https://doi.org/10.5392/IJoC.2017.13.2.057

A Decision Tree Approach for Identifying Defective Products in the Manufacturing Process

Sungsu Choi
R&D Center
YURA Co., Ltd., Seongnam, Gyeunggi, South Korea

Lkhagvadorj Battulga, Aziz Nasridinov, Kwan-Hee Yoo
Department of Computer Science
Chungbuk National University, Chungbuk 28644, South Korea

ABSTRACT

Recently, due to the significance of Industry 4.0, the manufacturing industry is developing globally. Conventionally, the manufacturing industry generates a large volume of data that is often related to process, line and products. In this paper, we analyzed causes of defective products in the manufacturing process using the decision tree technique, that is a well-known technique used in data mining. We used data collected from the domestic manufacturing industry that includes Manufacturing Execution System (MES), Point of Production (POP), equipment data accumulated directly in equipment, in-process/external air-conditioning sensors and static electricity. We propose to implement a model using C4.5 decision tree algorithm. Specifically, the proposed decision tree model is modeled based on components of a specific part. We propose to identify the state of products, where the defect occurred and compare it with the generated decision tree model to determine the cause of the defect.

Key words*: Manufacturing Data, Decision Tree, C4.5.*

1. INTRODUCTION

Recently, due to the importance of Industry 4.0, the manufacturing industry is developing all over the world. Specifically, governments and companies use various ways to increase the profit, such as decreasing the defective products and improve overall efficiency of manufacturing process. In the past, worker-oriented process management system was used, where the quality of the product was determined by the worker [7]. However, putting the worker in all processes is an obstacle to the process control, as it may lead to decrease in productivity and the deterioration of quality. Thus, current systems adopt the data-driven approach, where various data are being collected about a process, and automation of the process is performed to improve productivity and quality.

Conventionally, manufacturing industry generates a huge amount of data that is often related to process, line and products. Data mining is a process that extracts useful knowledge and information by modeling and finding patterns, rules, etc., based on data obtained from manufacturing industry [1]. We can simply explain the data mining through the following example. In a US supermarket, a survey of customers' shopping trends showed that newly-married couples with young children bought diapers and beer on Friday afternoon. The reason for this is that couples with young children often spend their weekend indoor, which shows a tendency to buy beer and diapers for young children at the same time. As a result, the large supermarket displayed diapers and beer at the same place, which was well received by customers who felt the convenience of reducing shopping time. Likewise, in the manufacturing industry, not only formal data generated in the process but also unprocessed forms of unstructured data are collected, and data mining techniques are needed to derive patterns, trends and meaningful results based on the collected data.

In this paper, we analyze the causes of defective products in the manufacturing process using the decision tree technique, which is a well-known technique in data mining. We use the data obtained from the domestic manufacturing industry that includes Manufacturing Execution System (MES), Point of Production (POP), equipment data accumulated directly in equipment, in-process/external air-conditioning sensors and static electricity. More specifically, we make the following contributions in this paper:

- We propose to assemble a training data set using Critical to Quality (CTQ) tables and handwritten data, key factors, in-process external air-conditioning sensors, meteorological data, static electricity, vibration data and other resources.

* *Corresponding author, Email: khyoo@chungbuk.ac.kr*
Manuscript received May. 02, 2017; revised Jun. 19, 2017; accepted Jun. 19, 2017

- We propose to extract key factors that affect the manufacturing process through pre-processing step.
- We propose to implement a model using C4.5 decision tree algorithm. In this study, a decision tree model is modeled based on the components of a specific part.
- We propose to identify the state of the product where the defect occurred and compare it with the generated decision model to determine the cause of the defect.

The rest of the paper is organized as follows. In Section 2, we discuss the related study. In Section 3, we describe the proposed method in details. In Section 4, we present results of experiments. In Section 5, we conclude the paper and highlight the future work.

2. RELATED STUDY

In this section, we first explain decision tree algorithm and its variants, and then discuss methods that have been done towards the studied field.

2.1 Decision Tree

Decision trees are a typical method of data mining analysis. [5] defined the process of finding meaningful new patterns or trends from a large amount of data by using pattern recognition techniques as well as statistical and mathematical techniques. Decision trees are classified into several subgroups by charting decision rules. Since the analysis process is represented by a tree structure, it is necessary to classify the group of interest into several subgroups. Therefore, it can be used for discriminant analysis and regression analysis as decision tree structure is easy to understand comparing to methods such as neural networks.

The structure of the decision tree is shown in Fig. 1. It has the following components. There is the root node at the top of the tree. When a node is differentiated into lower nodes, the nodes located above of a certain node are called parent nodes, and nodes located below of a node are called child nodes. The final node where the node is no longer differentiated is called a leaf node.

