Steel Plates Faults Diagnosis with Data Mining Models

Mahmoud Fakhr and Alaa M. Elsayad
Department of Computers and Systems,
Electronics Research Institute, 12622 Bohoth St., Dokki, Geza, Egypt

Abstract: Problem statement: Over the last two decades, Fault Diagnosis (FD) has a major importance to enhance the quality of manufacturing and to lessen the cost of product testing. Actually, quick and correct FD system helps to keep away from product quality problems and facilitates precautionary maintenance. FD may be considered as a pattern recognition problem. It has been gaining more and more attention to develop methods for improving the accuracy and efficiency of pattern recognition. Many computational tools and algorithms that have been recently developed could be used. **Approach:** This study evaluates the performances of three of the popular and effective data mining models to diagnose seven commonly occurring faults of the steel plate namely; Pastry, Z_Scratch, K_Scatch, Stains, Dirtiness, Bumps and Other_Faults. The models include C5.0 decision tree (C5.0 DT) with boosting, Multi Perception Neural Network (MLPNN) with pruning and Logistic Regression (LR) with step forward. The steel plates fault dataset investigated in this study is taken from the University of California at Irvine (UCI) machine learning repository. **Results:** Given a training set of such patterns, the individual model learned how to differentiate a new case in the domain. The diagnosis performances of the proposed models are presented using statistical accuracy, specificity and sensitivity. The diagnostic accuracy of the C5.0 decision tree with boosting algorithm has achieved a remarkable performance with 97.25 and 98.09% accuracy on training and test subset. C5.0 has outperformed the other two models. **Conclusion:** Experimental results showed that data mining algorithms in general and decision trees in particular have the great impact of on the problem of steel plates fault diagnosis.

Key words: Fault Diagnosis (FD), University of California at Irvine (UCI), Logistic Regression (LR), Multi Perception Neural Network (MLPNN), Artificial Neural Networks (ANNs)

INTRODUCTION

A fault may be defined as an unacceptable difference of at least one characteristic property or attribute of a system from acceptable usual typical performance. Therefore, fault diagnosis is the description of the kind, size, location and time of discover of a fault. The main purpose of any fault diagnosis system is to determine the location and occurrence time of possible faults on the basis of accessible data and knowledge about the performance of diagnosed processes. Manual fault diagnosis system is the traditional way where an expert with electronic meter tries to obtain some information about relevant operational equipment, check the maintenance manual and then diagnosed the probable causes of a particular fault. However, intelligent fault diagnosis techniques can provide quick and correct systems that help to keep away from product quality problems and facilitates precautionary maintenance. These intelligent systems have been used different artificial intelligent and data

mining models and they should be simple and efficient. Decision tree, support vector machine, fuzzy logic algorithm, neural network and statistical algorithms are alternative approaches that are commonly employed nowadays in the industrial context to detect the occurrence of failure or faults (Seng-Yi and Chang, 2011). The numbers of fault diagnosis papers that are published in the sciencedirect database in 2010, 2011 and 2012 (in press) are 512, 732 and 39 studys respectively (searching date 31-10-2011). Faults diagnosis problems are representing challenging and attracting applications for experts and researchers. Recent reviews articles can be found in (Faiz and Ojaghi, 2009; Venkatasubramanian *et al.*, 2003a; 2003b; Zhang and Jiang, 2008; Ma and Jiang, 2011; Maurya *et al.*, 2007).

This study evaluates the performances of three of the popular and effective data mining models to diagnose seven commonly occurring faults of the steel plate namely; Pastry, Z_Scratch, K_Scatch, Stains,

---

Corresponding Author: Mahmoud Fakhr, Department of Computers and Systems, Electronics Research Institute, Bohoth St., Dokki, Geza, Egypt

Dirtiness, Bumps and Other_Faults. The models include C5.0 decision tree (C5.0 DT) with boosting, Multi Perceptron Neural Network (MLPNN) with pruning and logistic regression (LR) with step forward. DTs focus on conveying the relationship among the rules that expressed the results. They have expressive design and allow for non-linear relations between independent attributes and their outcomes and isolates outliers. The C5.0 DT model is a recent invented DT algorithm; it includes discretization of numerical attributes using information theory based functions, boosting, pre and post-pruning and some other state-of-the-art options for building DT model. Logistic Regression (LR), also known as nominal regression, is a statistical technique for classifying records based on values of input attributes. It is similar to linear regression but takes a categorical target field instead of a numeric one. LR works by building a set of equations that relate the input attribute values to the probabilities associated with each of the output attribute categories. Once the model is generated, it can be used to estimate probabilities for new data. For each record, a probability of membership is computed for each possible output category. The target category with the highest probability is assigned as the predicted output value for that record (Maalouf, 2011). A neural network, sometimes called a multilayer perception, is basically a simplified model of the way the human brain processes information. It works by simulating a large number of interconnected simple processing units that resemble abstract versions of neurons. The processing units are arranged in layers. There are typically three parts in a neural network: an input layer, with units representing the input fields; one or more hidden layers and an output layer, with a unit or units representing the output field (s). The units are connected with varying connection strengths or weights.

