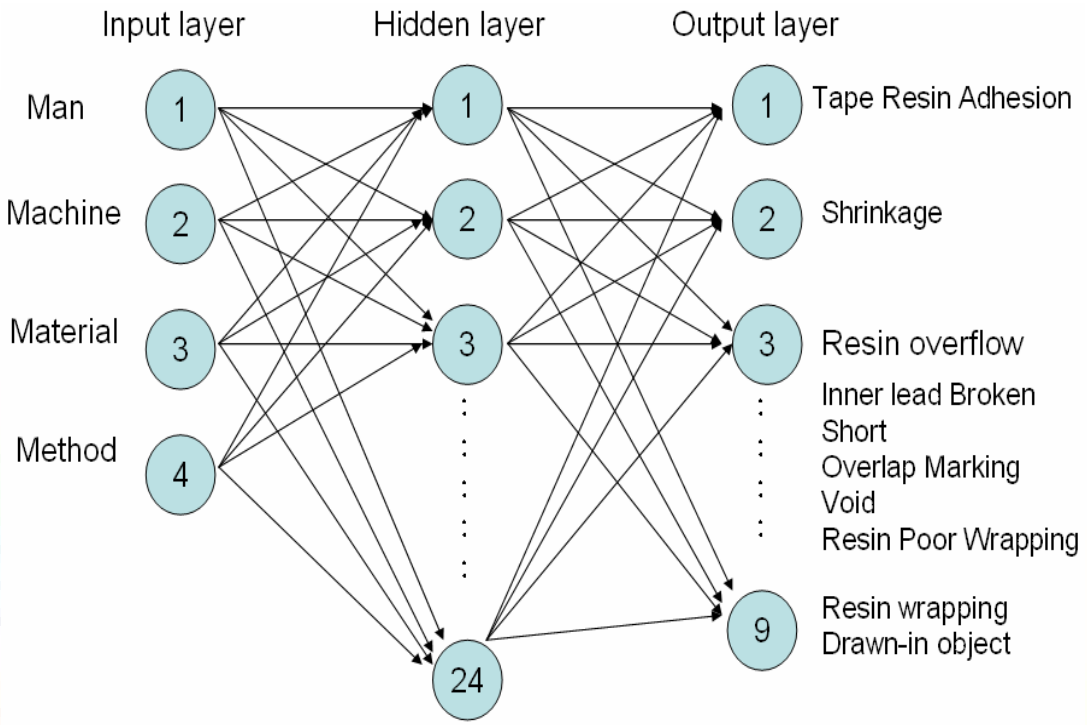


Figure 33. The architecture of neural network by the research.

2. The visualized result with neural network is represented in Figure 34. For example, if material = "3", then the probability of problem "5" is larger than that of problem "2".

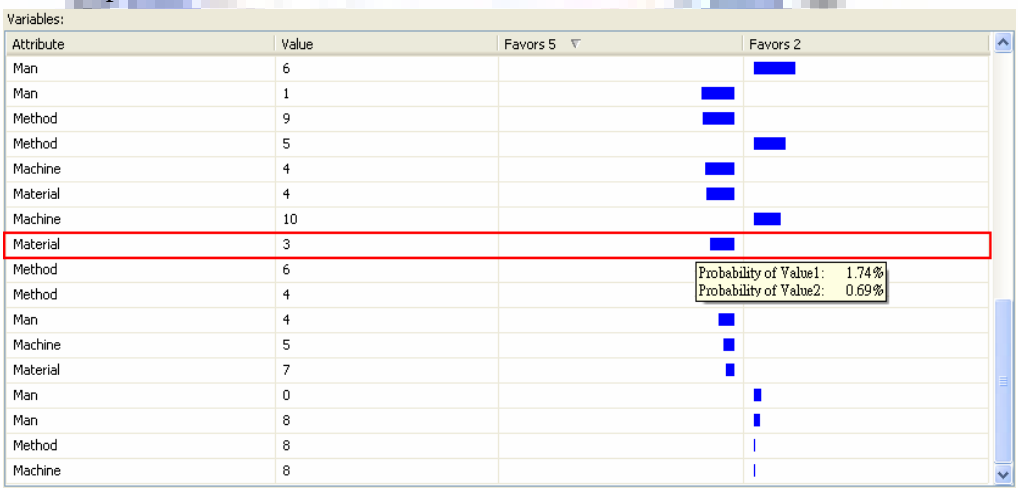


Figure 34. Display of neural network model classification.

3. For classification and analysis with the neural network, there are 5,326 entries of correct data, representing a success rate of 84.3%. The matrix analysis diagram is shown in Figure 35, and Table 20.

		True Class									
		1	2	3	4	5	6	7	8	9	
Predicted Class	1	329	12		66	22		1	1	16	447
	2	19	669	20	6	19	17	1	11	45	807
	3		32	359	25	16		2		10	444
	4	13	22		1794		12	3		21	1865
	5			12	65	776			16	46	915
	6	21	15	19	47		306			31	439
	7		13		48	23		53	51	13	201
	8		31	5	41	35			490	21	623
	9	8		11	36	30	28	3	12	449	577
		390	794	426	2128	921	363	63	581	652	6318

Figure 35. Matrix classification of neural network.

Table 20 Total and accurate entries of neural network

Entries \ Problems	1	2	3	4	5	6	7	8	9	Total	accuracy rate
Accurate entries	329	669	359	1794	776	306	53	490	449	5326	84.3%
Total entries	390	794	426	2128	921	363	63	581	652	6318	

4. The lift chart of neural network is 85.3%, shown in Figure 36.

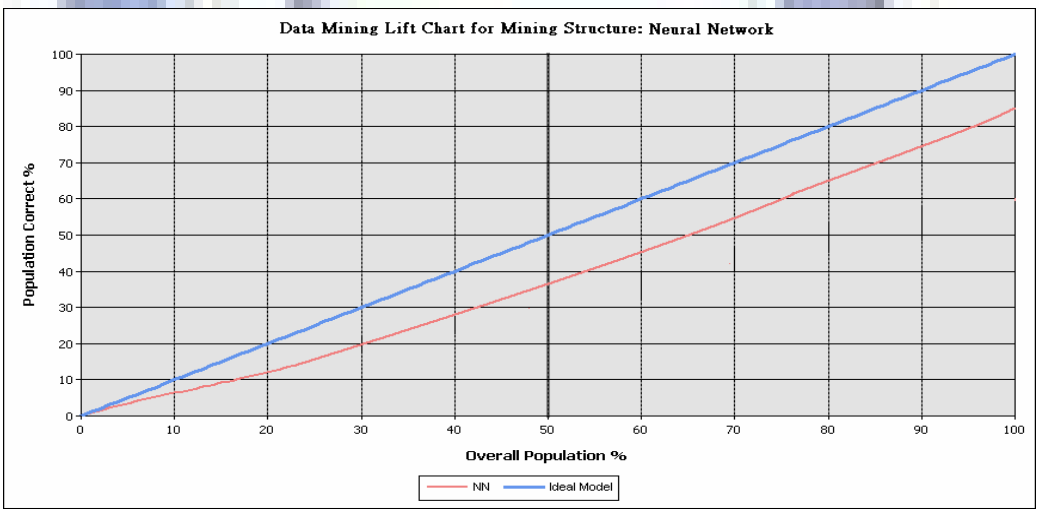


Figure 36. Lift chart of neural network.

5. The processing speed of neural network is illustrated in Figure 37. The duration of process is 46 seconds.

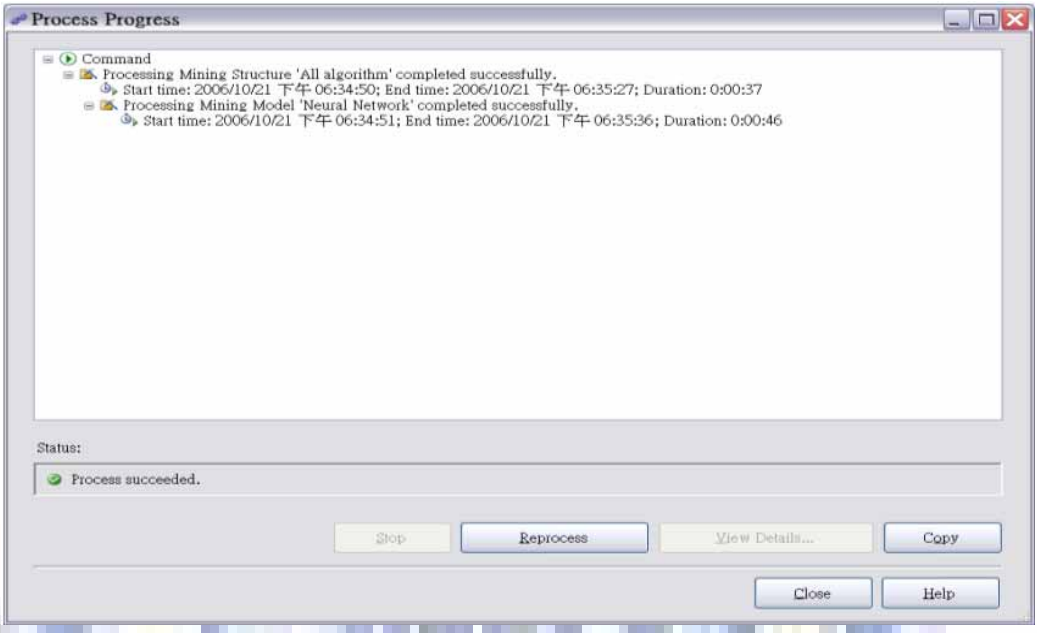


Figure 37. Processing time of neural network.

6. The robustness of neural network is shown in Figure 38. For example, if the attribute of method in 4Ms is "9", then the probability of missing value percentage is 0.01%.

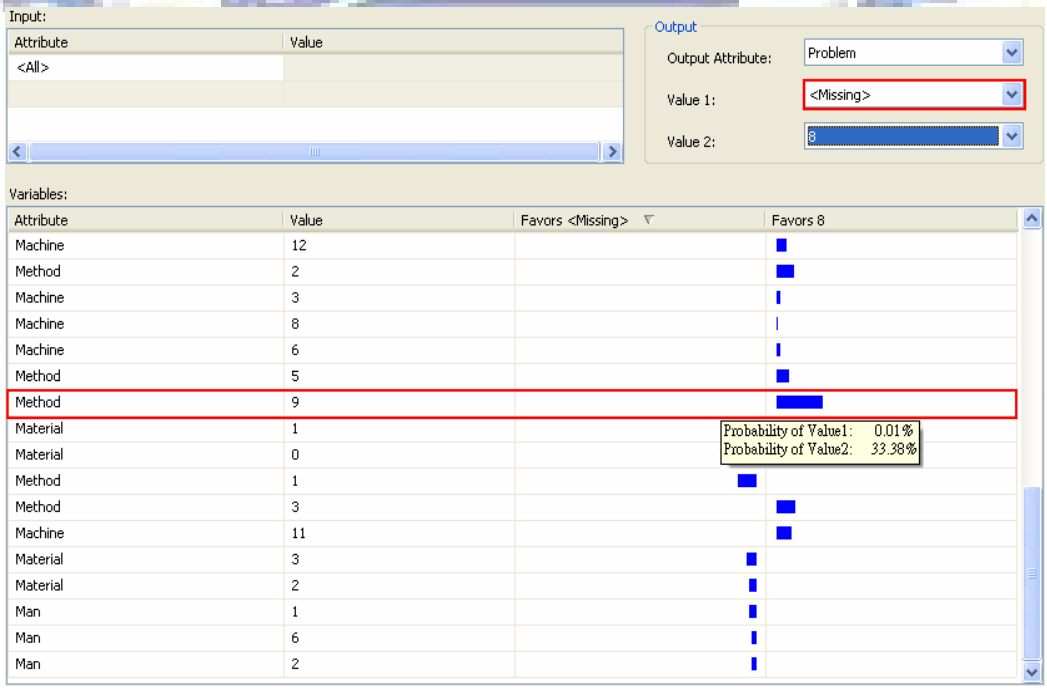


Figure 38. Missing value automatically filled in missing with neural network.