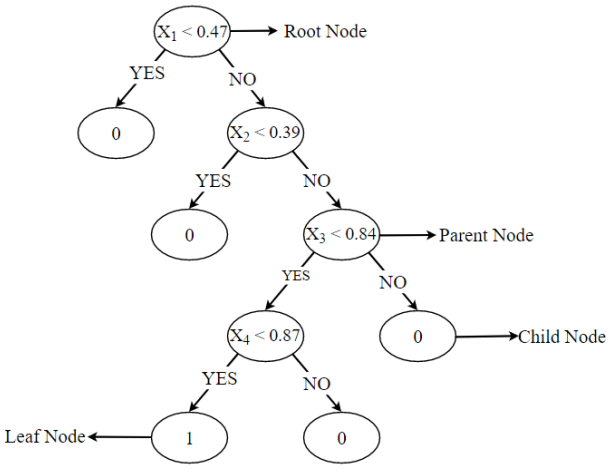

Fig. 1. Structure of Decision Tree

Decision trees can be divided into classification trees that differentiate nodes based on nominal target variables and regression trees that differentiate nodes based on continuous target variables. Decision trees used in this study are classified trees whose nodes are differentiated based on the nominal target variable. There are various kinds of decision tree algorithms, ID3, C4.5, CART, and CHAID algorithms are the most commonly used ones. In this paper, we refer to the algorithm of C4.5 which overcomes the shortcomings of ID3.

The ID3 algorithm is the representative decision trees algorithm. The ID3 algorithm was proposed by J. Ross Quinlan in the late 1986, and various decision tree based classification algorithms (Ex. C4.5, CART, CHAID, etc.) were developed afterwards that were basically based on the idea of the ID3 algorithm. The ID3 algorithm classifies data by entropy and information gain. Entropy refers to the congestion of a given data set. Entropy is high when records of a given data set are mixed with many different classes, and entropy is low when many records of the same kind are involved. The value of entropy has a value between 0 and 1, the value of the highest congestion state is 1, and the state of only one class is 0. In the decision tree classification algorithm, the data is classified into tree shapes by finding the measurement conditions so that the entropy is high and the state is low.

The C4.5 is an algorithm developed by J. Ross Quinlan in 1993, which improves the previous ID3. In case of ID3 algorithm, numerical attribute cannot be used, and when attribute category value is large, there is a disadvantage in that the number of values of child nodes becomes very large. C4.5 is an algorithm that complements these disadvantages and adds new functionalities [6]. Additional and complementary points include the first numeric property. Due to these advantages of ID3 algorithm, we used it in our implementations.

2.2 Decision Tree in Manufacturing

There have been a number of approaches proposed to improve efficiency of manufacturing process using data mining techniques. In this subsection, we briefly describe them and discuss main differences.

Wang [2] discussed the nature and implications of data mining techniques and their implementations on product design and manufacturing. The authors show that using traditional data analysis approach is not suitable for current manufacturing enterprises. By using the data mining technique, the authors create an intelligent tool for extracting useful information automatically which enables the engineers and the managers to understand the complex manufacturing data easily. The paper concludes that the in modern society, only using sheer technological power in manufacturing is not suitable, as new innovative technologies, such as, Business Intelligence, data mining and knowledge managements can give sustainable competitive advantage for manufacturing companies.

Kusiak [3] proposed a framework for decision-making approach based on the knowledge provided by different data mining techniques. The authors proposed a novel concept of decision atlas, maps and tables that creates the user-friendly, transparent and convincing decision for manufacturing industries. The framework makes a decision based on the

knowledge provided by various kinds of data mining algorithms.

Rokach and Maimon [4] proposed a data mining for improving the quality of manufacturing process based on a feature set decomposition approach. The authors have found out the current classification methods are not suitable for the manufacturing industrial datasets due the nature of characteristics associated with quality. In order to examine it, they introduced a new algorithm called BOW (Breadth-Oblivious-Wrapper). The algorithm performs a breath first search while using a splitting method called F-measure criterion for multiple oblivious trees. It was tested over three real-life datasets and the results show that their framework tends to outperform other well-known methods in both accuracy and F-measure.

The main difference comparing to the aforementioned approaches is that we use data mining for determining the cause of defective products in manufacturing industry.

3. DECISION TREE FOR PROCESS MANAGEMENT

In this section, we design a decision tree for process control and introduce actual data set composition. In the current manufacturing industry, there is a lack of understanding of the relationship between collected data and accumulation of manufacturing data. In addition, it is still difficult to identify the cause of the defective products due to the experience of the field manager. In this study, the analysis was conducted based on the data collected in the actual process, and the preprocessing was required because the number of variables collected in the facility was too large. The target product is an electronic equipment product, which is a core part of automobile, and analyzed by each component of PCB board for each process. Fig. 2 demonstrates a conceptual design of the proposed method.