MATERIALS AND METHODS

Steel plate's faults dataset: The Steel Plates Faults Data Set used in the study comes from the UCI Machine Learning Repository (Frank and Asuncion, 2010). Steel Plates Faults Data Set is one of the datasets in the Repository, which classifies steel plates' faults into 7 different types: Pastry, Z_Scratch, K_Scatch, Stains, Dirtiness, Bumps and Other_Faults. The goal was to train machine learning for automatic pattern recognition. The dataset includes 1941 instances, which have been labeled by different fault types. Table 1 shows class distribution and list of attributes. The detailed information and the whole dataset can be

accessed from http://archive.ics.uci.edu/ml/datasets/Steel+Plates+Faults. The dataset was donated by Semeion, Research Center of Sciences of Communication, Via Sersale 117, 00128, Rome, Italy. The first used of the dataset in July 2010 (Buscema *et al.*, 2010). Each instance of the dataset owns 27 independent variables and one fault type.

Literature review: Artificial Intelligence (AI) tools are introduced for enhancing the accuracy of faults identification. In (Leger *et al.*, 1998) the author examined the feasibility of applying cumulative summation control charts and artificial neural networks together for fault diagnosis. Simani and Fantuzzi (2000) proposed a two-stage faults diagnosis method to test a neural network model of a power plant. Lo *et al.* (2002) tried to address the problem of fault diagnosis via integration of genetic algorithms and qualitative bond graphs. Hung and Wang (2004) presented a novel cerebellar model articulation controller neural network method for the fault diagnosis of power transformers. Dong *et al.* (2008) combined rough set and fuzzy wavelet neural network to diagnose faults of power transformers, concluding that the diagnosis accuracy may be limited by the hidden layer numbers and correlated training parameters of neural networks. Lau *et al.* (2010) presented an adaptive neuro–fuzzy inference system for online fault diagnosis of a gas-phase polypropylene production process. Eslamloueyan (2011) proposed a hierarchical artificial neural network for isolating the faults of the Tennessee–Eastman process, which was proved efficient.

Classification models:
Decision tree: DT models are powerful classification algorithms. They are becoming increasingly more popular with the growth of data mining applications (Nisbet *et al.*, 2009). As the name implies, this model recursively separates data samples into branches to construct a tree structure for the purpose of improving the classification accuracy. Each tree node is either a leaf node or decision node. All decision nodes have splits, testing the values of some functions of data attributes. Each branch from the decision node corresponds to a different outcome of the test. Each leaf node has a class label attached to it. General algorithm to build a DT is as follows:

- Start with the entire training subset and a vacant tree
- If all training samples at the current node n are of the same class label c, then the node becomes a leaf node with label c

- Or else, select the splitting attribute x that is the most important in separating the training samples into different classes. This attribute x becomes a decision node
- A branch is created for each individual value of x and the samples are partitioned accordingly
- The process is iterated recursively until a certain value of specified stopping criterion is achieved

Different DT models use different splitting algorithms that maximize the purity of the resulting classes of data samples. Popular DT models include ID3, C4.5 (Quinlan, 1986; 1993), CART (Breiman, 1984), QUEST (Loh and Shih, 1997), CHAID (Berry and Linoff, 1997) and C5.0. Common splitting algorithms include Entropy based information gain (used in ID3, C4.5, C5.0), Gini index (used in CART) and Chi-squared test (used in CHAID). This study uses the C5.0 DT algorithm which is an improved version of C4.5 and ID3 algorithms. It is a commercial product designed by Rule Quest Research Ltd Pty to analyze huge datasets and is implemented in SPSS Clementine workbench data mining software. C5.0 uses information gain as a measure of purity, which is based on the notion of entropy as in Eq. 1 and 2. If the training subset consists of n samples $(x_1,y_1),\ldots,(x_n,y_n)$, $x_i \in R^p$ is the independent attributes of the sample i and y_i is a predefined class $Y=\{c_1,c_2,\ldots,c_k\}$. Then the entropy, entropy(X), of the set X relative to this n-wise classification is defined as:

$$entropy(X) = (\sum_{i=0}^{n} -p_i \log_2 p_i) \qquad (1)$$

where, p_i is the ratio of X fitting in class c_i.