5.7.3 Logistic regression

1. All parameters of the tool we used are default values, shown as Table 21 [35].

Table 21 Parameters of logistic regression algorithm setting by this research

Parameter	Description	Parameter Setting by this research
HOLDOUT_PERCENTAGE	Specifies the percentage of cases within the training data used to calculate the holdout error, which is used as part of the stopping criteria while training the mining model. The default value is 30.	The validation set select 30% within whole data. The value set to 30.
HOLDOUT_SEED	Specifies a number that is used to seed the pseudo-random generator when the algorithm randomly determines the holdout data. If this parameter is set to 0, the algorithm generates the seed based on the name of the mining model, to guarantee that the model content remains the same during reprocessing. The default value is 0.	To guarantee that the model content remains the same during reprocessing. The value is 0.
MAXIMUM_INPUT_ ATTRIBUTES	Determines the maximum number of input attributes that can be supplied to the algorithm before feature selection is employed. Setting this value to 0 disables feature selection for input attributes. The default value is 255.	There are 4 input attributes in the study. The value is at least 4.
MAXIMUM_OUTPUT_ATTRIBUTES	Determines the maximum number of output attributes that can be supplied to the algorithm before feature selection is employed. Setting this value to 0 disables feature selection for output attributes. The default value is 255.	There are 9 output attributes in the study. The value is at least 9.
MAXIMUM_STATES	Specifies the maximum number of discrete states per attribute that is supported by the algorithm. If the number of states for a specific attribute is greater than the number that is specified for this parameter, the algorithm uses the most popular states for that attribute and treats the remaining states as missing. The default value is 100.	The maximum number of discrete states of machine in the study is 13. The value is at least 13.
SAMPLE_SIZE	HOLDOUT_PERCENTAGE is set to 30, the algorithm will use either the value of this parameter, or a value equal to 70 percent of the total number of cases, whichever is smaller. The default value is 10000.	There are 25270 cases in the study. The training data contain 70% within the whole data. The value is at least 18000.

2. The visualized result with logistic regression is represented as Figure 39. For example, if man = “1” then the probability of problem “5” is larger than that of problem “6”.

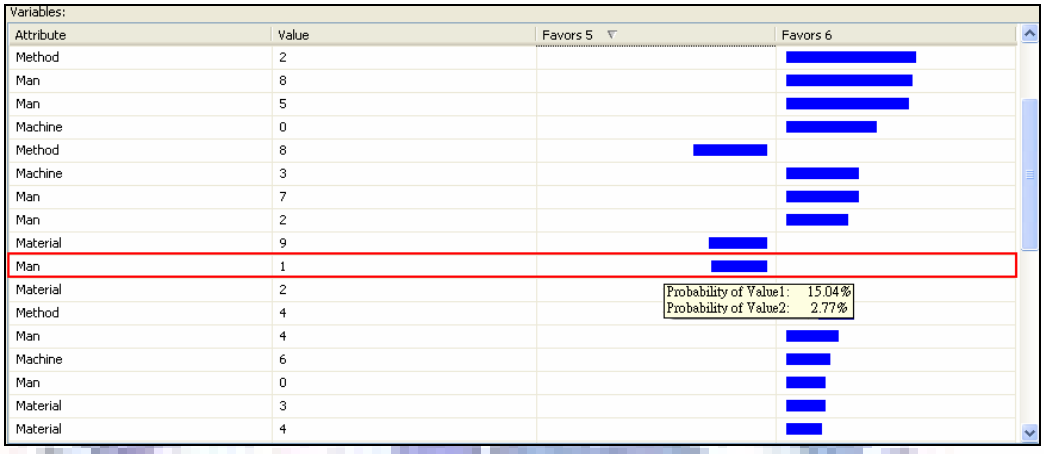


Figure 39. Display of logistic regression model classification.

3. For classification and analysis with the logistic regression, there are 5,294 entries of correct data, representing a success rate of 83.8%. The matrix analysis diagram is shown in Figure 40, and Table 22.

		Ture Class									
		1	2	3	4	5	6	7	8	9	
Predicted Class	1	327	21	12	66	23	15	3	16	23	506
	2		665	15	21	34	3		21	16	775
	3	13		357	45	16		1			432
	4		11		1783		16			2	1812
	5	15	31	13	42	771			10	3	885
	6	17	12	12	34		304	4	35	21	439
	7		31		42	21	21	53	11		179
	8	8		10	62	25		2	487	41	635
	9	10	23	7	33	31	4		1	546	655
		390	794	426	2128	921	363	63	581	652	6318

Figure 40. Matrix classification of logistic regression.

Table 22 Total and accurate entries of logistic regression

Entries \ Problems	1	2	3	4	5	6	7	8	9	Total	accuracy rate
Accurate entries	327	665	357	1783	771	304	53	487	547	5294	83.8%
Total entries	390	794	426	2128	921	363	63	581	652	6318	

4. The lift chart of logistic regression is 84.7%, shown in Figure 41.

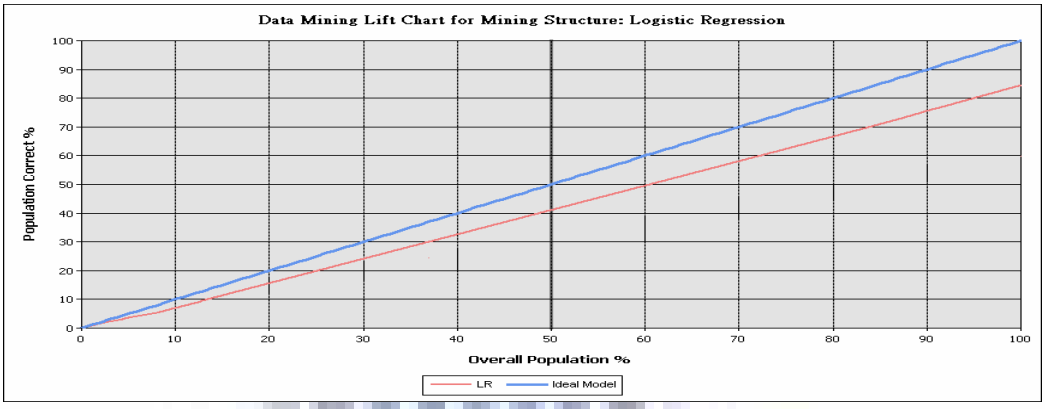


Figure 41. Lift chart of logistic regression.

5. The processing speed of logistic regression is illustrated in Figure 42. The duration of process is 14 seconds.

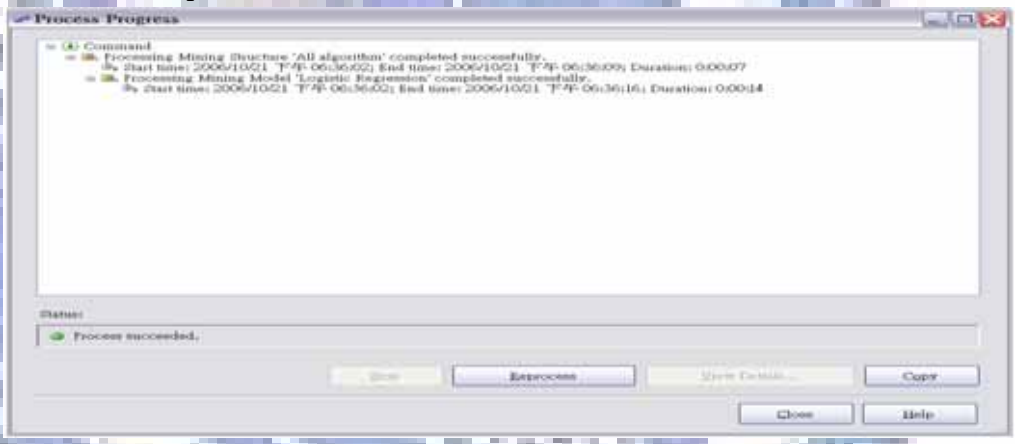


Figure 42. Processing speed of logistic regression.

6. The robustness of logistic regression is shown in Figure 43. For example, if the attribute of machine in 4Ms is “4” then the probability of missing value percentage is 0.01%.

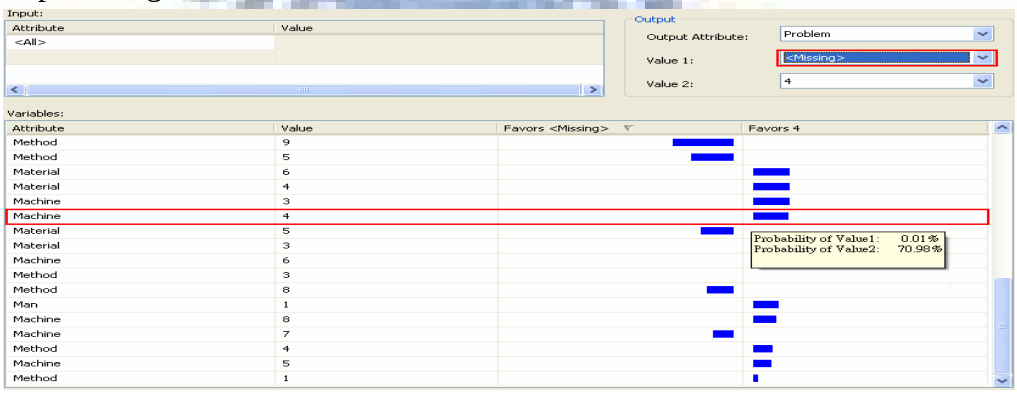


Figure 43. Missing value automatically filled in missing with logistic regression.

5.7.4 Bayesian

1. All parameters of the tool we used are default values, shown as Table 23 [36].

Table 23 Parameters of Bayesian algorithm setting by this research

Parameter	Description	Parameter Setting by this research
MAXIMUM_INPUT_ ATTRIBUTES	Specifies the maximum number of input attributes that the algorithm can handle before it invokes feature selection. Setting this value to 0 disables feature selection for input attributes. The default is 255.	There are 4 input attributes in the study. The value is at least 4.
MAXIMUM_OUTPUT_ ATTRIBUTES	Specifies the maximum number of output attributes that the algorithm can handle before it invokes feature selection. Setting this value to 0 disables feature selection for output attributes. The default is 255.	There are 9 output attributes in the study. The value is at least 9.
MINIMUM_DEPENDENCY_ PROBABILITY	Specifies the minimum dependency probability between input and output attributes. This value is used to limit the size of the content that is generated by the algorithm. This property can be set from 0 to 1. Larger values reduce the number of attributes in the content of the model. The default is 0.5.	After we run the proper range of parameters in the study, the better value is 0.5.
MAXIMUM_STATES	Specifies the maximum number of attribute states that the algorithm supports. If the number of states that an attribute has is greater than the maximum number of states, the algorithm uses the attribute's most popular states and treats the remaining states as missing. The default is 100.	There are 25270 cases in the study. The training data contain 70% within the whole data. The value is at least 18000.

2. The visualized result with Bayesian is represented as Figure 44. For example, if problem = "1" and machine's attribute = "3" then the probability is 0.5.