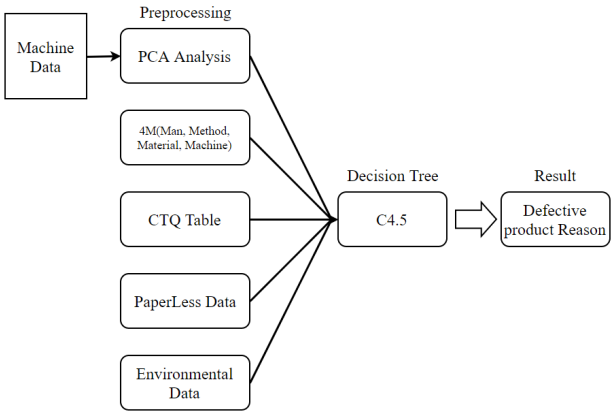

Fig. 2. A conceptual design of the proposed method.

3.1 Dataset

Since the inspection data measured from the three inspection processes are collected and the attributes of the inspection data are too many and varied, it is difficult to construct all the attributes with data sets. 4M data is collected through the existing MES, POP, and ERP and the CTQ table used the values of the allowable range. This range is defined by the manufacturer based on the attributes that affect the quality of each product. PaperLess refers to the data collected by constructing a computerized system for a process management. PaperLess is measured twice a day (day and night), and the measured value is displayed along with the occupant's number when entered. Environmental data is collected in the unit of temperature/humidity in the process, temperature/humidity sensor inside the warehouse, and temperature/humidity provided by the meteorological office. Table 1 demonstrates time cycles where data is collected.

Table 1. Data Cycle Collected on Site

Division	Collection period
Equipment data (3 target processes)	1 hours
LEGACY(POP,MES,ERP)	1 hours
PaperLess	Twice a day
Sensor (process and inside / outside temperature / humidity)	Real-time
Meteorological Information	1 hours

Table 2 shows the number of attributes to be used in the decision tree. It is obvious that the data set is different based on the facility data, LEGACY data, PaperLess data, sensor data, and weather data of each facility.

Table 2. Number of attributes per facility

Fair	Division	Number of properties
A process	Facility Data	8
	LEGACY(POP,MES,ERP)	1
	PaperLess	1
	Sensor (process and inside / outside temperature / humidity)	4
	Meteorological Information	2
	Total Attributes	16
B process	Facility Data	8
	LEGACY(POP,MES,ERP)	1
	PaperLess	1
	Sensor (process and inside / outside temperature / humidity)	4
	Meteorological Information	2
	Total Attributes	16
C process	Facility Data	5
	LEGACY(POP,MES,ERP)	1
	PaperLess	1
	Sensor (process and inside / outside temperature / humidity)	4
	Meteorological Information	2

Table 3 demonstrates the dataset consisting of the measured values of the five products of Process A, Component 10. There are a total of 16 properties. Temperature and humidity inside the factory, temperature / humidity inside the warehouse, and temperature / humidity provided by the Korea Meteorological Agency. This data was configured according to the time when the product was made. The attribute H has the values of 'Good', 'E.ins', 'E.exe', 'E.Bri', 'E.Pos', 'W.Exc', and 'W.Hei'.

Table 3. Process A 10th Component 5 data sets

Property	A	B	C	D	E	F	G	H
Product1	112.816	177.223	95.487	0.031	-0.069	2.48E+08	1400721	GOOD
Product2	112.14	174.304	96.504	0.02	0.006	2.47E+08	1415647	GOOD
Product3	113.349	176.884	96.122	0.003	-0.072	2.49E+08	1410038	GOOD
Product4	116.223	181.189	96.217	0.037	-0.001	2.56E+08	1411436	GOOD
Product5	121.613	181.237	100.653	0.041	-0.078	2.68E+08	1476501	GOOD
Property	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**
Product1	159.2	56.4	57.28	22.65	55.57	24.14	90	16.1
Product2	159.2	56.4	57.28	22.65	55.57	24.14	90	16.1
Product3	159.2	56.4	57.28	22.65	55.57	24.14	90	16.1
Product4	159.2	56.4	57.28	22.65	55.57	24.14	90	16.1
Product5	159.2	56.4	57.28	22.65	55.57	24.14	90	16.1

Table 4 is the dataset consisting of the measured values of the five products of Process B, Component 10. There are a total of 16 properties and the attribute G is the target variables' Good ',' E.ins', 'E.exe', 'E.Bri', 'E.Pos',' W.Exc ',' W.Hei'.