Information gain, gain(X, A) is simply the expected reduction in entropy caused by partitioning the set of samples, X, based on an attribute A:

$$gain(X,A) = entropy(X) - \sum_{v \in values(A)} \frac{|X_v|}{|X|} entropy(X_v) \qquad (2)$$

where, values(A) is the set of all possible values of attribute A and X_v is the subset of X for which attribute A has the attribute value v, i.e., $X_v = \{x \in X \mid A(x) = v\}$.

Boosting, winnowing and pruning are three methods used in the C5.0 tree construction; they propose to build the tree with the right size (Berry and Linoff, 1997). They increase the generalization and reduce the over fitting of the DT model. Boosting is a method for combining classifiers; it works by building multiple models in a sequence. The first model is built

in the usual way. Then, the second model is built in such a way that it focuses on the samples that were misclassified by the first model. Then the third model is built to focus on the second model's errors and so on (Nisbet *et al*., 2009). When a new sample is to be classified, each model votes for its predicted class and the votes are counted to determine the final class. Winnowing method investigates the usefulness of predictive attributes before starting to build the model (Lau *et al*., 2010). This ability to pick and choose among the predictive attributes is an important advantage of tree-based modeling techniques. Winnowing method preselect a subset of the attributes that will be used to construct the tree. Attributes that are irrelevant are excluded from the tree-building process.

In case of the current steel plate dataset, only 13 attributes have been selected to build the tree. Pruning is the last method used to increase the performance of the C5.0 DT model here. It consists of two steps; pre-pruning and post-pruning (Eslamloueyan, 2011). Pre-pruning step allows only nodes with minimum number of samples (node size). Post-pruning step reduces the tree size based on the estimated classification errors.

Multilayer perceptron neural network: Artificial Neural Networks (ANNs) are normally known as biologically motivated and highly sophisticated analytical techniques. They are capable of modelling extremely complex non-linear functions. Formally defined, ANNs are analytic techniques modelled after the processes of learning in the cognitive system and the neurological functions of the brain and capable of predicting new patterns (on specific attributes) from other patterns (on the same or other attributes) after executing a process of so-called learning from existing data (Haykin, 2009). Multilayer Perceptron Neural Network (MLPNN) with back-propagation is the most popular ANN architecture. MLPNN is known to be a powerful function approximator for prediction and classification problems. MLPNN's structure is organized into layers of neurons input, output and hidden layers. There is at least one hidden layer, where the actual computations of the network are processed. Each neuron in the hidden layer sums its input attributes x_i after multiplying them by the strengths of the respective connection weights w_{ij} and computes its output y_j using Activation Function (AF) of this sum. AF may range from a simple threshold function, or a sigmoidal, hyperbolic tangent, or radial basis function Eq. 3:

$$y_i = f(\sum w_{ij} x_i) \qquad (3)$$
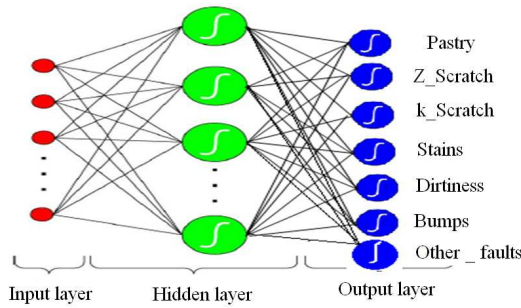
where, f is the activation function

Fig. 1: Three-layer neural network for steel plates faults diagnosis

Back-Propagation (BP) is a common training technique for MLPNN. BP works by presenting each input patterns to the network where the estimated output is computed by performing weighted sums and transfer functions. The sum of squared differences between the desired and estimated values of the output neurons E is defined as:

$$E = \frac{1}{2}\sum_i (y_{dj} - y_j)^2 \tag{4}$$

where, y_{dj} is the desired value of output neuron j and y_j is the estimated output of that neuron.

In Equation 3, each weight w_{ij} is adjusted to reduce the error E of Eq. 4 as fast, quickly as possible. BP applies a weight correction to reduce the difference between the network estimated outputs and the desired ones; i.e., the neural network can learn and can thus reduce the future errors (Quinlan, 1993; Berry and Linoff, 1997).

Figure 1 shows the architecture of three layers perceptron neural network used for the diagnosis of Steel plates faults. Although, BP is standard, simple to implement and in general it works well, it has slow convergence approach and can get stuck in local minima (Haykin, 1994). Another drawback of MLPNN models is that they require the initialization and adjustment of many individual parameters to optimize their performance. In this study, the network is trained using the pruning approach to find the optimal network structure (Chaudhuri and Bhattacharya, 2000; Thimm *et al.*, 1996). The network starts with a large network and removes (prunes) the weakest neurons in the hidden and input layers as training proceeds.

