International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 05 | May 2019 www.irjet.net p-ISSN: 2395-0072

Comparison Analysis of Machine Learning algorithms for Steel Plate Fault Detection

Ashish Kumar Srivastava[1]

[1]*Assistant Professor*, Dept. of *Mechanical Engineering*, *Goel Institute of Technology & Management, Lucknow, U.P., India*

Abstract - *Steel has played an important role in evolution of man as a society. From making machines to beautiful art works and form huge structures to various modes of transport such as Titanic and Boeing. Having so wide spread applications steel also have certain restrictions which may result in failure or fault if not monitored properly time to time with precision. In a machine where due to extreme level of sound, heat and other types of hazardous factors it becomes a tedious job for a man to monitor the same, here this can be easily and precisely countered by using Machine learning techniques and any unwanted failure and fault can be predicted within time to avoid loss of man, power and money. This paper presents the comparative study of machine learning algorithms. All experiments have been performed on Steel Plates faults dataset provided by UCI repository using Waikato Environment for Knowledge Analysis (WEKA) tool. This work includes study of different machine learning algorithm named decision tree, random forest, AdaBoost, K-Nearest neighbor and Support Vector Machine. Our experiments have shown that random forest algorithm is the best performing with79.23 % with 0.203 RMSE.*

***Key Words*: Steel Plate, Machine learning, Classification, WEKA**

1. INTRODUCTION

Steel is an integral part for most of the engineering application such as: rail bridges, steam generators, railway wagons, automobile chasis, building construction materials, etc. Steel is classified as: Low carbon steel (Carbon % = 0.05 to 0.25), Mild Steel (Carbon % = 0.29 to 0.54), High carbon steel (Carbon % = 0.55 to 0.95), Very high carbon steel (Carbon % = 0.96 to 2.1) [12]. Carbon percentage provides the strength and toughness to the steel.

Steel plates offer a wide application in manufacturing of machineries, ships structure, bridges structures, etc. But they are subjected to various types of catastrophic failures like Titanic (in 1912) sank due to failure of steel plate which occurred due to thermal conditions. When we talk about mechanical testing and prediction of supposed failure of the steel the methods involved are, use of UTM (Universal Testing Machine) which is used to predict the mechanical properties such as tensile strength and compressive strength for various types of steel [13]. In this paper we have tried to cut short this tedious and long process to effectively predicting the fault chances so, that any such catastrophic event can be avoided while we use steel plates. In industry many a times production is severely hampered due to failure of machine components or structural defects which is very tedious and hard job to find before time of failure by a human being. But with the coupling of Machine learning to inspection and monitoring of this can help a lot to reduce the effort as well as wastage of time and money.

Many researches have worked on steel plate fault detection. [10] They have used Multi-Layer Perceptron Neural Network (MLPNN), C5.0 Decision Tree and Logistic Regression. This paper concludes that the C5.0 decision tree gives better performance [11] amongst similarly used machine learning algorithms models namely C5.0, MLPNN, Bayesian network (BN) and Ensemble model. This paper also concludes that C5.0 is performing better than other models. Most of the researchers have found that tree-based methods are performing well. Since there are many existing tree-based methods that can be used to increase the fault prediction rate.

The main aim of this paper is to provide the best solution for this problem using machine learning algorithms. In this paper, we have also provided the comparative study of machine learning algorithms and how they can be applied to achieve higher accuracy for steel plate fault detection.

The rest of the paper is organized as follow: Section II explains about different machine learning classification algorithms. Section III explains Experimental result which is further subdivided into multiple subsections like dataset description, performance parameter, experimental setup, results and discussion. Section IV has conclusion and last section presents the future direction of this work.

2. MACHINE LEARNING ALGORITHMS

This section explains about different machine learning algorithms used in experiments.

2.1 Decision Tree

Decision tree algorithm also known as J48 Algorithm is a tree-based classifier. This algorithm builds graph-based tree according to feature vectors and this trained tree is used at the time of inference to classify the new instances [3]. This

tree consists of nodes and edges where nodes are feature and edges represent value or rules that a child node can represent. To decide the order of features in tree different methods are used like information gain, entropy method etc. [4]

2.2 Random Forest

Random forest is an enhanced version of decision tree algorithm. This algorithm is presented by Leo Breiman [5]. Instead of building one tree, this algorithm builds multiple unpruned tree by randomly selecting features. At the time of inferencing, new test instance will pass through all trees and final prediction is taken with the help of majority voting method. This method gives equal chance to features to decide and provide final decision. This algorithm leads to give more accurate prediction over decision tree.