Table 4. Process B 10th Component 5 data sets

Property	A	B	C	D	E	F	G	H
Product1	135.369	196.992	103.077	0.02	-0.018	1.28E+08	649154	GOOD
Product2	134.852	206.07	98.16	0.014	0.019	1.27E+08	618188	GOOD
Product3	128.427	181.017	106.422	0.029	-0.078	1.21E+08	670217	GOOD
Product4	130.345	180.675	108.215	0.002	-0.088	1.23E+08	681513	GOOD
Product5	133.486	186.214	107.526	0.021	-0.064	1.26E+08	677173	GOOD
Property	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**
Product1	77.2	27.9	45.51	23.74	49.63	24.71	82	11.7
Product2	77.2	27.9	45.51	23.74	49.63	24.71	82	11.7
Product3	77.2	27.9	45.51	23.74	49.63	24.71	82	11.7
Product4	77.2	27.9	45.51	23.74	49.63	24.71	82	11.7
Product5	77.2	27.9	45.51	23.74	49.63	24.71	82	11.7

Table 5 shows the data set consisting of the measured values of the five products of Process C, Component 10. There are 13 properties in total, and attribute G has the value of 'OK' and 'NG' as target variables and becomes the leaf node which is the last node in the decision tree.

Table 5. C Process 10 Component 5 data sets

Property	A	B	C	D	E	F	G
Product1	100	0	10	10	0	103.74	OK
Product2	100	0	10	10	0	103.64	OK
Product3	100	0	10	10	0	103.39	OK
Product4	100	0	10	10	0	104.39	OK
Product5	100	0	10	10	0	102.84	OK
Property	H	I	J	K	L	M	H
Product1	35.95	22.92	51.27	24.27	69	12	35.95
Product2	35.95	22.92	51.27	24.27	69	12	35.95
Product3	36.79	22.97	50.72	23.92	65	12.9	36.79
Product4	36.79	22.97	50.72	23.92	65	12.9	36.79
Product5	36.79	22.97	50.72	23.92	65	12.9	36.79

The following is the process of calculating the baseline statistic based on the measurements from the dataset. Table 6 summarize the basic statistics of Process A, Component 10. It is important to observe the characteristics of the data by calculating the minimum, maximum, average, and standard deviation for each data before analysis. The data with the highest standard deviation are scattered with the F property, and the data with the smallest standard deviation have I and J properties.

Table 6. Component A 10th Component Basic Statistics

Statistics	A	B	C	D	E	F	G
Minimum value	2.638	46.672	8.063	-0.373	-0.411	5804332	118282
Maximum value	229.	256.	133.999	0.057	0.056	50473350	1965673
Medium	125.8	182.10	102.819	0.017	-0.033	276847032	1508324
Standard Deviation	20	19.726	10.697	0.04	0.058	43360271	156875

Statistics	I	J	K	L	M	N	O	P
Minimum value	159.2	56.4	43.5	21.42	50.39	23.24	70	-3.3
Maximum value	159.2	56.4	59.14	25.06	55.57	24.9	94	18.8
Medium	159.2	56.4	56.76	22.576	55.11	24.166	91.196	14.988
Standard Deviation	0	0	1.826	0.285	0.591	0.113	2.675	1.71

Table 7 summarizes the basic statistics of Process B, Component 10. The property with the highest standard deviation is the F property and the property with the lowest standard deviation is the J property.

Table 7. B Process 10 Component Basic statistics

Statistics	A	B	C	D	E	F	G
Minimum value	0.272	40.509	1.008	-0.193	-0.459	257112	6347
Maximum value	179.2	234.042	140.402	0.493	0.047	176995100	924077
Medium	122.6	175.364	102.472	-0.012	-0.015	120609686	671789
Standard Deviation	25.666	26.933	18.978	0.045	0.079	25095260	124793

Statistics	I	J	K	L	M	N	O	P
Minimum value	74.5	27.9	42.13	22.14	49.52	24.04	49	9.8
Maximum value	77.2	27.9	58.94	25.38	56.15	25.79	92	21.3
Medium	74.754	27.9	55.778	22.709	54.727	24.225	89.859	15.083
Standard Deviation	0.789	0	3.748	0.414	1.777	0.235	5.639	1.358

Table 8 summarizes the basic statistics of Process C, Component 10. The attributes with the highest standard deviation are M properties, and the lowest attributes are 0, A, B, C, D, and E.

Table 8. C Process 10 Component Basic statistics

Statistics	A	B	C	D	E	F
Minimum Value	100	0	10	10	0	102.64
Maximum Value	100	0	10	10	0	116.08
Medium	100	0	10	10	0	103.6
Standard Deviation	0	0	0	0	0	0.946

Statistics	I	J	K	L	M	N
Minimum Value	35.26	22.78	50.39	23.92	57	0.1
Maximum Value	49.84	25.02	52.56	24.27	90	12.9
Medium	36.377	22.948	50.953	24.084	66.159	12.356
Standard Deviation	0.975	0.131	0.032	0.167	3.789	0.883

3.2 Data Analysis

The algorithm C4.5 of Decision Tree uses the method of dividing by entropy index and information gain.

$$Entropy(S) = \sum_{i=1}^{m} p_i log_2(p_i) \quad (1)$$

$$p_i = \frac{freq(C_i, S)}{|S|}$$

The entropy value is calculated by Equation (1). This equation is for calculating the entropy value for a given data set *S*. The entropy value is calculated by applying logarithm to the content ratio of each class value and adding all the values multiplied by the weight value again. Since a negative (-) value appears through the application of the log_2 function, a value between 0 and 1 can be obtained by pasting the whole formula value with (-). *S* is a set of given data and C is a set of class values. $freq(C_i, S)$ is the number of records belonging to Class *C* in a given set of data, and |*S*| is the number of data in a given set of data.