Logistic regression: Logistic Regression (LR) is a generalization of linear regression (Maalouf, 2011). It is a nonlinear regression technique for prediction of dichotomous (binary) class attribute in terms of the predictive ones. The class attribute represent the status of the consumer (creditworthy, y = 1 or Not

creditworthy, y = 0). Actually, the algorithm does not predict the class attribute but predicts the odds of its occurrence. The expected probability of a positive outcome P(y=1) for the class attribute is modeled as follows:

$$P(y = 1) = \frac{1}{1 + e^{-(B_0 + \sum_{i=1}^{n} B_i x_i)}} \tag{5}$$

where, x_i, i = 1,…,n are the predictive attributes with real values, B_i are the corresponding regression coefficients and B_0 is a constant, all of which contribute to the probability.

While LR is a very powerful modeling tool, it assumes that the class attribute (the log odds, not the event itself) is linear in the coefficients of the predictive attributes (Bewick *et al.*, 2005). Eq. 5 is reduced to a linear regression model for the logarithm of Odds Ratio (OR) of positive outcome, i.e.:

$$\ln\left(\frac{P(y = 1)}{1 - P(y = 1)}\right) = B_0 + \sum_{i=1}^{n} B_i x_i \tag{6}$$

However, the right inputs must be chosen with their functional relationship to the class attribute.

RESULTS

The classification performance of each model is evaluated using three statistical measures; classification accuracy, sensitivity and specificity which are described in Eq. 7, 8 and 9. These measures are defined using the values of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). A true positive decision occurs when the positive prediction of the classifier coincided with a positive prediction of the expert. A true negative decision occurs when both the classifier and the expert suggested the absence of a positive prediction. False positive occurs when the classifier labels a negative case as a positive one. Finally, false negative occurs when the system labels a positive case as negative one. Classification accuracy is defined as the ratio of the number of correctly classified cases and is equal to the sum of TP and TN divided by the total number of cases N:

$$Accuracy = \frac{TP + TN}{N} \tag{7}$$

Sensitivity refers to the rate of correctly classified positive and is equal to TP divided by the sum of TP and FN. Sensitivity may be referred as a True Positive Rate:

$$Sensitivity = \frac{TP}{TP + FN} \tag{8}$$

Specificity refers to the rate of correctly classified negative and is equal to the ratio of TN to the sum of TN and FP. False Positive Rate equals (100-specificity):

$$\text{Specificity} = \frac{TN}{TN + FP} \qquad (9)$$

Sensitivity in Equation 8 measures the proportion of actual positives which are correctly identified as such while specificity in Eq. 9 measures the proportion of negatives which are correctly identified. Finally, accuracy in Eq. 7 is the proportion of true results, either true positive or true negative, in a population. It measures the degree of veracity of a diagnostic test on a condition. Fig. 2 demonstrates the component nodes of the proposed stream. This stream is implemented in SPSS Clementine data mining workbench using Intel core 2 Dup CPU with 2.1 GHz. Clementine uses client/server architecture to distribute requests for resource-intensive operations to powerful server software, resulting in faster performance on larger datasets. The software offers many modeling techniques, such as prediction, classification, segmentation and association detection algorithms.

Faults dataset node is connected directly to an Excel file that contains the source data. The dataset includes 1941 instances, which have been labeled by one of the prescribed 7 fault types: Pastry, "Z Scratch", "K Scratch", Stains, Dirtiness, Bumps and "Other Faults". Each instance of the dataset owns 27 independent variables and one fault type. The dataset is explored for incorrect, inconsistent or missing data. These predictive attributes are of mixed numeric types; flag or range, while the target class is of type nominal. Type node specifies the field metadata and properties that are important for modeling and other works in Clementine. These properties include specifying a usage type, setting options for handling missing values, as well as setting the role of an attribute for modeling purposes; input or output. The first 27 attributes in Table 1 are defined as input (predictive) attributes and the fault's type is defined as target class. Partition node is used to generate a partition field that splits the dataset into separate subsets for the training and test the models by the ratio of 70:30% respectively. The training subset is used to estimate the model parameters, while the test one is used to independently assess the individual model. These models are applied again to the entire dataset and to any new data. Logistic node is trained using forward method where the model is built by moving forward step by step. With this method, the initial model is the simplest model and only the constant and terms can be added to the model as in Eq. 6. At each step, attributes not yet in the model are tested based on how much they would improve the model; and the best of those attributes is added to the model.
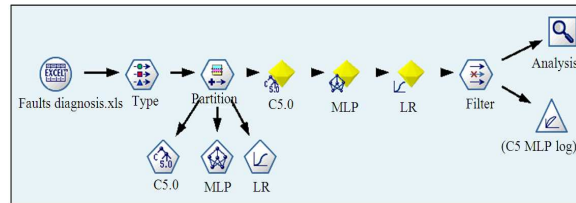

Fig. 2: Stream of fault diagnosis using three classification models: DT, MLPNN and LR