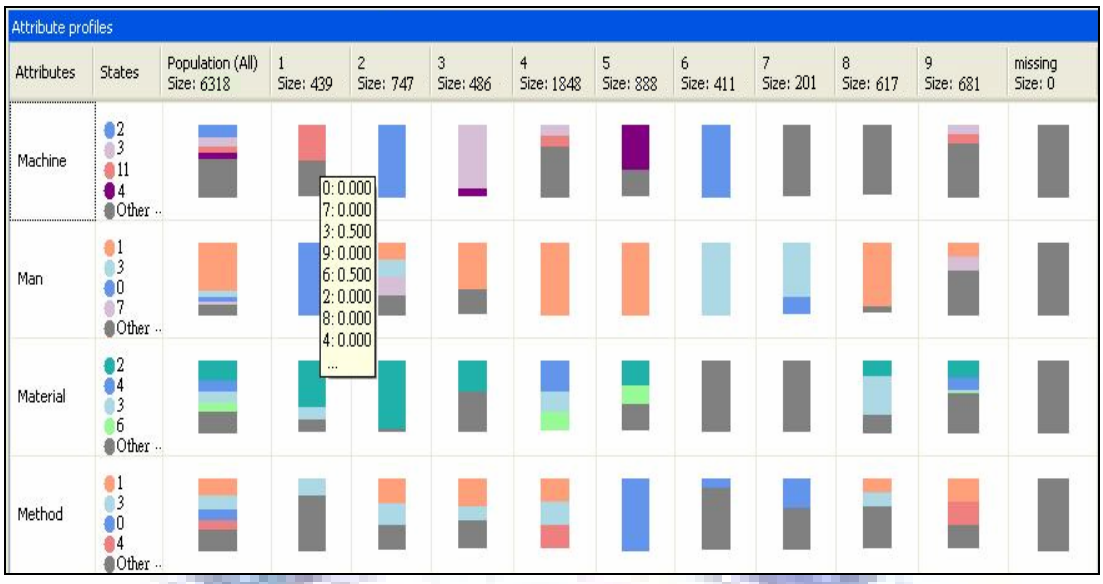


Figure 44. Display of Bayesian classification.

3. For classification and analysis with the Bayesian, there are 5,250 entries of correct data, representing a success rate of 83.1%. The matrix analysis diagram is shown in Figure 45, and Table 24.

		True Class									
		1	2	3	4	5	6	7	8	9	
Predicted Class	1	324	17	8	34	25	26	3		2	439
	2		660	5	43	7			32		747
	3	20		354	47	18		2	22	23	486
	4	11	24		1768	11	23		11		1848
	5	21		28	18	765		4	21	31	888
	6		15		55	22	301			18	411
	7	10	22	15	85			53		16	201
	8		32	16	33	19	13		483	21	617
	9	4	24		45	54		1	12	541	681
		390	794	426	2128	921	363	63	581	652	6318

Figure 45. Matrix classification of Bayesian.

Table 24 Total and accurate entries of Bayesian

Problems / Entries	1	2	3	4	5	6	7	8	9	Total	accuracy rate
Accurate entries	324	660	354	1768	765	301	53	483	541	5250	83.1%
Total entries	390	794	426	2128	921	363	63	581	652	6318	

4. The lift chart of Bayesian is 84.1%, shown in Figure 46.

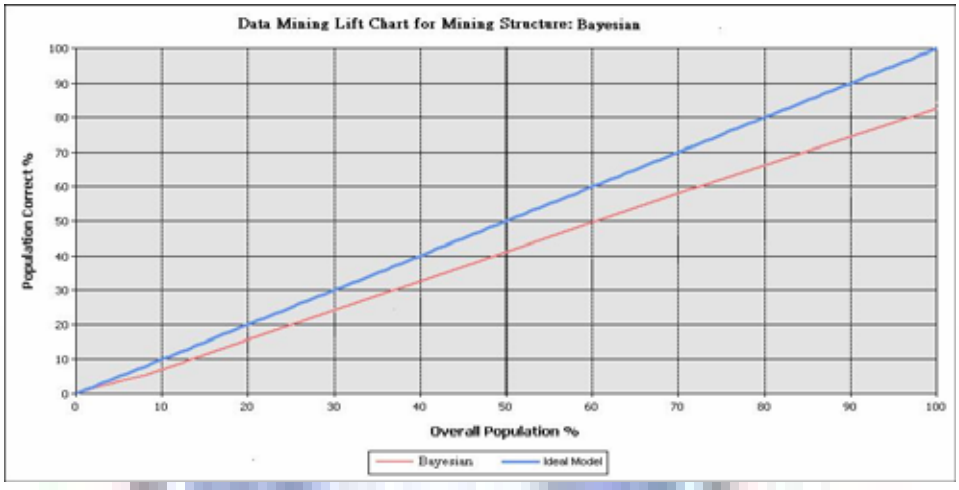


Figure 46. Lift chart of Bayesian

5. The processing speed of Bayesian is illustrated in Figure 47. The duration of process is 15 seconds.

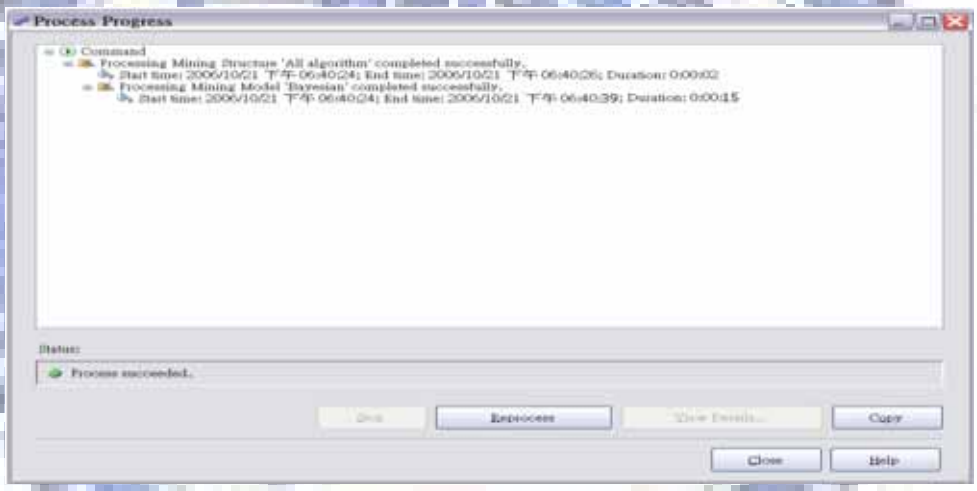


Figure 47. Processing speed of Bayesian.

6. The robustness of Bayesian is shown in Figure 48. For example, the missing value of machine is 0.00%.

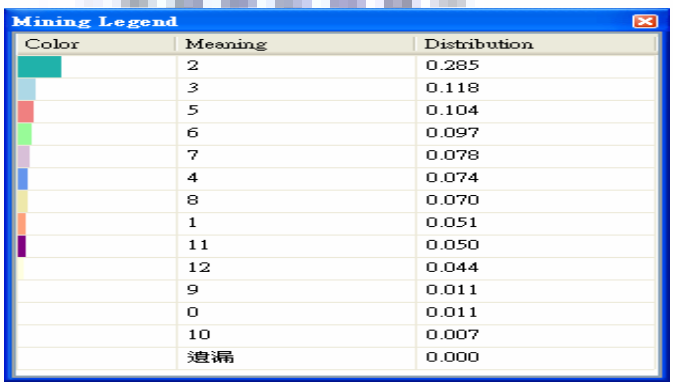


Figure 48. Missing value automatically filled in missing with Bayesian.

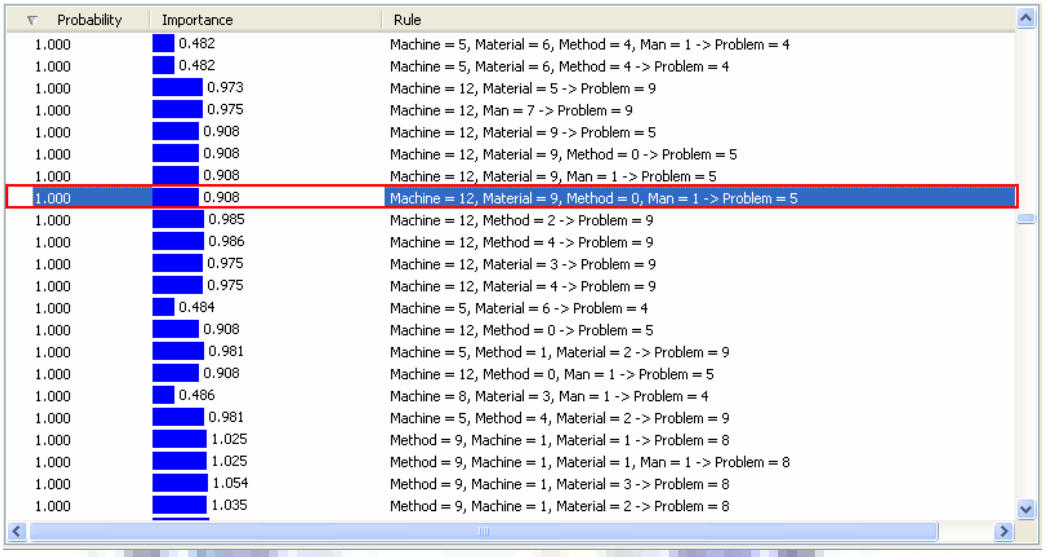
5.7.5 Association rule

1. The parameter setting by the research needed to specify is listed as Table 25 [37].

Table 25 Parameters of association rule algorithm setting by this research

Parameter	Description	Parameter Setting by this research
MAXIMUM_ITEMSET_SIZE	Specifies the maxi- mum number of items that are allowed in an itemset. Setting this value to 0 specifies that there is no limit to the size of the itemset. The default is 3.	Setting *MAXIMUM_ITEMSET_SIZE* to 0 to generate all possible size of the itemset. That is no limit to the size of the itemset.
MINIMUM_SUPPORT	Specifies the minimum number of cases that must contain the itemset before the algorithm generates a rule. Setting this value to less than 1 specifies the minimum number of cases as a percentage of the total cases. Setting this value to a whole number greater than 1 specifies the minimum number of cases as the absolute number of cases that must contain the itemset. The algorithm may increase the value of this parameter if memory is limited. The default is 0.03.	After we run the proper range of parameters in the study, the better value is 0.03.
MAXIMUM_SUPPORT	Specifies the maximum number of cases in which an itemset can have support. If this value is less than 1, the value represents a percentage of the total cases. Values greater than 1 represent the absolute number of cases that can contain the itemset. The default is 1.	After we run the proper range of parameters in the study, the better value is 1.
MINIMUM_PROBABILITY	Specifies the minimum probability that a rule is true. For example, setting this value to 0.5 specifies that no rule with less than fifty percent probability is generated. The default is 0.4.	After we run the proper range of parameters in the study, the better value is 0.4.

2. The visualized result with association rule is represented as Figure 49. For example, if machine = “12” and material =”9” and method = “0” and man = “1” then problem = “5”.



Probability	Importance	Rule
1.000	0.482	Machine = 5, Material = 6, Method = 4, Man = 1 -> Problem = 4
1.000	0.482	Machine = 5, Material = 6, Method = 4 -> Problem = 4
1.000	0.973	Machine = 12, Material = 5 -> Problem = 9
1.000	0.975	Machine = 12, Man = 7 -> Problem = 9
1.000	0.908	Machine = 12, Material = 9 -> Problem = 5
1.000	0.908	Machine = 12, Material = 9, Method = 0 -> Problem = 5
1.000	0.908	Machine = 12, Material = 9, Man = 1 -> Problem = 5
1.000	0.908	Machine = 12, Material = 9, Method = 0, Man = 1 -> Problem = 5
1.000	0.985	Machine = 12, Method = 2 -> Problem = 9
1.000	0.986	Machine = 12, Method = 4 -> Problem = 9
1.000	0.975	Machine = 12, Material = 3 -> Problem = 9
1.000	0.975	Machine = 12, Material = 4 -> Problem = 9
1.000	0.484	Machine = 5, Material = 6 -> Problem = 4
1.000	0.908	Machine = 12, Method = 0 -> Problem = 5
1.000	0.981	Machine = 5, Method = 1, Material = 2 -> Problem = 9
1.000	0.908	Machine = 12, Method = 0, Man = 1 -> Problem = 5
1.000	0.486	Machine = 8, Material = 3, Man = 1 -> Problem = 4
1.000	0.981	Machine = 5, Method = 4, Material = 2 -> Problem = 9
1.000	1.025	Method = 9, Machine = 1, Material = 1 -> Problem = 8
1.000	1.025	Method = 9, Machine = 1, Material = 1, Man = 1 -> Problem = 8
1.000	1.054	Method = 9, Machine = 1, Material = 3 -> Problem = 8
1.000	1.035	Method = 9, Machine = 1, Material = 2 -> Problem = 8

Figure 49. Display of association rule classification.