Information Gain distinguishes data well in selecting an attribute. For example, assuming that the English score is higher than the physical score in discriminating the SAT, the English score attribute is higher than the physical score attribute. To calculate the information gain, we first need to understand the concept of entropy and the calculation method.

$$Gain(S) = I(s_1, s_2, \ldots, s_m) - E(propertyA) \quad (2)$$

Equation (2) is a formula for obtaining an information gain index. Where $I(s_1, s_2, ..., s_m)$ is the entropy of the original ancestor. That is, the entropy of the upper node is subtracted from the entropy of the lower node. If *E (A)* is selected as the attribute *A*, it is calculated by dividing the small *m* nodes into a lower value. Then, the entropy is calculated by calculating the entropy of each lower node, and then the entropy is averaged using the number of records belonging to the node as the weight. Equation (2) is an expression for calculating the amount of information gain when the attribute *A* is selected. The entropy of the original node is obtained, the overall entropy is obtained by dividing the entropy of the original node into *m* sub nodes after the attribute *A* is selected, *B* means *AB*. In other words, the larger the value, the greater the information gain, and the property *A* means better discrimination [11].

In this study, the experiments were conducted through the WEKA workbench. Weka was developed at the University of Waikato in New Zealand and stands for Waikato Environment for Knowledge Analysis. WEKA was developed in Java and is distributed under the GNU general public license. It can be operated on almost any platform, and is well portable with Java and is often used to derive results from data mining on the web [8]. Based on the data set of the A process, the decision model was created through the C4.5 algorithm.

Fig. 3 shows the model of 10th Component of A process through decision tree. The root node starts with attribute B, and the numerical value is 189.554. If the value of attribute B is lower than or equal to 189.554, the child node is classified by reference value 96.122 again by the value of attribute C, and classified as Good leaf node when the value of attribute C is higher than 96.122. K is divided into 'E.Ins' and 'Good', which are the last leaf node by the reference value 56.76. If the value of attribute B is larger than the reference value 189.554, it is classified as the value of attribute A. If it is larger than the reference value 169.901 of attribute A, it is classified as 'E.Exe'. Finally, if it is less than or equal to the reference value 24.09 of the property L, it is classified as 'W.Hei', and if it is large, the model classified as 'W.Exc' can be identified.

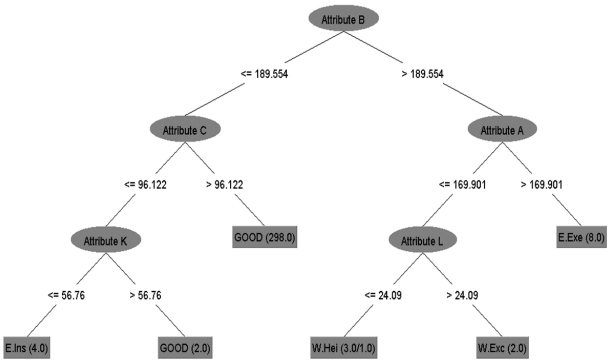

Fig. 3. A Process 10 Component Decision Tree Model Creation

The result of generating decision tree of Component 10 of B process is slightly different from that of A process by the number of tree structure and leaf nodes. First, the root node starts with attribute B and is classified as attribute C if it is less than or equal to the reference value 216.008, and classified as attribute A when it is large. In case of property C, it is classified as 'E.ins' when it is less than or equal to the reference value 38.99, and 'Good' when it is greater than the reference value 38.99. When the property A is less than or equal to the reference value 164.537, 'W.Hei' was created.

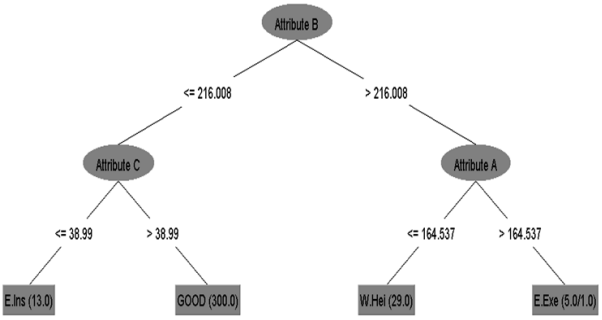

Fig. 4. shows the model generated by decision tree of Component 10 of B process

Fig. 5 shows the result of model generation through decision tree of Component 10 of C process. The root node starts from attribute G and is classified as OK and attribute I by reference value 104.34. In attribute I, which is a child node, it can be seen that it is classified as OK and NG by reference value 36.79.