When no more terms can be added, or the best candidate term does not produce a large-enough improvement in the model, the final model is generated. Two min and twenty four seconds are required to build this model for the steel plate's faults dataset. However, LR system came out the second best classifier with classification accuracies of 73.26% of training and 72.59% of the test samples. C5.0 node is a boosted decision tree model with C5.0 algorithm which is trained using pruning and winnowing methods to increase the model accuracy. The number of trails of boosting algorithm is 10, the minimum number of samples per node is set to be 2 and the system uses equal misclassification costs. The high speed property is a notable feature of C5.0 DT model; it clearly uses a special technique, although this has not been described in the open literature. The classification accuracies without boosting algorithms are 90.57% of training subset and 90.57% for test one while these accuracies with boosting algorithms are 97.25 and 98.09%. Boosting with 10 trails enhances the accuracies of the tree to reach higher accuracies for training and test samples. The time required to build single C5.0 tree is below one sec. While boosting tree requires 11 seconds with ten trails. This model is the best one among the probability estimation classifiers. MLPNN classifier node is trained using the pruning method (Thimm *et al.*, 1996). It begins with a large network and removes the weakest neurons in the hidden and input layers as training proceeds. The stopping criterion is set based on time. The network is given two min for training. Using the steel plate's faults dataset, the MLPNN with pruning method has achieved 74.79 and 79.14% classification accuracies for training and test datasets. The resulting structure consists of four layers; one input, two hidden layers and the output one with 6, 20, 15 and 7 neurons respectively. Filter, Analysis and Evaluation nodes are used to select and rename the classifier outputs in order to compute the performance statistical measures and to graph the evaluation charts.

The steel plate's faults dataset is divided for training the models and test them by the ratio of 70:30% respectively.

Table 1: Steel plates dataset; class distribution and predictive attributes

Output class	# of Cases	---Predictive attributes---			
Pastry	158	Attribute 1	X_Minimum	Attribute 14	Steel_Plate_Thickness
Z_Scratch	190	Attribute 2	X_Maximum	Attribute 15	Edges_Index
K_Scatch	391	Attribute 3	Y_Minimum	Attribute 16	Empty_Index
Stains	72	Attribute 4	Y_Maximum	Attribute 17	Square_Index
Dirtiness	55	Attribute 5	Pixels_Areas	Attribute 18	Outside_X_Index
Bumps	402	Attribute 6	X_Perimeter	Attribute 19	Edges_X_Index
Other_Faults	673	Attribute 7	Y_Perimeter	Attribute 20	Edges_Y_Index
		Attribute 8	Sum_of_Luminosity	Attribute 21	Outside_Global_Index
		Attribute 9	Minimum_of_Luminosity	Attribute 22	LogOfAreas
		Attribute 10	Maximum_of_Luminosity	Attribute 23	Log_X_Index
		Attribute 11	Length_of_Conveyer	Attribute 24	Log_Y_Index
		Attribute 12	TypeOfSteel_A300	Attribute 25	Orientation_Index
		Attribute 13	TypeOfSteel_A400	Attribute 26	Luminosity_Index
				Attribute 27	SigmoidOfAreas

Table 2: Confusion matrices of individual models and their ensemble for test subset (997 data samples)

Classifiers	Desired results	Output results						
		Pastry	Z_Scratch	K_Scatch	Stains	Dirtiness	Bumps	Other_Faults
C5.0	Pastry	79	0	0	0	0	0	3
	Z_Scratch	0	89	0	0	0	0	5
	K_Scatch	0	0	191	0	0	0	1
	Stains	0	0	0	47	0	0	0
	Dirtiness	0	0	0	0	22	0	1
	Bumps	1	0	0	1	1	203	4
	Other_Faults	1	0	0	0	0	1	347
MLPNN	Pastry	34	4	0	0	2	18	24
	Z_Scratch	0	84	0	0	0	1	9
	K_Scatch	0	0	181	0	0	1	10
	Stains	0	0	0	43	0	2	2
	Dirtiness	0	0	0	0	16	4	3
	Bumps	3	4	1	0	4	156	42
	Other_Faults	12	14	2	0	2	44	275
LOG	Pastry	53	11	0	3	4	4	7
	Z_Scratch	0	85	1	0	1	2	5
	K_Scatch	1	1	173	7	1	4	5
	Stains	0	0	0	46	0	1	0
	Dirtiness	1	0	0	0	19	2	1
	Bumps	13	26	1	12	8	112	38
	Other_Faults	38	40	6	43	24	58	140