3. For classification and analysis with the association rule, there are 5,099 entries of correct data, representing a success rate of 80.7%. The matrix analysis diagram is shown in Figure 50, and Table 26.

		True Class									
		1	2	3	4	5	6	7	8	9	
Predicted Class	1	315	15	4	48	41	21	2		23	469
	2	12	641	21	61	23		1	41		800
	3			344	53	14	11	4		32	458
	4	32	18		1717	30			12	1	1810
	5		41	23	36	743	16	3	23	18	903
	6	18	32	19	56		293		36		454
	7	13			41	15		51		30	150
	8		22	12	45	29	19	2	469	22	620
	9		25	3	71	26	3			526	654
		390	794	426	2128	921	363	63	581	652	6318

Figure 50. Matrix classification of association rule.

Table 26 Total and accurate entries of association rule

Entries \ Problems	1	2	3	4	5	6	7	8	9	Total	accuracy rate
Accurate entries	315	641	344	1717	743	293	51	469	526	5099	80.7%
Total entries	390	794	426	2128	921	363	63	581	652	6318	

4. The lift chart of association rule is 81.0%, shown in Figure 51.

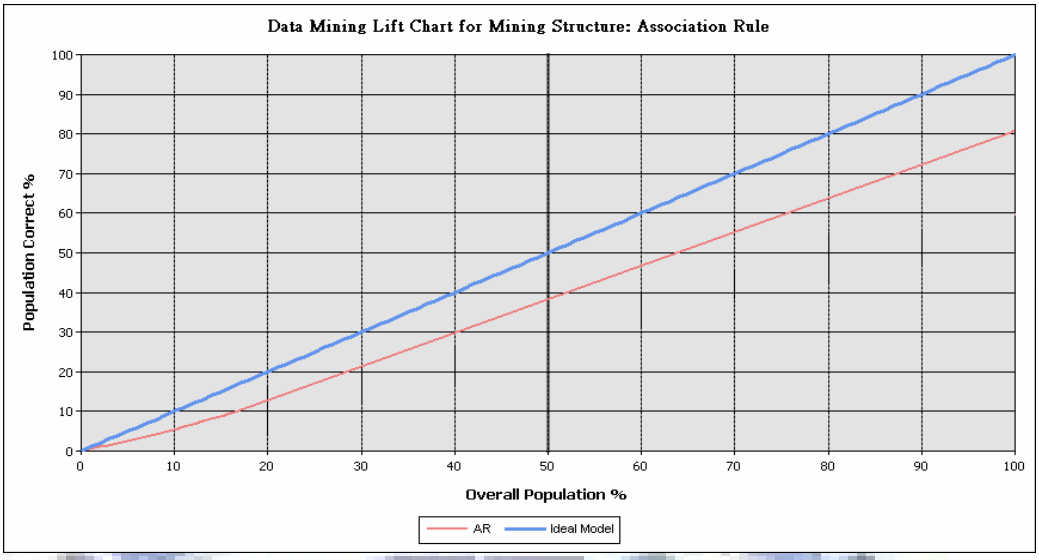


Figure 51. Lift chart of association rule.

5. The processing speed of association rule is illustrated in Figure 52. The duration of process is 6 seconds.

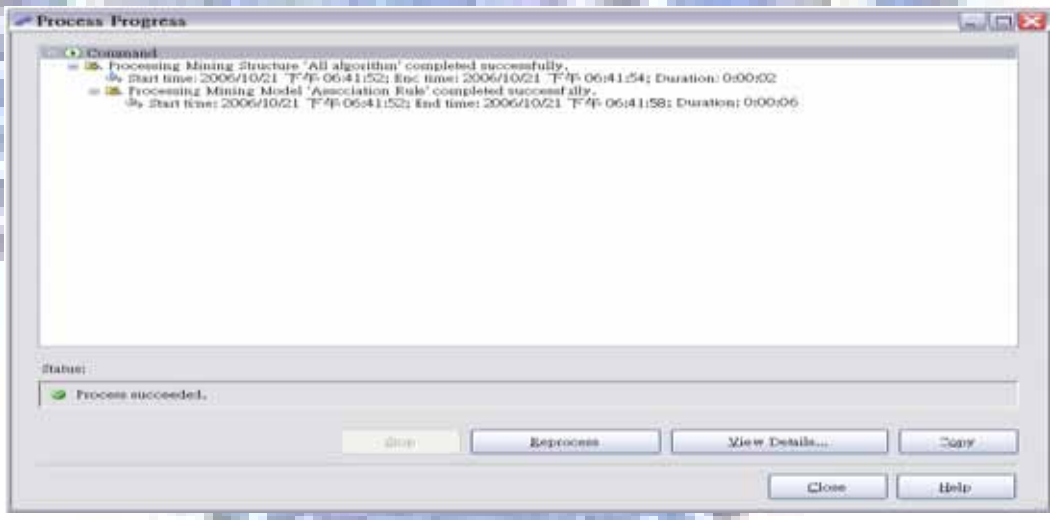


Figure 52. Processing speed of association rule.

6. It is not found in the tool to provide the robustness of association rule.

5.8 Comparing and Evaluating Data Mining Models

Classification and prediction methods can be compared and evaluated according to the following criteria [10].

1. Predictive accuracy: It relates to the capability of the model to correctly predict the class label of new or previously undiscovered data.

2. Speed: It relates to the calculation costs in generating and utilizing the model.
3. Robustness: It is the capability of the model to predict according to noisy data or data with missing values.
4. Scalability: It relates to the ability to construct the model efficiently given large amounts of data.
5. Interpretability: It relates to the level of comprehension and insight that applied by the model.

Comparisons and evaluations are conducted in accordance with the five algorithms - decision tree, neural network, logistic regression, Bayesian, and association rule-applied in this study, illustrated as follows. Moreover, the relative parameters setting of all algorithms are listed in Table 27.

1. Predictive accuracy: SQL Server 2005 offers two tools, the classification matrix and the lift chart, for evaluating the accuracy of data mining models.

(1) Classification matrix:

Table 27 Accuracy rate of classification matrix

Item \ Algorithm	Decision Tree	Neural Network	Logistic Regression	Bayesian	Association Rule
Accuracy entries	5680	5326	5294	5250	5099
Total entries	6318	6318	6318	6318	6318
Accuracy rate	89.9%	84.3%	83.8%	83.1%	80.7%

(2) Lift chart:

The accuracy of the decision tree algorithm is much better than the other four algorithms. The accuracy of five algorithms, shown in Figure 53, can be ranked as follows.

1. Decision tree.
2. Neural network.
3. Logistic regression.

4. Bayesian.

5. Association rule.

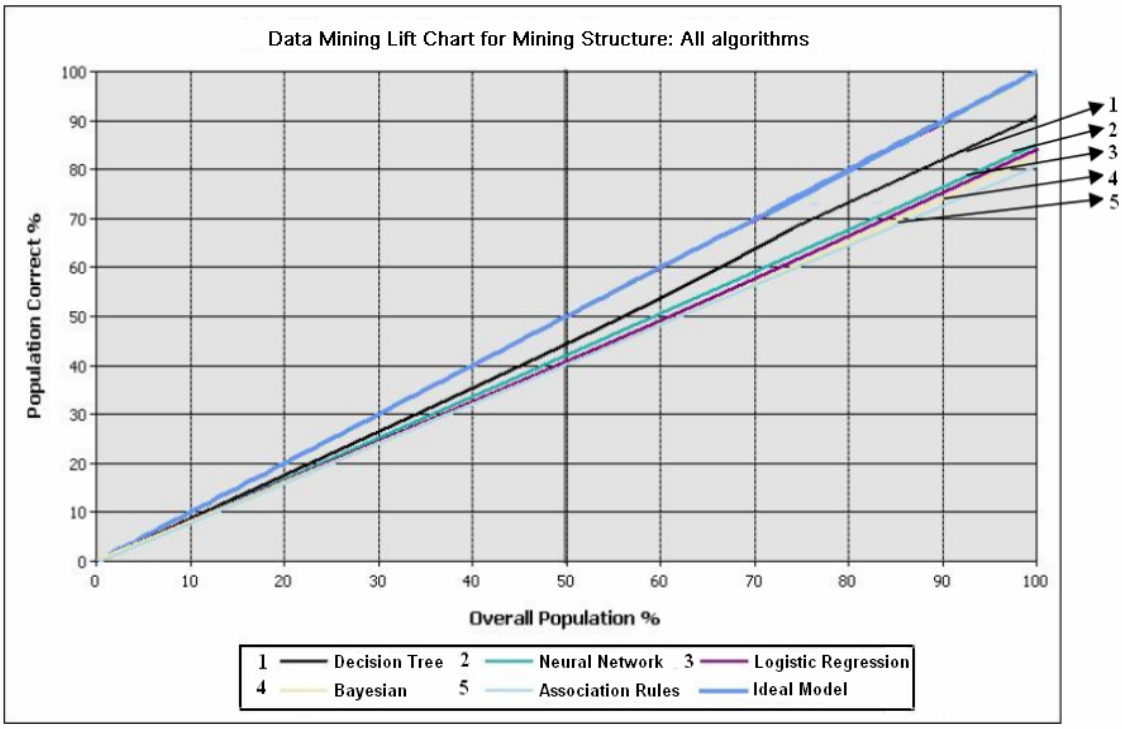


Figure 53. Lift chart of proposed data mining algorithms.

2. Speed: Time consumed for the execution of five classification algorithms, illustrated in Figure 54. The duration of process time of five algorithms, shown in Figure 54, can be ranked as follows.

   1. Association rule.

   2. Logistic regression.

   3. Bayesian.

   4. Decision tree.

   5. Neural network.

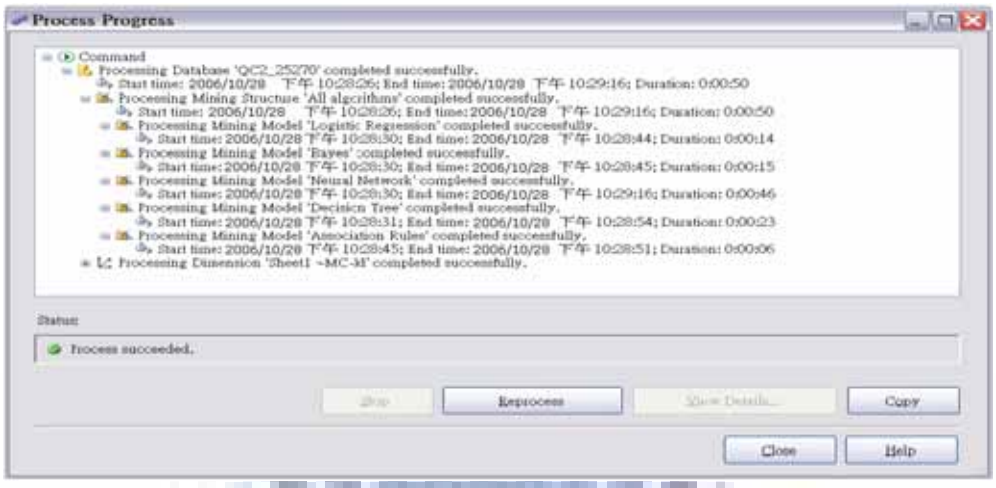


Figure 54. Process progress and speed for mining structures.