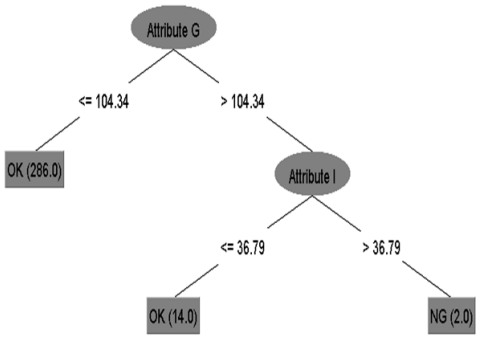

Fig. 5. Component C 10th Component Decision Tree Model Creation

4. EXPERIMENTAL RESULT

4.1 System implementation

In this section, based on the decision tree model generated in Section 3, visualization is performed on the web and the causes of defective products are derived. If we simply create a model with only training data, we will not know the cause of the faulty product. Therefore, in this study, we derive the cause of the defect by comparing it with the tree model generated when the defective product occurs.

Fig. 6 shows the system structure of the cause of defective product for final product in this study. There are a total of 5 databases, each of which has facility data, 4M Data, CTQ Table, PaperLess, environmental data, etc. Silver data has a huge property of collected data. Based on this data, when a bad product occurs in the server, it requests the data and the database transfers the data for configuring the data set to be used in the decision tree in the server. Once the dataset is constructed, the C4.5 algorithm of the decision tree is analyzed through the WEKA library and the analysis results are received. Since the analysis result is simply a result of a string form, a parsing method like (a) and (b) in Fig. 7 is used.

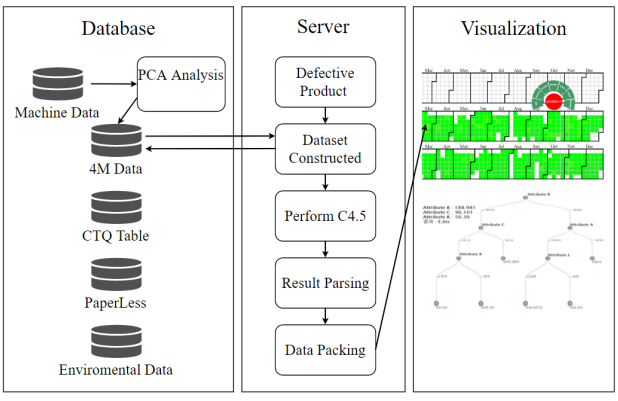

Fig. 6. The proposed system configuration diagram

(A) is a method stores the attributes of root node and then generate child node and outputs it when it becomes leaf node. (B) is a method to output connectivity between parent node and child node. For both methods, the output result is packed into the data to be used on the web and visualized.

```
public static void writeNode(ArrayList<TreeClass> a, Node node, String parent) throws Exception {
    int i;
    String tag;
    boolean leaf;

    // leaf?
    leaf = (node.getChild(0) == null);
    if (leaf) {
        tag = "leaf";
    } else {
        tag = "branch";
    }

    // the node itself
    TreeClass t = new TreeClass();
    t.setName(node.getLabel());
    t.setParent(parent);
    a.add(t);
    // the node's children, if any
    if (!leaf) {
        for (i = 0; (node.getChild(i) != null); i++) {
            writeEdge(a, node.getChild(i), node.getLabel());
        }
    }

}
```

(A) Result of analysis Parsing - writeNode Method

```
public static void writeEdge(ArrayList<TreeClass> a, Edge edge, String label) throws Exception {
    if (edge.getLabel().length() > 0) {
        TreeClass t = new TreeClass();
        t.setName(edge.getLabel());
        t.setParent(label);
        a.add(t);
        writeNode(a, edge.getTarget(), edge.getLabel());
        // writer.write("</branch>"); writer.newLine();
    } else {
        writeNode(a, edge.getTarget(), edge.getLabel());
    }
}
```

(B) Result of analysis Parsing - writeEdge Method

Fig. 7. Analysis result using parsing Method

4.2 Implementation Result

Fig. 8 shows the decision tree model of Component 10 of Component A and the cause of product failure. In case of bad product, the measured value of Attribute B is 188.981, so it is classified as child node Attribute C, and the measurement value of Attribute C is 96.101, which is smaller than the reference value of 96.122. Therefore, in order to produce good products, it can be considered that the measurement value of Attribute B is lower than 188.554, the measurement value of Attribute C is higher than 96.122, and the measurement value of Attribute K is higher than 56.76.