The training set is used to estimate the model parameters, while the test set is used to independently assess the individual model. These models are applied again to the entire dataset and to any new data. The time required to build each model with the dataset is variable; ranging from few seconds up to two min for the neural network. In C5.0 DT model, boosting can significantly improve the accuracy of model, but it also requires longer training. It works by building multiple models in a sequence. Cases are classified by applying the whole set of models to them and using a voting procedure to combine the separate predictions into one overall prediction. The predictions of all models are compared to the original classes to identify the values of true positives, true negatives, false positives and false negative. These values have been computed to construct the confusion matrix as shown in Table 2. The values of the statistical measures (sensitivity, specificity and total classification accuracy) of the three models were

computed and presented in Table 3 and 4. Sensitivity and Specificity approximate the probability of the positive and negative labels being true. They assess the usefulness of the algorithm on a single model. Using the results shown in Table 2, it can be seen that the sensitivity, specificity and classification accuracy of C5.0 DT model has achieved the best success of test samples classification.

The gain chart provides a visual summary of the usefulness of the information provided by the classification models for predicting categorical dependent attributes. The chart summarizes the utility that can be expected by using the respective predictive models, as compared to the baseline information only. The gain charts for Decision tree with C5.0, Logistic regression with forward method and neural network models trained in SPSS Clementine for training and test subsets are shown in Fig. 3. The higher lines in the gain charts indicate better models, especially on the left side of the chart.

Table 3: Values of the statistical measures of logistic regression, neural network and C5.decision tree for only test subsets

Classifier	Desired results	Classification accuracy (%)	Specificity (%)	Sensitivity (%)
LR	Pastry	91.78	94.21	64.63
	Z_Scratch	91.27	91.36	90.43
	K_Scatch	97.29	99.01	90.10
	Stains	97.29	93.16	97.87
	Dirtiness	95.79	96.10	82.61
	Bumps	83.05	90.98	53.33
	Other_faults	73.42	91.36	40.11
MLPNN	Pastry	93.68	98.36	41.46
	Z_Scratch	96.79	97.56	89.36
	K_Scatch	98.60	99.63	94.27
	Stains	99.60	100.00	91.49
	Dirtiness	98.50	99.18	69.57
	Bumps	87.56	91.11	74.29
	Other_faults	60.18	86.11	12.03
C5.0	Pastry	99.50	96.34	99.78
	Z_Scratch	99.50	94.68	100.00
	K_Scatch	99.90	99.48	100.00
	Stains	99.90	100.00	99.89
	Dirtiness	99.80	95.65	99.90
	Bumps	99.20	96.67	99.87
	Other_faults	98.40	99.43	97.84

Table 4: Overall performance of the three classifiers

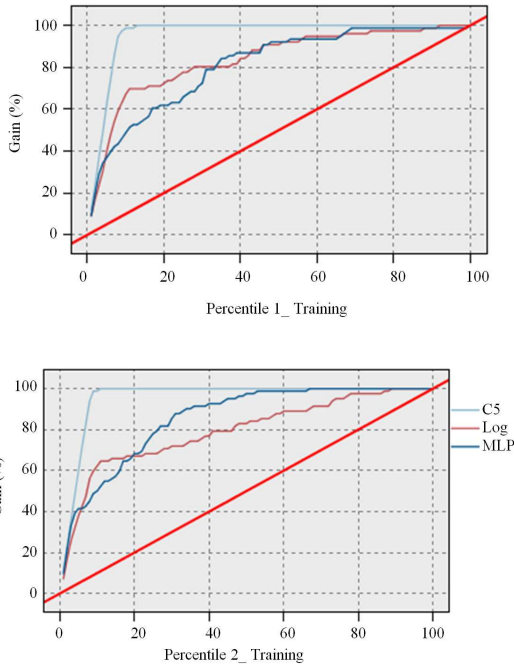| Classifier	Testing		Training	
	Correct (%)	Wrong (%)	Correct (%)	Wrong (%)
LR	605.00	339.00	628.00	369.00
	64.09	35.91	62.99	37.01
MLP	706.00	238.00	789.00	208.00
	74.79	25.21	79.14	20.86
C5.0 DT	918.00	26.00	978.00	19.00
	97.25	2.75	98.09	1.91

Fig. 3: The cumulative gain charts of the three models for training and test subsets
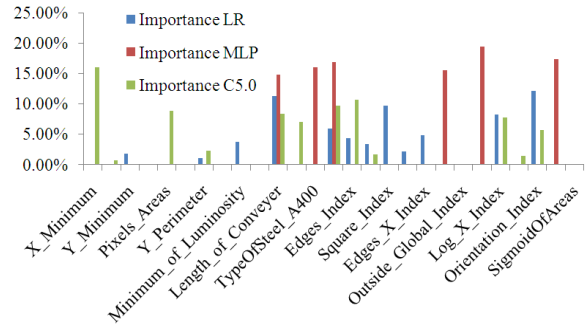