2.3 AdaBoost

This algorithm has been proposed by Freund and Schapire [6]. AdaBoost stands for adaptive boosting means increasing or boosting the accuracy of existing algorithm. This existing algorithm is named as weak classifier. This weak classifier can be any algorithm like decision tree, decision stump, K-NN algorithm. Main goal of this algorithm is to increase the classification accuracy of weak classifier. This algorithm uses iterative method to improve the weak classifier. Each time classifier learns from error and improve the cost function.

2.4 K- Nearest Neighbor

K-nearest neighbor algorithm is one of the instance-based learning algorithms. This method is also known as the lazy learning [7]. They require more time in testing phase as compared to training time. In k-nearest neighbor algorithm a new instance gets the class label according to the nearness to the training instance. There are different methods that can be used to find the nearness of any two instances. These methods are called distance metrics like Euclidean, Manhattan, City block, Chebyshev etc. In k-NN algorithm k represent the number of neighbors. For example, 5-NN represent five neighbors are considered to classify the new instance.

2.5 Support Vector Machine

SVM is one the famous machine learning algorithm. This algorithm divides the two class by finding the best decision boundary. A decision boundary said to be best if it has maximum margin between two classes. To find the maximum margin decision boundary, it uses hyperplane and projection concept. There is different variant of SVM algorithm, one is sequential minimal optimization (SMO) which divides the problem into sub problems which makes the algorithm [8].

3. EXPERIMENTAL RESULTS

This section presents experimental setup for steel plate faults detection and results. Subsection A explains dataset description, subsection B explains performance parameters to measure the correctness of the Machine learning algorithm. Subsection C has experimental setup information and last subsection D explains the result of machine learning algorithms.

3.1 Dataset Descriptions

This comparative study used Steel Plates Faults Data Set provided by UCI Machine Learning Repository. This dataset consists of 1941 instances and corresponding class labels. Steel Plate Fault's has seven categories namely Pastry, Z_Scratch, K_Scatch, Stains, Dirtiness, Bumps, other_Faults. Table 1 shows class label and number of instances corresponding to that class. Every instance has 27 features presented in Table 2 [1].

Table-1 Class Labels of Steel plates faults dataset [1]

Class label	Number of instances
1.Pastry	158
2.Z_Scratch	190
3.K_Scatch	391
4.Stains	72
5.Dirtiness	55
6.Bumps	402
7.Other_Faults	673

Table 2 Name of Features [1]

1.x_minimum	2.x_maximum	3.y_minimum
4.y_maximum	5.pixels_areas	6.x_perimeter
7.y_perimeter	8.sum_of_luminosity	9.minimum_of_luminosity
10.maximum_of_luminosity	11.length_of_conveyer	12.typeofsteel_a300
13.typeofsteel_a400	14.steel_plate_thickness	15.edges_index
16.empty_index	17.square_index	18.outside_x_index
19.edges_x_index	20.edges_y_index	21.outside_global_index
22.logofareas	23.log_x_index	24.log_y_index
25.orientation_index	26.luminosity_index	27.sigmoidofareas

3.2 Performance parameter

Evaluating machine learning algorithm is essential step. There are different types of evaluation parameters available

like classification accuracy, confusion matrix, Area under curve, Mean Squared Error, F1 score etc. In this paper, we are using classification accuracy, Root mean square error (RMSE) and confusion matrix to evaluate the machine learning algorithm.

$$Accuracy = \frac{Number\ of\ correctly\ classified\ instances}{Total\ number\ of\ instances}$$

RMSE is the standard deviation in predicted instance class labels. It measures, how well-trained machine learning model performs in testing time [9].

Confusion matrix describe performance of the classifier in tabular format. Rows of the table represent the actual class labels and columns represents the predicted class labels. Corresponding cell represents the number of instances classified.

3.3 Experimental setup

In this paper, we use WEKA tool for analysis [2]. WEKA is open source software which is collection of machine learning algorithm. It has specific input file format called arff. We have preprocessed dataset converting into arff format.

3.4 RESULTS

This subsection presents the result analysis of machine learning algorithms. Different machine learning algorithms such as k-nearest neighbors (K-NN), decision tree, random forest classifier and AdaBoost will be compared. All experiments have been performed on 10-fold cross validation and 20- fold cross validation datasets. Decision tree algorithm is information gain method to form tree. AdaBoost is using decision stump as weak classifier. In KNN algorithm, K is five means it uses five nearest neighbors to predict the final prediction.

Table 3 Comparison of Different Machine Learning Algorithm (10 Fold cross validation)

S. No.	Algorithm	Accuracy (%)	RMSE
1	Decision Tree	76.04	0.246
2	Random forest	79.39	0.204
3	AdaBoost	78.41	0.237
4	KNN	71.35	0.235
5	SVM	74.90	0.308

Table 3 presents the comparison of different machine learning algorithm with 10-fold cross validation dataset. Random forest with 79.39 % accuracy is the best machine learning algorithm among the algorithm present in Table 3. AdaBoost algorithm uses decision stump algorithm as weak classifier and this algorithm is also performed well with 78.41 % accuracy.