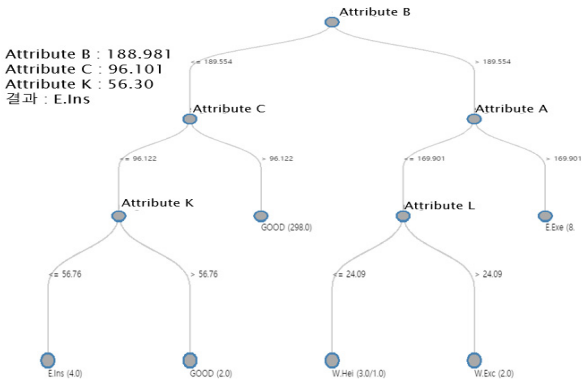

Fig. 8. Cause of component defect 10 of A process

Fig. 9 shows the decision tree model of Component B, Component 10, and the cause of the product failure. In case of

bad products, the measurement value of Attribute B is 216.235, so it is higher than the standard value of 216.008, and the measurement value of Attribute A is 164.998, but the reference value is 164.537, so the size is larger. Thus, classification result is 'E.Exe'. The criterion for producing good products is that the measured value of Attribute B should be less than or equal to the reference value of 216.006 and the measured value of Attribute C must be higher than the reference value of 38.99 to produce a product that is 'Good'.

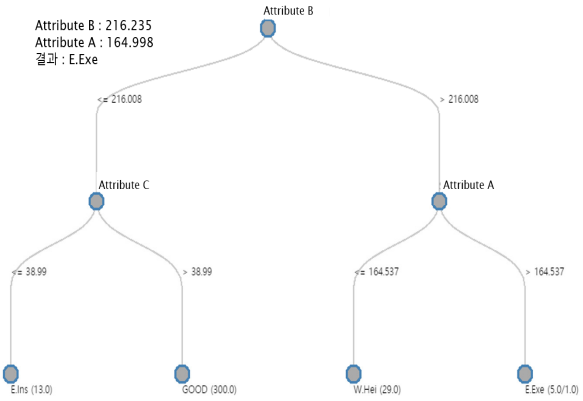

Fig. 9. Cause of component defect 10 of B process

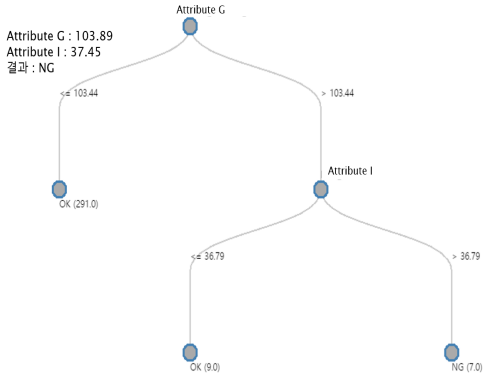

Fig. 10. Cause of component defect 10 of C process

Fig. 10 shows the decision tree model of Component 10 of C process to generate the cause of defective product. In case of bad products, the measurement value of Attribute G is 103.89, which is higher than the reference value of 103.44, and the measured value of Attribute I is 37.45, but the reference value is 36.79, so the size is larger. In order to produce good products, the measurement value of Attribute G should be less than or equal to the reference value of 103.44, and if it is big, the measurement value of Attribute I should be 36.79 smaller than or equal to the reference value.

4.3 Performance evaluation

In this subsection, we describe the performance evaluation. As test dataset, we used the training data described in Section 3. Even if a good decision tree model is generated, it cannot be a good decision tree model if the classification accuracy with real data is not appropriate. Therefore, in this section, the number of good and bad products is evaluated by 50, 100, and 300 pieces.

Table 9 summarizes classification accuracy according to the number of data. When the number of data is 50, and the ratio of good and bad is about 8:2, classification accuracy is about 95%. It can be said that about 2.5 out of 50 cases cannot be categorized. When the number of data is 100, and the ratio of good and bad is about 8:2, classification accuracy is about 95%. Finally, if the number of data is 300 and the number of good products is 263, the classification accuracy is 96%, which is the highest value. For the training data of Process A, a decision tree model was created based on 890 good data and 110 bad data out of a total of 1000 data.

Table 9. Performance evaluation of decision tree model of A process

Number of data	Number of Products	Number of bad Products	Accuracy
50	39	11	95.24%
100	81	19	95.32%
300	263	37	96.01%

Table 10 summarizes the classification accuracy according to the number of data. As with the A process, we evaluated the performance of decision trees based on 50, 100, and 300 test data. In the case of 50 cases, the number of good products is 35, and the number of defective cases is 15, and the classification accuracy is about 91%. This is the case that about 4.5 out of 50 products are not classified. Likewise, about 100 cases were classified as 91%, and finally 300 cases were classified as 92%. In the case of the B process, a decision tree model was created based on 934 good data and 66 bad data for 1000 training data.