Fig. 4: Sensitivity analysis of the twenty seven attributes with each classification model

These charts depict that the performances of the decision tree with C5.0 learning algorithm is the best model for training and test subsets. Neural network model is the second best classifier and finally the logistic regression is the worst one. Sensitivity analysis helps to gain some insight into the predictive attributes used in the present classification problem. The analysis provides information about the relative importance of the predictive (input) attributes in predicting the output attribute(s). The basic idea is that the inputs to the classifier are perturbed slightly and the corresponding change in the output is reported as a percentage change in the output (Principe *et al*., 2000). The first input is varied between its mean plus (or minus) a user-defined number of standard deviations, while all other inputs are fixed at their respective means. The classifier output is computed and recorded as the percent change above and below the mean of that output channel. This process is repeated for each and every input attribute. The sensitivity analysis is performed for this study and presented in a graphical format in Fig. 4. Seven attributes have the most importance; X_Minimum, Steel_Plate_Thickness, TypeOfSteel_A400, Orientation Index, Maximum of Luminosity, Luminosity_Index and LogOfAreas. The MLPNN model uses only six attributes to predict the output, both LR and C5.0 DT models use 13 attributes.

DISCUSSION

Experimental results have demonstrated that advanced data mining techniques can be used to develop models that possess a high degree of diagnostic accuracy. However, there are several issues involved with the data collection and data mining that warrant for further discussion. The amount and quality of the data are key components of the diagnostic accuracy. The measuring process may contain many features that create problems for the data mining techniques. The

datasets could be consisted of a large volume of heterogeneous data fields which usually complicates the use data mining techniques. The main criticism about data mining techniques is that they are not following all of the requirements of classical statistics. They use training and testing data sets drawn from the same dataset. In classical statistics, it can be argued that the testing set used in this case is not truly independent and for that reason, the results may be biased.

Despite these criticisms, data mining can be an important tool in the fault diagnosis problem by identifying patterns within the large sums of data, data mining can and should, be used to gain more insight into the faults, generate knowledge that can potentially fuel lead to further research in many areas of manufacturing. The high degree of diagnostic accuracy of the models evaluated here is just one example of the value of data mining.

CONCLUSION

The objective of this study was to demonstrate the application of classification techniques in the problem of steel plates fault diagnosis and to describe the strengths and weaknesses of the methods described. Three different classification models have been evaluated. These models are derived from different family namely; Multi-Layer Perceptron Neural Network (MLPNN), C5.0 Decision Tree (DT) and Logistic Regression (LR). These models are optimized using different methods. Pruning method was used to find the optimal structure of MLPNN model. The C5.0 DT has been built using boosting algorithm with 10 trails. Finally, the LR model was constructed using the stepwise forward method to gradually build the system. The performance of these models were investigated using known sets of steel plates proven faults features obtained from the University of California at Irvine (UCI) machine learning repository. Experimental results have shown that the C5.0 decision tree gives better performance than the other models. Furthermore the boosting algorithm enhances the performance of C5.0 DT algorithm. Although data mining techniques are capable of extracting patterns and relationships hidden deep into large datasets, without the cooperation and feedback from the experts and professional, their results are useless. The patterns found via data mining techniques should be evaluated by professionals who have years of experience in Predicting steel plates faults.

REFERENCES

Berry, M.J.A. and G. Linoff, 1997. Data Mining Techniques: for Marketing, Sales and Customer Support. 1st Edn., Wiley, New York, ISBN-10: 0471179809 pp: 454.

Bewick, V., L. Cheek and J. Ball, 2005. Statistics review 14: Logistic regression. Crit Care 9: 112-118. DOI: 10.1186/cc3045

Breiman, L., 1984. Classification and Regression Trees. 1st Edn., Wadsworth International Group, Belmont, ISBN-10: 0534980538 pp: 358.

Buscema, M., S. Terzi and W. Tastle, 2010. A new meta-classifier. Proceedings of the North American Fuzzy Inform Processing Society, Jul. 12-14, IEEE Xplore Press, Toronto, pp: 1-7. DOI: 10.1109/NAFIPS.2010.5548298

Chaudhuri, B.B. and U. Bhattacharya, 2000. Efficient training and improved performance of multilayer perceptron in pattern classification. Neurocomputing, 34: 11-27. DOI: 10.1016/S0925-2312(00)00305-2

Dong, L., D. Xiao, Y. Liang and Y. Liu, 2008. Rough set and fuzzy wavelet neural network integrated with least square weighted fusion algorithm based fault diagnosis research for power transformers. Elec. Power Syst. Res., 78: 129-136. DOI: 10.1016/J.EPSR.2006.12.013

Eslamloueyan, R., 2011. Designing a hierarchical neural network based on fuzzy clustering for fault diagnosis of the Tennessee–Eastman process. Applied Soft Comput., 11: 1407-1415. DOI: 10.1016/J.ASOC.2010.04.012

Faiz, J. and M. Ojaghi, 2009. Different indexes for eccentricity faults diagnosis in three-phase squirrel-cage induction motors: A review. Mechatronics, 19: 2-13. DOI: 10.1016/j.mechatronics.2008.07.004

Frank, A. and A. Asuncion, 2010. UCI machine learning repository Irvine. University of California.