3. Robustness: “Missing” is automatically assigned for replacement in the instances of the given noise data or missing value. We can rank in the order of robustness, decision tree and Bayesian are better algorithms, which is followed by neural network and logistic regression; association rule is least better.
4. Scalability: The five data mining and classification computing tools used in this study are all designed using the parallel structure, which does not only accommodate large corporate data sets and allow the simultaneous data storage and retrievals by thousands of users, but also respond to millions of data analyses and inquiries on a daily basis [29]。
5. Interpretability: For the five classification algorithms used in this study, each one is illustrated by the graphical interface. However, the neural network and the logistic regression are composed of complex weights and activation functions, which have long been considered as the “black box” rules. Through the use of the content viewer, it permits users only to understand the importance of the variables but does not enable the comprehensive observations of the structures of these two algorithms.

Through the inclusion of six above evaluation indexes and discussions with the experts in this field of specialization, weighting is assigned to each index according to the magnitude of importance assessed by the corporations, followed by the sequential

ranking of each index based on its advantages and disadvantages. The effectiveness of five algorithms can therefore be statistically computed, illustrated in Table 28.

Table 28 Summary of evaluating data mining models

Evaluated Items \ Algorithms / Weight		Decision Tree	Neural Network	Logistic Regression	Bayesian	Association Rule
Classification Matrix(%)	0.4	89.9	84.3	83.8	83.1	80.7
Lift Accuracy(%)	0.4	91.0	85.3	84.7	84.1	81.0
Speed(sec)	0.05	23	46	14	15	6
Robustness	0.05	high	high	high	high	low
Scalability	0.05	high	high	high	high	high
Interpretability	0.05	high	low	low	medium	medium

*The preference: “high” set to 3, “medium” set to 2, “low” set to 1.

Simple Additive Weighting Method

After we obtain the five criteria to measure the performance of the system, the measurement procedures are listed as follows. Simple additive weighting method is one of most used in multiple criteria decision making (MCDM) [38]. The calculation steps are as follows.

1. All evaluation indices need to be standardized.
   (1) Standardization of performance index

$$r_{ij} = \frac{x_{ij}}{\max_i x_{ij}}, \quad \forall_j$$, where $j=1,2,\ldots n$

   (2) Standardization of cost index

$$r_{ij} = \frac{\min_i x_{ij}}{x_{ij}}, \quad \forall_j$$, where $j=1,2,\ldots n$

2. Calculate the performance of all selected solutions

$$S_i = \sum_{j=1}^{n} w_j r_{ij} \quad , \quad i = 1,2,\ldots,m$$, where $\sum_{j=1}^{n} w_j = 1$

3. According to the value of S_i, the selected solutions can be ranked. The lager the values are, the higher preferences of solutions are.

According to the calculations of above steps, the cases evaluations of the study are listed as follows.

Step1: Standardization of performance and cost can be calculated as follows. The x-axis stands for the values of classification accuracy, lift accuracy, speed, robustness, scalability, and interpretability; y-axis stands for decision tree, neural network, logistic regression, Bayesian, and association rule algorithm.

$$R = \begin{array}{c} \begin{matrix} c & l & sp & r & s & i \end{matrix} \\ \begin{pmatrix} 89.9 & 91.0 & 23 & 3 & 3 & 3 \\ 84.3 & 85.3 & 46 & 3 & 3 & 1 \\ 83.8 & 84.7 & 14 & 3 & 3 & 1 \\ 83.1 & 84.1 & 15 & 3 & 3 & 2 \\ 80.7 & 81.0 & 6 & 1 & 3 & 2 \end{pmatrix} \end{array} \begin{matrix} D \\ N \\ L \\ B \\ A \end{matrix}$$

After the standardization, we obtain the matrix as follows.

$$R = \begin{array}{c} \begin{matrix} c & l & sp & r & s & i \end{matrix} \\ \begin{pmatrix} 1 & 0.261 & 1 & 1 & 1 & 1 \\ 0.938 & 0.937 & 0.130 & 1 & 1 & 0.333 \\ 0.932 & 0.931 & 0.429 & 1 & 1 & 0.333 \\ 0.924 & 0.924 & 0.4 & 1 & 1 & 0.667 \\ 0.898 & 0.890 & 1 & 0.3333 & 1 & 0.667 \end{pmatrix} \end{array} \begin{matrix} D \\ N \\ L \\ B \\ A \end{matrix}$$

The abbreviations of c, l, sp, r, s, and i stand for classification accuracy, lift accuracy, speed, robustness, scalability, and interpretability; D, N, L, B, and A stand for decision tree, neural network, logistic regression, Bayesian, and association rule.

Step2: Assign the weight, W = (0.4, 0.4, 0.05, 0.05, 0.05, 0.05), and the selected alternative weights after being calculated as follows:

D=1*0.4+1*0.4+0.261*0.05+1*0.05+1*0.05+1*0.05=0.96305

N=0.87315

L=0.88335

B=0.89255

A=0.8652

Step3: The better solution rank as (D,B,L,N,A). Therefore, the evaluation result ranks as follows:

1. Decision tree.
2. Bayesian.
3. Logistic regression.
4. Neural network.
5. Association rule.

Although neural network can be applied in many fields, such as industry, financial, and telecommunication etc, the interpretability and process speed are worse than other four proposed algorithms. In the Microsoft sample database, Adventure Works, the lift accuracy of decision tree, neural network and Bayesian can be ranked as Bayesian (66.09%), decision tree (63.58%) and neural network (48.06%) [11]. In our study for semiconductor packaging industry, the lift accuracy of decision tree, neural network and Bayesian can be ranked as decision tree (91.0%), neural network (85.3%) and Bayesian (84.1%). Moreover, there are six merits according to the evaluation of MCDM that decision tree, logistic regression, Bayesian are better than neural network listed as follows.

1. Neural network utilize the iteration to renew weights and thresholds. It cost a large amount of computing resource.
2. We did not realize it that the numbers of neurons are better in the training processes of neural network. Therefore, we have to utilize the try and error strategy.
3. The knowledge structure of neural network is implicit and the interpretability is too hard to understand.
4. The knowledge structure of decision tree is easy and the interpretability is concise to understand.

5. Although the accuracy of logistic regression is worse than that of neural network, the duration of process time of logistic regression is better than that of neural network. This is because logistic regression without hidden layers.
6. Although the accuracy of Bayesian is worse than that of neural network, the duration of process time and interpretability of Bayesian are better than those of neural network.

After we observe and appraise the situation, the decision tree method is more effective and accurate than the other methods to be applied to the quality problems in the semiconductor packaging industry. Therefore, we adopted the decision tree algorithm as the data mining tool for the quality improvement system in this study.

5.9 OLAP Analysis

OLAP basically provides some functions, such as roll-up, drill-down, slice and pivot and etc. We adapt drill-down as an example to operate to make decision makers can analyze the data collected from top to bottom. If the executives desire to understand further whether the process causes the product defeats, they can also track the details of process by using the functions of OLAP. Hence, the three-dimension cube consisting of month, problem, and quantity can be illustrated in Figure 55.

Through OLAP, we can clearly demonstrate the quantity of each month for each problem in 2004. We will also derive the major cause of product defects from the manufacturing process. Nine clusters of problems can be ranked as follows.

1. Cluster 4: Inner lead broken/inner lead bending/inner lead delamination/outer lead damage.
2. Cluster 5: Short/Open.
3. Cluster 2: Shrinkage/Sprocket hole damage.
4. Cluster 9: Resin wrapping drawn-in object /tape indent.
5. Cluster 8: Resin poor wrapping/riding.

6. Cluster 1: Tape resin adhesion.
7. Cluster 3: Resin overflow.
8. Cluster 6: Overlap marking.
9. Cluster 7: Void.

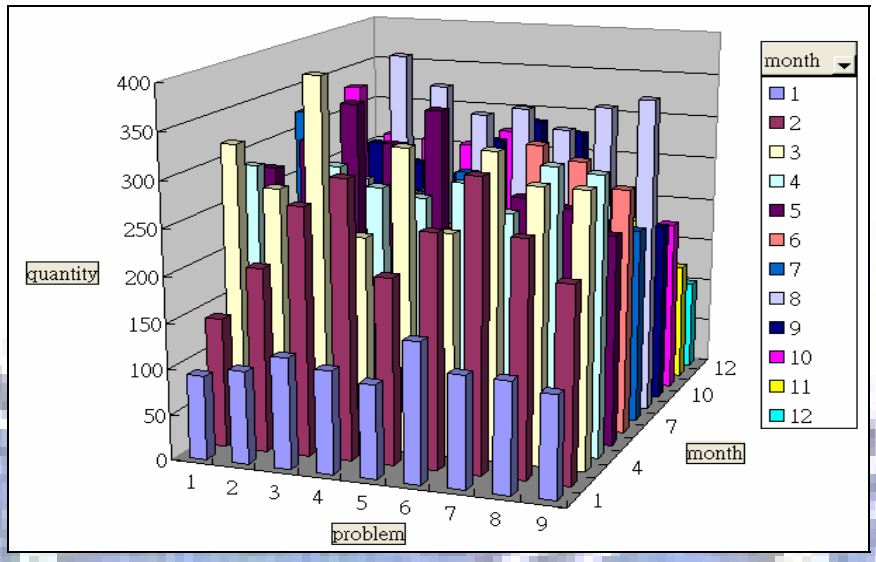


Figure 55. Drill-down of OLAP display for the three-dimension cube.

Through OLAP, among four attributes, man, machine, material and method, we also can roll up and explore the machine's major attributes, shown in Figure 56, are the ILB pressure error, Potting pressure error, ILB sensor error and ILB tool head error in the semiconductor packaging industry.

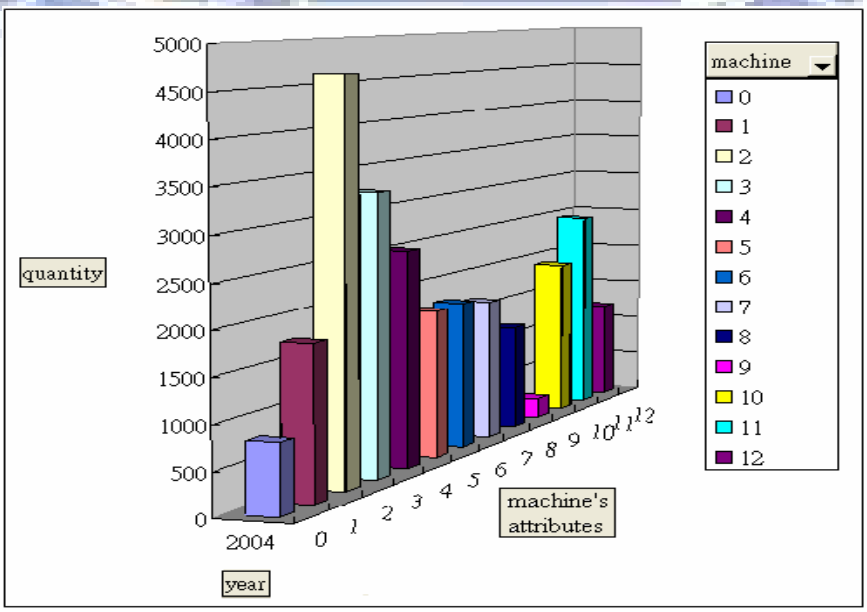


Figure 56. Roll-up of OLAP display for the machine's attributes.