Table 10. Performance evaluation of decision tree model of B process

Number of data	Number of Products	Number of bad Products	Accuracy
50	35	15	91.21%
100	72	28	91.58%
300	278	22	92.59%

Table 11 shows the performance evaluation of the decision tree model of the C process. The C process used 1230 good data and 270 bad data, and a decision tree model was created through a total of 1500 data. As a result with 50 data, 89% classification accuracy was obtained, which is the case that 5.5 out of 50 data were not classified. 100 cases showed an accuracy of 88% and 300 cases showed only 90% accuracy.

Table 11. Performance evaluation of decision tree model of C process

Number of data	Number of Products	Number of bad Products	Accuracy
50	29	21	89.56%
100	59	41	88.65%
300	235	65	90.15%

We can confirm the reason why the ratio of training data for each process does not exist is because the number of defective products is inevitably small due to the nature of the manufacturing company and the proportion of good products is higher than that of defective products. As a result of the performance evaluation, the classification accuracy tended to be higher as the number of test data increased. Process A showed higher classification accuracy than other processes, and process C had lower classification accuracy than other processes.

5. CONCLUSION

In this paper, based on the data collected in the process, we have proposed to investigate the causes of defective products using C4.5 algorithm of decision trees. The existing methods to increase the productivity in the manufacturing process did not show a great effect due to lack of work experience and the single process oriented analysis. In addition, most of the existing researches have been conducted to compare the results of the analysis according to the size of the data. In this study, we selected the electronic equipment products to be used in the automobile industry. The data set is configured to be applied in the actual process through facility data, LEGACY data, environmental data, and handwritten data provided by the manufacturer. The cause of the defective product was derived. Through the result, it was confirmed that the critical management items of each component of each process were identified and the classification accuracy of the generated decision tree was calculated to increase the reliability of the decision tree. Due to the nature of the manufacturing industry, the number of good products is far greater than the number of defective products. In this study, the ratio of defective products to good products in each process is not appropriate, so that the reliability of C process is lower than that of other processes. In the future, studies to increase the reliability of decision trees should be continued even when the number of defective products is small, and further improvements in defective products can be reflected through field application.

ACKNOWLEDGEMENT

This research was supported by the Knowledge Services Industry Core Technology Development Program (10051028,

Development of Predictive Manufacturing System using Data Analysis of 4M Data in Small and Medium Enterprises) funded By the Ministry of Trade, Industry & Energy (MI, Korea), and also by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2017-2013-0-00881) supervised by the IITP(Institute for Information & Communication Technology Promotion).

REFERENCES

[1] Ian H. Witten and Eibe Frank, "Data Mining: Practical machine learning tools and techniques," Morgan Kaufmann, 2005.

[2] K. Wang, "Applying data mining to manufacturing: the nature and implications," Journal of Intelligent Manufacturing, vol. 18, no. 4, 2007, pp. 487-495.

[3] A. Kusiak, "Data mining: manufacturing and service applications," International Journal of Production Research, vol. 44, no. 18-19, 2006, pp. 4175-4191.

[4] L. Rokach and O. Maimon, "Data mining for improving the quality of manufacturing: a feature set decomposition approach," Journal of Intelligent Manufacturing, vol. 17, no. 3, 2006, pp. 285-299.

[5] Mark A. Friedl and Carla E. Brodley, "Decision tree classification of land cover from remotely sensed data," Remote sensing of environment, vol. 61, no. 3, 1999, pp. 399-409.

[6] J. Ross Quinlan, *C4.5: programs for machine learning,* Elsevier, 1993.

[7] C. Y. Jae, "Leverage big data technology in manufacturing," Journal of the Korean Institute of Communication Sciences, vol. 29, no. 11, 2012, pp. 30-35.

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," ACM SIGKDD explorations newsletter, vol.11, no. 1, 2009, pp. 10-17.

Sungsu Choi

He is an assistant researcher who is working for Yura Corp., Korea. He received the B.S. in Mobile Software Division in 2015, and also received M.S. in Business Data Convergence from Chungbuk National University, Korea in 2017. His research interests include a database, big data analytics, data mining and web services.

Lkhagvadorj Battulga

He is a student who is doing his master's degree at Chungbuk National University in computer science. He received the B.S. in computer science from National University of Mongolia. His research interests include a database, big data analytics, data mining and programming languages.

Aziz Nasridinov

He is an assistant professor who is working for Department of Computer Science at Chungbuk National University, Korea. He received the B.S. in computer science from Tashkent University of Information Technologies, Uzbekistan in 2006, and also received M.S., Ph.D. in computer science from Dongguk University,

Korea in 2009 and 2012, respectively. His research interests include a database, big data analytics, data mining and web services.

Kwan-Hee Yoo

He is a professor who is working for Department of Computer Science at Chungbuk National University, Korea. He received the B.S. in Computer Science from Chonbuk National University, Korea in 1985, and also received M.S., Ph.D. in computer science from KAIST (Korea Advanced Institute of Science and Technology), Korea in 1988 and 1995, respectively. His research interests include a u-Learning system, computer graphics, 3D character animation, dental/medical applications.