Haykin, S.S., 1994. Neural Networks: A Comprehensive Foundation. 1st Edn., Macmillan, New York, ISBN-10: 0023527617, pp: 696.

Haykin, S.S., 2009. Neural Networks and Learning Machines. 3rd Edition, Prentice Hall, New York, ISBN-10: 0131471392 pp: 906.

Hung, C.P. and M.H. Wang, 2004. Diagnosis of incipient faults in power transformers using CMAC neural network approach. Elec. Power Syst. Res., 71: 235-244. DOI: 10.1016/j.epsr.2004.01.019

Lau, C.K., Y.S. Heng, M.A. Hussain and M.I.M. Nor, 2010. Fault diagnosis of the polypropylene production process (UNIPOL PP) using ANFIS. ISA Trans., 49: 559-566. DOI: 10.1016/j.isatra.2010.06.007

Leger, R.P., W.J. Garland and W.F.S. Poehlman, 1998. Fault detection and diagnosis using statistical control charts and artificial neural networks. Artif. Int. Eng., 12: 35-47. DOI: 10.1016/S0954-1810(96)00039-8

Lo, C.H., Y.K. Wong, A.B. Rad and K.M. Chow, 2002. Fusion of qualitative bond graph and genetic algorithms: A fault diagnosis application. ISA Trans., 41: 445-456. 10.1016/S0019-0578(07)60101-3

Loh, W.Y. and Y.S. Shih, 1997. Split selection methods for classification trees. Stat. Sinica, 7: 815-840.

Ma, J. and J. Jiang, 2011. Applications of fault detection and diagnosis methods in nuclear power plants: A review. Prog Nucl. Energy, 53: 255-266. DOI: 10.1016/j.pnucene.2010.12.001

Maalouf, M., 2011. Logistic regression in data analysis: An overview. Int. J. Data Anal. Tech. Strategies, 3: 281-299. DOI 10.1504/IJDATS.2011.041335

Maurya, M.R., R. Rengaswamy and V. venkatasubramanian, 2007. Fault diagnosis using dynamic trend analysis: A review and recent developments. Eng. Appli. Artif. Intell., 20: 133-146. DOI: 10.1016/j.engappai.2006.06.020

Nisbet, R., J. Elder and G. Miner, 2009. Handbook of Statistical Analysis and Data Mining Applications. 1st Edn., Academic Press, Burlington, ISBN-10: 9786612168314 pp: 824.

Principe, J.C., N.R. Eulianoand and W.C. Lefebvre, 2000. Neural and Adaptive Systems: Fundamentals Through Simulations. Wiley, New York, ISBN-10: 0471351679 pp: 656.

Quinlan J.R., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco, ISBN-10: 1558602380 pp: 302.

Quinlan, J.R., 1986. Induction of decision trees. Machine Learn., 1: 81-106. DOI: 10.1007/BF00116251

Seng-Yi, S.D.L. and W.L. Chang, 2011. Application of multiclass support vector machines for fault diagnosis of field air defense gun. Expert Syst. Appli., 38: 6007-6013. DOI: 10.1016/j.eswa.2010.11.020

Simani, S. and C. Fantuzzi, 2000. Fault diagnosis in power plant using neural networks. Inform. Sci., 127: 125-136. DOI: 10.1016/S0020-0255(00)00034-7

Thimm, G. and E. Fiesler 1996. Neural network pruning and pruning parameters. Proceedings of the 1st Workshop on Soft Computing, (WSC' 96), Nagoya, Japan, pp: 1-2.

Venkatasubramanian, V., R. Rengaswamy, K. Yin and S.N. Kavuri, 2003b. A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. Comp. Chem. Eng., 27: 293-311. DOI: 10.1016/S0098-1354(02)00160-6

Venkatasubramanian, V., R. Rengaswamy, S.N. Kavuri and K. Yin, 2003a. A Review of process fault detection and diagnosis: Part III: Process history based methods. Original Res. Article Comp. Chem. Eng., 27: 327-346. DOI: 10.1016/S0098-1354(02)00162-X

Zhang, Y. and J. Jiang, 2008. Bibliographical review on reconfigurable fault-tolerant control systems. Annual Rev. Control, 32: 229-252. DOI: 10.1016/J.ARCONTROL.2008.03.008