5.10 Practical Operation of a Better Model

To establish the database, we have collected the data form January to December in 2005. With the decision tree provided by the quality improvement system for reference of man operation analysis and problem settlement form January to December in 2004. We will discover the decision tree is more effective than neural network, logistic regression, Bayesian, and association rule. Moreover, decision tree algorithm makes knowledge rules easy to understand than the other proposed algorithms, thus, the decision tree selected as algorithm, serves as the nucleus of the data mining engine. We can then use decision tree algorithm to perform detailed calculation on all kinds of product defective problems to obtain better parameters in all manufacturing processes.

Below is the research of this paper in the major causes leading to product failures in the ILB and potting processes, pressure and temperature, as an example. The details of computation steps refer to 4.4.2.1.

Step 1: Prepare previously classified training data, shown in Table 29.

Table 29 ILB training data set

Item	Ilb_pressure (gf)	Ilb_temp (℃)	Potting_pressure (Kpa)	Potting_temp (℃)	Problem_class
1	<30	>380	>500	110	NG
2	30~60	>380	>500	70	OK
3	>60	>380	<=500	110	OK
4	<30	>380	>500	70	NG
5	>60	>380	<=500	110	OK
6	30~60	<=380	>500	70	OK
7	>60	<=380	>500	70	NG
8	<30	<=380	>500	110	NG
9	>60	>380	>500	90	OK
10	>60	<=380	<=500	110	NG
11	30~60	<=380	<=500	90	OK
12	<30	>380	<=500	90	OK
13	<30	<=380	<=500	90	OK
14	30~60	>380	<=500	90	OK
15	<30	<=380	>500	90	NG
16	<30	>380	<=500	110	OK
17	30~60	>380	<=500	110	OK
18	>60	<=380	<=500	70	NG

Step 2: Establish a decision tree node. Determine whether or not this node is a leaf node, or calculate information gain for the test attribute. The calculation method is shown as follows in steps 3-5.

Step 3: The expected information of the classified data samples selected for calculation: Let s_i be the number of samples of S in class C_i. The expected information needed to classify is given by

$$I(S_1,S_2,...,S_m) = -\sum_{i=1}^{m} P_i \log_2(P_i)$$

The class label attribute, problem, has two distinct values (namely, {OK, NG}); therefore, there are two distinct classes (m = 2). Let class C_1 is OK, C_2 is NG. To compute the information gain of each attribute, we first compute the expected information needed to classify a given sample: $I(S_1, S_2) = I(11, 7) = 0.964$

Step 4: The expected information of the test attribute selected for calculation:

First, select the ILB_pressure as test attribute. There are three different values for this test attribute: {<30, 30~60, >60gf}. Therefore, training data may be divided into three subsets: {S_1, S_2, S_3}. Thus, when calculating the expected information of the ILB_pressure, the following two steps may be used:

1. Calculate the expected information of the three subsets:

   (1) S_1 (ILB_pressure = " <30"): $I(s_{11},s_{21}) = I(3,4) = 0.985$

   (2) S_2 (ILB_pressure = "30~60"): $I(s_{12},s_{22}) = I(5,0) = 0$

   (3) S_3 (ILB_pressure = ">60"): $I(s_{13},s_{23}) = I(3,3) = 1.0$

2. Calculate the expected information with the ILB_pressure as test attribute:

   $E(\text{ILB\_pressure}) = ((3+4)/18)*I(s_{11},s_{21}) + ((5+0)/18)*I(s_{12},s_{22}) + ((3+3)/18)*I(s_{13},s_{23}) = 0.716$

Step 5: Calculate the information gain of the selected test attribute:

Split Gains are segmented subsets; the computation method is given by

$\text{Split Gains} = -(7/18)\log_2(7/18) - (5/18)\log_2(5/18) - (6/18)\log_2(6/18) = 1.572$

Gain(ILB_pressure) = (I(S_1, S_2, S_3) - E(ILB_pressure)) / Split Gains

= (0.964-0.716)/1.572 = 0.158

Step 6. Repeat steps 2-5 until the information gain of the test attributes are completely calculated.

(1) Gain(ILB_pressure) = 0.158 (2) Gain(ILB_temperature) = 0.140

(3) Gain(Potting_pressure) = 0.140 (4) Gain(Potting_temperature) = 0.060

Step 7: Select the test attribute with the highest information gain to act as the node of partition for the decision tree.

Since the ILB_pressure has the highest information gain at all levels, thus, it has been selected as test attribute. Moreover, the ILB_pressure acts as the starting node for the decision tree, moving downward and branching off into other nodes.

Step 8: To complete set up of the decision tree, follow this sequence of steps to find test attribute nodes at each level.

Training data will be partitioned in accordance with the selected test attribute.

The completed decision tree is illustrated in Figure 57.

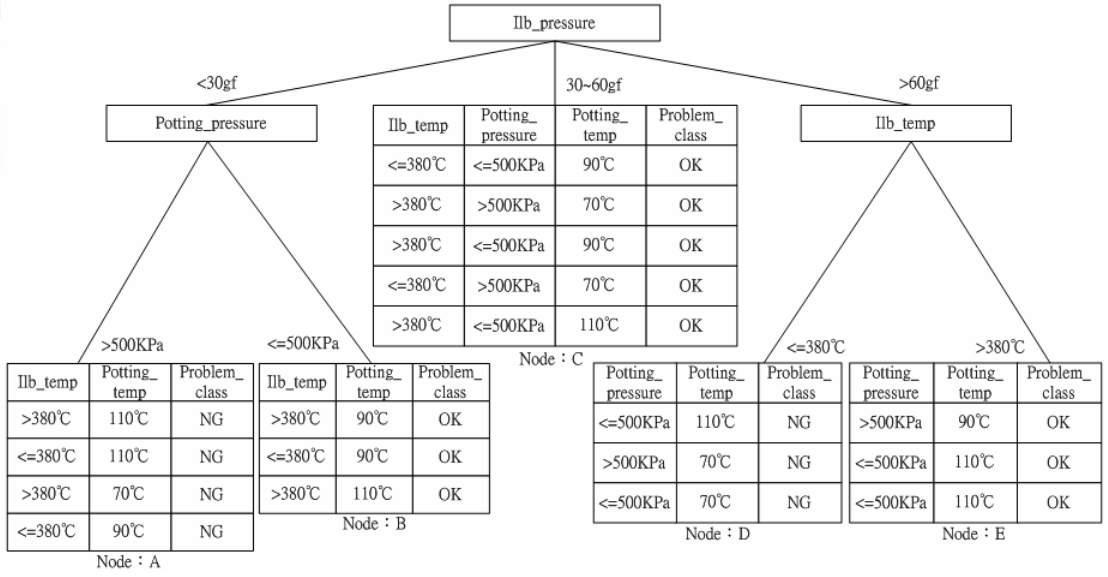


Figure 57. The decision tree for training data.

The knowledge rules processed by the decision tree described above allow for the convenient gathering of information, by tracing this information along the path

from root node to leaf node. In this research, knowledge rules are expressed through the rule based knowledge presentation method, thus, revealing the knowledge rules possessed by the decision tree. These rules are described as follows.

1. IF *ILB_pressure* = “<30gf” AND *Potting_pressure* = ”>500KPa” THEN *Problem_class* = ”NG”
2. IF *ILB_pressure* = “<30gf” AND *Potting_pressure* = ”<=500 KPa” THEN *Problem_class* = ”OK”
3. IF *ILB_pressure* = “30~60gf” THEN *Problem_class* = ”OK”
4. IF *ILB_pressure* =“>60gf” AND *ILB_temp* = ”<=380℃” THEN *Problem_class* = ”NG”
5. IF *ILB*_pressure = “>60gf” AND *ILB*_temp = ”>380℃” THEN *problem _class* = ”OK”

From the knowledge rules, we can collect literature survey [24] and discuss with experts. Therefore, we can obtain four aspects, i.e. man, machine, material and method, to improve the processes listed as follows.

1. Man: it is important to focus on the training of beginners and they should obey the standard operation procedures on their duties.
2. Machine: the parameters should be adjusted to proper ranges of each process, such as temperature setting, pressure setting, and time setting. The related suggestions are listed as follows.
   (1) Coplanarity control is mainly affected by temperature and pressure parameters, main method of solution is in the leveling and temperature distribution in the bonding press head under different temperatures, must ensure in the bonding process maintain a even distributions of temperature.
   (2) To overcome disadvantages in the potting process, can use screen printing manufacturing process instead.

(3) Using the vacuum screen-printing process could still cause void, but one could utilize second pressure differential vacuum products to increase the void resistance in manufacturing and increase surface evenness of the product.

3. Material: they should enhance the quality control of material, such as selection of glues, tape quality, and inner lead quality, etc. They ought to pay attention on inner lead quality, such as lead bonding, lead fracture, lift, bending, etc.
4. Method: they should reinforce the operations resulting in major defects, such as machine positioning, temperature condition error, and improper glue clearing.

- Inner lead bonding and positioning:

(1) Maintain the positioning precision between inner lead and bump.

(2) Maintain the high lead number, fine distance in bumps, and the positioning and leveling between inner leads and bonding press head.

5.11 Benefits Analysis

The decision tree algorithm as the kernel engine of data mining system is actually applied in quality improvement in the semiconductor packaging industry. Moreover, we have discussed with the domain specialists and refer to the literature survey. Therefore, based on the mined knowledge rules, the company is able to explore the characteristics and attributes of the critical reason causes from quality, allowing it to focus on top of the quality issue. The benefits analysis of the quality improvement in product is illustrated as follows.

1. Increase the rate of product yield

The quality improvement system has improved the rate of product yield: the average rate before the use of the quality improvement system was 94.59%, and it became 99.67% after the quality improvement system was used, improving the overall rate by 5.37%, shown in Figure 58.

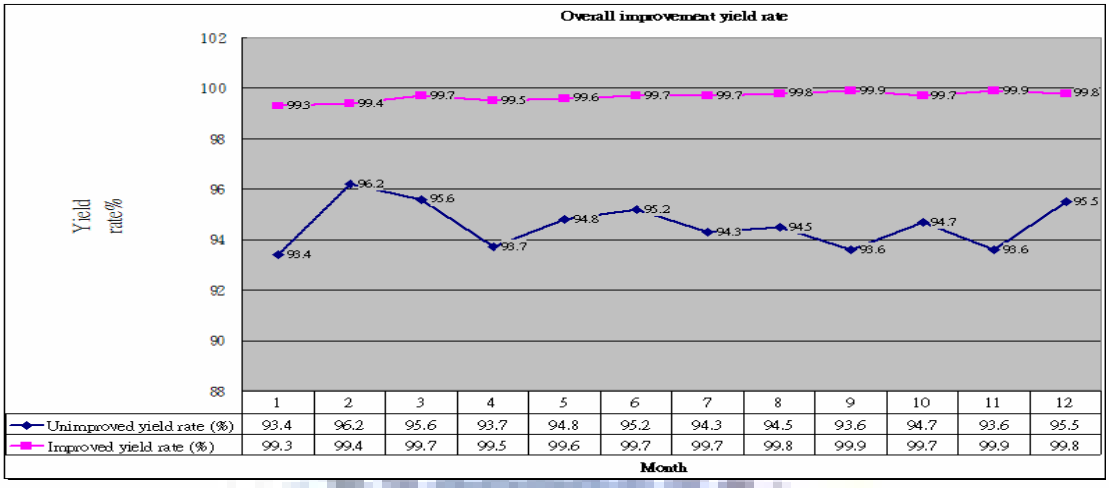


Figure 58. Diagram of overall improvement rate curves for each quality problem monthly.

2. Lower the manufacturing cycle time

The quality improvement system has improved the manufacturing cycle time: the average cycle time before the use of the quality improvement system was 7.4 days and it became 5.3 days after the quality improvement system was used, improving the overall rate by 28.4%, shown in Figure 59.

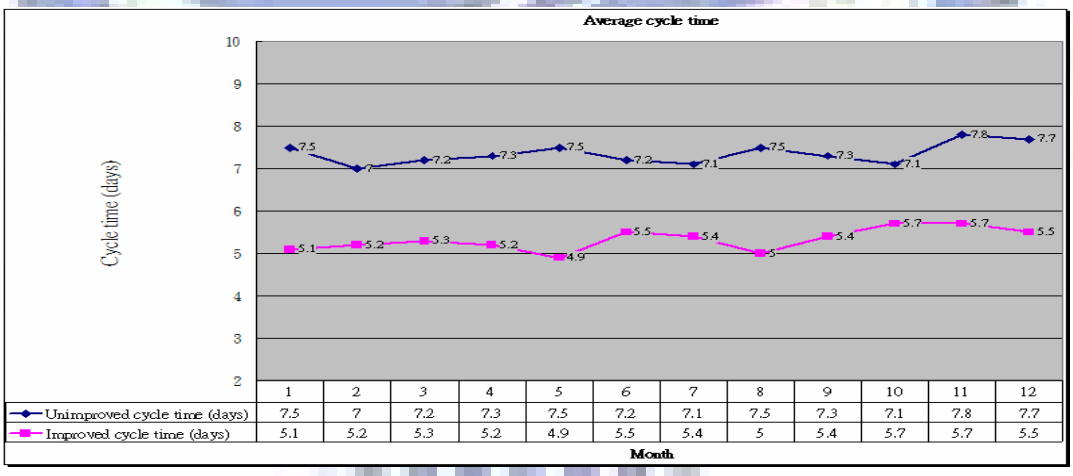


Figure 59. Diagram of average cycle time curves monthly.

3. Lower the frequency of machine holding lot

The quality improvement system has improved the frequency of machine holding lot: the average frequency before the use of the quality improvement system was 53.7 times and it became 29.3 times after the quality improvement system was used, improving the overall rate by 45.4%, shown in Figure 60.

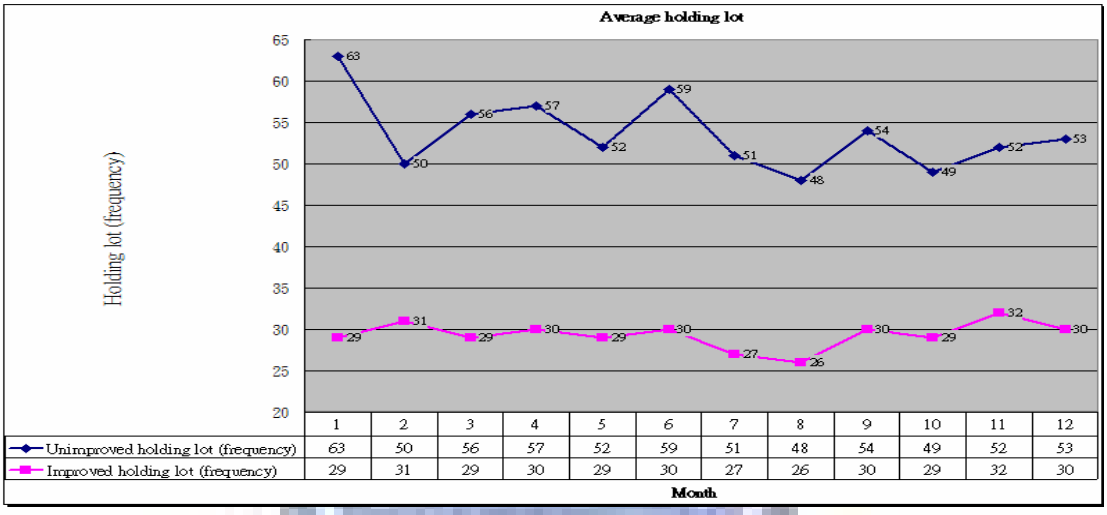


Figure 60. Diagram of average holding lot curves monthly.

Chapter 6
Contributions

After finishing the practical system implementation, we obtain the research contributions for the reference of semiconductor packaging industry and future works for the researchers in their relative fields.

6.1 Contributions

In order to meet the target mentioned above, our research involves in using data warehouse, OLAP, decision tree, neural network, logistic regression, Bayesian, and association rule algorithms to perform classification analysis to bring about yield increases in the manufacturing process of semiconductor packaging industry. Comparing the accuracy and applicability of proposed algorithms, we can provide a decision-making policy for the executives. With a view to identifying the causes of problems, countermeasures of main variables to the problems, and making decisions quickly, we can solve quality problems efficiently. The results and contributions of this research are listed as follows.

1. It is found that decision tree algorithm is more effective and appropriate than neural network, logistic regression, Bayesian and association rule to analyze the quality problems in the semiconductor packaging industry.
2. Lower the defect rate of parts by 5.37% to improve product yield.
3. Lower the cycle time by 28.4% during the manufacturing process.
4. Lower the frequency of machine holding lot by 45.4% during the manufacturing process.

6.2 Future Works

The related works in the near future can be developed toward the following three aspects.

1. Owing to several stages and complicated processes, we can only devote to

focusing on discovering the ILB and Potting in the packaging process to find out the main causes of defect products. If other processes can be taken into account, the higher yield rate will be promoted.

2. The product quality improvement system can be integrated with the customer relationship management system to increase the clients' satisfactions.
3. Moreover, the proposed system can be embedded into experts system so as to store the analyzed knowledge with various domains through data mining tools. With the knowledge databases, it will assist quality control engineers to figure out the countermeasures for solving the problems and to eventually enhance the yield rate.
4. The study utilize back-propagated network of neural network algorithm in Microsoft SOL server 2005. There are 7 parameters in the system; however, we can not adjust the most important parameters of the layers of hidden layer, learning speed, imbedded into the system. That is so-called a "black-box" and regrets to compare with the other data mining models. In view of this, the other researchers can utilize other tools or design the programs by themselves to deal with the 9 clusters of defective types in the study, so that they can observe the convergence and divergence among each cluster and adjust the parameters, such as hidden layers and learning speed, etc. Moreover, they can also observe performance that neural network is adapted to classify and predict in the semiconductor packaging industry. Maybe, there might exit an application better than decision tree algorithm.

Reference

[1] Agrawal, R., Imilienski, T. and Swami, A., “Mining Association Rules between Sets of Items in Large Databases”, In Proceedings of ACM SIGMOD international Conference on Management of Data, pp. 207-216, 1993.

[2] Berry, Michael J. A., Gordon S. Linoff, “Data mining techniques: for marketing, sales, and customer support”, John Wiley & Sons, New York, pp. 297-305, 1997.

[3] Brachman, R. J., T. Khabaza, W. Kloesgen, G.P. Shapiro, E. Simoudis, “Mining business databases”, Communications of the ACM, 39(11), pp. 42-48, 1996.

[4] Cabena, P., P. O. Hadjinian, R. Stadler, DR. J. Verhees, and A. Zanasi, ”Discovering Data Mining from Concept to Implementation”, Prentice Hall, pp. 12, 1997.

[5] Chaudhuri, Surajit, Umeshwar Dayal, “An overview of data warehousing and OLAP technology”, SIGMOD, 26, pp. 65-74, 1997.

[6] Chen, S. W. et al., “Electronic packaging technology and material”, Gau Lih Books Co., Ltd., pp. 149-161, 2004.

[7] Chen, Zhengxin, “Data mining and uncertain reasoning: an integrated approach”, John Wiley & Sons, New York, pp. 81-84, 2001.

[8] Fan, Cang-Ren, “A Web-Based Integrated Data Mining System for Computer Integrated Manufacturing- A Case Study On Semiconductor Packaging Industry”, Institute of Information Management, National Chiao Tung University, pp. 61-63, 2003.

[9] Frawley, W. J., G. Paitetsky-Shapiro, C. J. Matheus, “Knowledge discovery in database: an overview. Knowledge Discovery in Database”, AAAI/MIT Press, pp. 1-30, 1991.

[10] Han, Jiawei, Micheline Kamber, “Data mining: concepts and techniques”, Morgan Kaufmann Publishers, pp. 41-78, pp. 285-328, 2001.

[11] Hsieh, C. B., “Data Mining and Business intelligence: SQL Server 2005”, Ting Mao Publish Company, pp. 129-149, pp. 153-170, pp. 278-291, 2005.

[12] Hsu, G. H., “An Advanced Packaging Technology: Wafer Packaging Technology”, Materials Magazine, 151, pp. 86-91, 1991.

[13] Huang, Cheng-Lung, ”Semiconductor Production Rate Prediction with Data Mining Approach”, Information Management, Hua Fan University, pp. 36-44 2004.

[14] Inmon, W. H., “Building the Data Warehouse”, John Wiley & Sons, New York, pp. 71-135, 1996.

[15] Kleissner, C., “Data mining for the enterprise”, IEEE Proceedings of the 31st Annual Hawaii International Conference on System Sciences, Vol.7, pp. 295-304, 1998.

[16] Lee, Perry, “Constructing A Semiconductor Manufacturing Data Mining Framework, Developing A Decision Tree Algorithm for Classification, and Conducting Empirical Studies”, Industrial Engineering and Engineering Management, National Tsing Hua University, pp. 66-78, 2002.

[17] Quinlan, J. Ross, “C4.5: Programs for machine learning”, Morgan Kaufmann Publishers, pp. 81-106, 1993.

[18] Rau H., J. W. Tzeng, L. S. Huang, “Defect classification and recognition for semiconductor manufacturing packaging”, pp. 849-855, 1998.

[19] Shen, C. Y., “Data warehouse and analysis services: SQL Server 2000 OLAP solutions”, Kings Information Co. Ltd., pp. 2.7-2.8, 2001.

[20] Tsai, Tsan-Lian, “Research in Taiwan LCD driver IC finishing process optimum work distribution model”, National Chiao Tung University Masters in High Executive Management, pp. 11-21, 2001.

[21] Tseng, S. S. et al., “Data mining”, Flag Publishing Co., Ltd, pp. 5.24-35, 2005.

[22] Tseng, Wei-Lin, “The Development of Internal Process Quality Handling Model for Printed Circuit Board Industries Using Data Mining Techniques”, Industrial Engineering and Management, Yuan Ze University, pp. 53-71, 2004.

[23] Wang, Chin-Tsan, Qiu Zhi-Wei, Xu Chuan-Yuan and Chen Ming- Zhen, “A Study of Quality Theorems on Process Capability of Mechanic Manufacturing Industry - An Example of MH Company”, The 6th Reliability and Maintainability Symposium Proceedings, pp. 1-15, 2003.

[24] Weiss, S. M. and C. A. Kulikowski, “Computer Systems That Learn—Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert System”, Morgan Kaufmann, pp. 93-110, 1991.

[25] Yang et al., “Analysis and reliability assessment in inner lead welding machine characteristics for tape carrier package IC”, Electronics and Materials Magazine, pp. 132-142, 2001.

[26] Yin, H. G., “SQL Server 2005 Data Mining Bible”, XBOOK Marketing Co., pp. 4.3-4.21, pp. 5.5-5.14, pp. 6.3-6.9, pp. 7.7-7.21, pp. 10.7-10.17, pp. 12.3-12.8, pp. 15.2-15.20, 2006.

[27] http://ieknet.itri.org.tw/_~~mvc2~mvc1_glwpmgn/_~~publish~pdf_upload~xyz/ 2004.

[28] http://www.kdnuggets.com/polls/2006/data_mining_methods.htm.

[29] http://www.microsoft.com/taiwan/sql/prodinfo/overview/whats-new-in-sqlserver 2005.mspx.

[30] http://www.microsoft.com/taiwan/technet/prodtechnol/sql/2000/maintain/dmperf .aspx.

[31] http://cdnet.stpi.org.tw/techroom/analysis/pat_A066.htm.

[32] http://msdn2.microsoft.com/en-us/library/ms174879.aspx

[33] http://msdn2.microsoft.com/en-us/library/ms175312.aspx

[34] http://msdn2.microsoft.com/en-us/library/ms174941.aspx

[35] http://msdn2.microsoft.com/en-us/library/ms174828.aspx

[36] http://msdn2.microsoft.com/en-us/library/ms174806.aspx

[37] http://msdn2.microsoft.com/en-us/library/ms174916.aspx

[38] http://www.ceci.org.tw/book/53/ch53_1.htm

Appendix A

Decision Tree

Whichever variable is used to generate child-nodes, there is situation that child-nodes distribution is not clean in decision tree. Hence while we are hoping to find out which variable could have best result, it is necessary to determine which variable after branch has the highest overall data purity. Here the problems listed as follows [25].

1. The number of branches generated by each variable is different. It must be able to provide the overall weighting of purity in order to evaluate which variable could allow the highest purity (in other words, it is possible to compare the purity changes between binary branch and multiple-element branch).
2. The sum of purities of child-nodes must be able to be compared with the purity of the parent-node to determine whether the branch should be kept. This is because if the child-node could not have significantly higher purity than the originating branch, the leaf branch will be meaningless.

We call the answers to these problems as split criteria. Split criteria are the core of decision tree algorithm. Through these split criteria, it is possible to find out better variable. After generation of the first branching node, the decision tree repeats the steps of growth. Each generated child-node is treated as a root node to continue with the generation of the next best branch variable and generate new branch. In this way, decision tree will continue to grow until there is no way to generate branch with higher purity.

If a decision tree is generated based on the procedure mentioned above, the predicted result of the training set is guaranteed to be extremely good (as the entropy is reduced continuously). However, when new data are encountered, the prediction result might not be as accurate as the training data. This is because blind branch could

be a coincident in finding more data and result in inaccuracy of the prediction model. This is also called “over-fitting” phenomena. To avoid over-fitting problem, the whole process of the learning rule must be modified slightly as follows.

1. The algorithm will randomly divide the whole data into training set and validation set.
2. Use the training set to generate the first branching point based on the split criteria.
3. Use the validation set to test whether the first branching point is better branch. If the rule could resurface, continue to perform the subsequent branch. If the rule does not resurface, abandon the variable and restart to filter out better branching variable from the rest of the variables.
4. Repeat the above procedures until no more leaf node with higher purity can be generated. Such child-node at the end is called “Leaf Node”.
5. Use pruning technique to prune off the redundant or invalid branching points.

In summary, the building up of decision tree is the result of two strengths pulling each other. The first is the growing strength for the generation of branches, while the other strength is for the depression of the growth of decision tree through testing and pruning, or removing invalid rule. Looking from the time of depression, depression of growth can be divided into two types: one is synchronous suppression or test and pruning through validation data at the same time of training model; the other is post suppression, or remove the redundant rule from the growing decision tree.

Pruning method of synchronous depression validation set is to divide the training set and validation set into two parts. When generating new branch from training set data, use the validation set immediately to test whether the split criteria resurface. If the answer is no, it will be treated as over-fitting and is pruned off. If this rule can resurface in the validation set, the branch will be kept and continue to branch downward. Post depression is mainly to prune off invalid rule through decision tree pruning techniques. Currently it is possible to set the Microsoft decision tree through

Minimum_Support parameter.

The decision tree algorithm is a quite mature technology at present. Four techniques often used are C5, CART, CHAID and QUEST, and the most famous decision tree algorithm is ID3 (Iterative Dichotomiser 3, previous version of C4.5), which uses Information Gains as the split criteria. However, the number of decision tree rules generated using the split criteria is on the high side and easily results in the effect of over-fitting. To correct such deviation, the calculation formula of "Gain Ratio" is redefined to replace the original split criteria.

The purpose of split criteria is to examine the overall purity variance of parent-node and child-nodes when using certain variable as the branch variable so that variable with higher purity will become the effective variable. Based on this purpose, the equation of gain ratio becomes:

$$Gain\ Ratio=(Entropy_{before}-Entropy_{after})/Split\ Gains$$

What the whole equation wants to express the fact that the gain ratio equals to the difference in entropies between the parent-node and child-node divided by the modifier of split gains. Where entropy is the randomness, it indicates the distribution state of the object. The higher the entropy is, the more irregular the distribution is. Hence, the goal of the decision tree algorithm is the desire of reducing the resultant entropy of data classification. The equation is shown as follows.

$Entropy_{parent\text{-}node} = \sum X \log_2 X$ ← meaning the ith ratio n_i / n of the predicted variable.

$$Entropy_{child\text{-}node} = \Sigma\ (nth_{child\text{-}node} - nth_{parent\text{-}node}) * Entropy_{child\text{-}node}$$

The gain ratio is:

$$Gains\ Ratio = (Entropy_{parent\text{-}node} - Entropy_{child\text{-}node})/Split\ Gains$$

In comparison, the gain ratio when splitting is higher and more suitable to be used as split variable.

However, the system will recalculate once based on the validation set of automatic random sampling division. If the gain ratio of the validation set calculated is within the range of tolerance of the training set values, it will be confirmed as the first level split variable. At this time, the distribution of the child-node case will be used as the predicted probability. Subsequently, each child-node will be treated as a new root node for the continuation of split criteria calculation so that the decision tree algorithm will continue to grow.

Appendix B

Neural Network

Back-propagation network is formed by multi-level neuron structure. The outmost layer is the input layer that receives input variables. The neuron in the middle is called hidden layer, shown in Figure B.1.

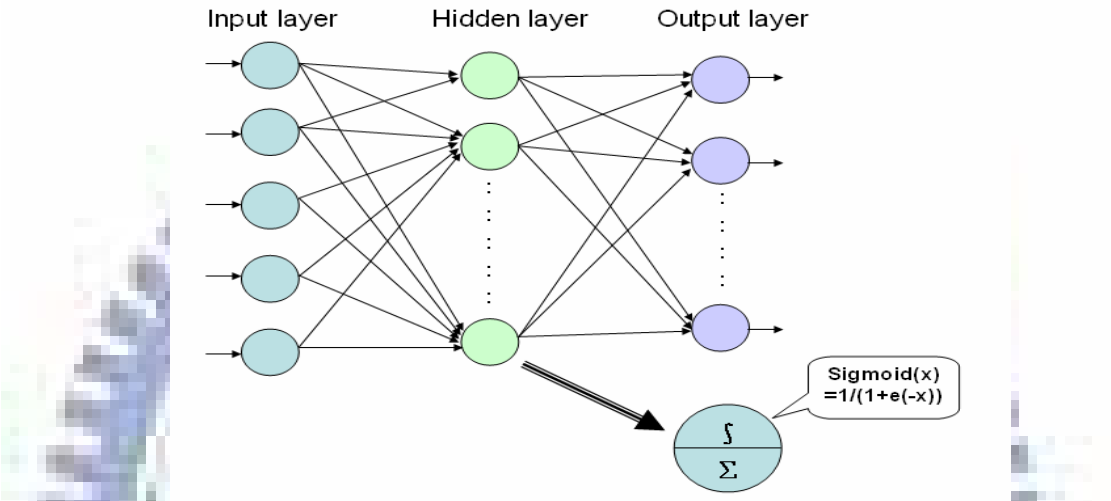


Figure B.1. Flow chart of neural network.

The learning process of back-propagation network can be divided into the feed-forward transmission and feed-backward transmission. For feed-forward transmission, the input signal enters the hidden layer from input layer neuron and transmits to output layer. Each neuron is only responsible for transmitting signal to neuron downstream to generate final signal (prediction result). The starting weight of the back-propagation network is assigned through random process. Therefore, the output result generated from the beginning will have drastic difference with the actual result. When there is difference between the output value and the actual result, the deviation signal will return back from the original feed-forward transmitting direction and modify the weight of each neural connection synchronously so that the deviation can be minimal. This whole process is what we called “Learning” [25].

The core of the learning of neural network is in how to automatically and efficiently adjust the magnitude of the weights. Currently, the method used most often

is the gradient steepest descent method, where the deviation is expressed by derivable function. We use the gradient of the differential slope to generate the modifier of the weights, when the deviation is positive (output value is greater than actual value), the neural weight is reduced. If the situation is reverse, the neural weight is increased. The process is repeated to lower the deviation. Based on the theory, deviation will approach zero if the learning time is infinite. Actually, the longer the training the more probable the neural network will memorize the data content of the memory training network. In this way, though the deviation of training set data can approach zero, it will result in the lowering prediction capability when inputting new data. Therefore, the neural network needs a validation set sample to validate the correctness of weight modifier. Most algorithms will sample validation set from training data through automatic random sampling method. During the training process, the deviation of the training set theoretically will approach zero indefinitely. However, the deviation of validation set will be reduced in the beginning. As the over-fitting phenomena occur, the deviation of validation set will start to increase from lowest point backward. Then, the weight combination of the validation set deviation at the lowest point is better training result.

There is a biggest weakness in the gradient steepest descent method. We are looking for the minimum value of deviation function through the correction weight in the gradient direction. However, as the deviation function itself is a nonlinear function, it will approach the part area with lowest deviation directly if the randomly given weight at the beginning falls in the incorrect places, shown in Figure B.2. It will be trapped here and could not come out. But the “local optimal solution” may not be the same as the “absolute optimal solution”. Therefore, it is possible to generate a weight random number mode in the beginning and try several times when undergoing neural network process to avoid the problem of local optimal solution.

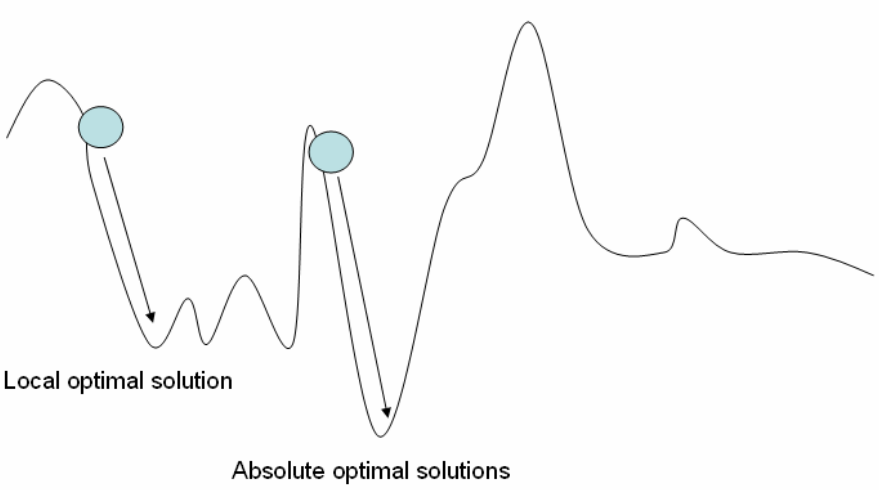


Figure B.2. The local optimal solution of the gradient steepest descent method.

In addition, neural network is to generate learning result through the weight modification method. The algorithm itself does not have the function of variable filtering. In other words, all variables will calculate the weights belong to themselves. Hence, if input invalid or unsteady variables, it easily results in poor learning result of the model or too slow convergence situation. Besides, when filtering variables it is necessary to test the relevancy of variables beforehand, or it will easily result in the problem of co-linearity. In other words, when two input variables are showing high degree of positive relevancy, the function will have unlimited sets of solutions render the neural network unable to converge.