Table 4 presents the comparison of different machine learning algorithm with 20-fold cross validation dataset. Random forest and AdaBoost achieved 79% accuracy. But Random forest has less RMSE value than AdaBoost algorithm.

TABLE 4 Comparison of Different Machine Learning Algorithm (20 Fold cross validation)

S. No.	Algorithm	Accuracy (%)	RME
1	Decision Tree	77.27	0.241
2	Random forest	79.23	0.203
3	AdaBoost	79.08	0.232
4	KNN	71.61	0.236
5	SVM	75.16	0.308

From Table 3 and Table 4, it is clear that Random forest is performing best with 79.23% accuracy. Table 5 presents confusion matrix of Random forest algorithm with 10- Fold cross validation where rows have actual class label and columns have predicted class labels. Cell corresponding to same class label represents the correct classified instance of that class. Class label Pastry (P) got 83 correct classified instances out of 158 instances (refer to Table 1). Table 6 shows the class label percentage actuary of random forest algorithm.

Table 5 Confusion Matrix of Random Forest Algorithm (10 Fold cross validation)

ACTUAL/ PREDICTED	(P)	(Z)	(K)	(S)	(D)	(B)	(O)
Pastry (P)	**83**	3	0	0	0	16	56
Z_Scratch (Z)	0	**169**	2	0	0	1	18
K_Scatch (K)	1	0	**373**	0	0	1	16
Stains (S)	0	0	0	**64**	0	2	6

Dirtiness (D)	1	0	0	0	**46**	1	7
Bumps (B)	13	1	0	1	1	**277**	109
Other_Fault (O)	20	14	6	3	3	98	**529**

Table 6 Class Label Accuracy of Random Forest Algorithm (10 Fold cross validation)

Class Label	Class Label Accuracy
1.Pastry	53%
2.Z_Scratch	89%
3.K_Scatch	95%
4.Stains	89%
5.Dirtiness	84%
6.Bumps	69%
7.Other_Faults	79%

4. CONCLUSION

Experimental results have demonstrated that machine learning is extensively being used in the fault detection field. The main objective of this paper is to compare different machine learning algorithms applied for steel plate fault detection. We have used five machine learning algorithms named Decision tree, Random forest, AdaBoost, KNN, SVM. This work determines the most suitable algorithm for fault detection problem. Results show that random forest is the best algorithm for steel plate fault detection. Random forest algorithm achieves 79.23 % accuracy with minimum RMSE. Other than random forest, AdaBoost also performed well on this dataset. AdaBoost algorithm uses decision stump method. This algorithm improves the performance of the decision stump. Tree based machine learning algorithms are outperforming than other algorithms.

5. FUTURE SCOPES

This work can be further extended to find the best classification algorithm using deep learning approaches. One of the other areas can be finding an optimal subset of the features to do reduce the algorithm complexity and increasing the accuracy rate by removing irrelevant features. This work can also lead to develop an accurate and precise simulation model which can predict health of component depending upon various parameters that are time bounded.

REFERENCES

[1] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[2] Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

[3] S. K. Murthy, Automatic construction of decision trees from data: A multi-disciplinary survey, Data mining and knowledge discovery, vol. 2, no. 4, pp. 345-389, 1998.

[4] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, Supervised machine learning: A review of classification techniques, 2007.

[5] L. Breiman, Random forests, Machine learning, vol. 45, no. 1, pp. 5-32, 2001.

[6] Y. Freund and R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of computer and system sciences, vol. 55, no. 1, pp. 119-139, 1997.

[7] A.-L. Jousselme and P. Maupin, Distances in evidence theory: Comprehensive survey and generalizations, International Journal of Approximate Reasoning, vol. 53, no. 2, pp. 118-145, 2012.

[8] J. C. Platt, 12 fast training of support vector machines using sequential minimal optimization, Advances in kernel methods, pp. 185-208, 1999.

[9] Willmott, Cort J., and Kenji Matsuura. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate research 30.1 79-82, 2005.

[10] Fakhr, Mahmoud, and Alaa M. Elsayad. "Steel plates faults diagnosis with data mining models." Journal of Computer Science 8.4 506, 2012.

[11] Kazemi, Mohammad Ali Afshar, Sima Hajian, and Neda Kiani. "Quality Control and Classification of Steel Plates Faults Using Data Mining." Applied Mathematics & Information Sciences Letters 6.2, 59-67, 2018.

[12] https://simple.wikipedia.org/wiki/Carbon_steel

[13]https://en.wikipedia.org/wiki/Universal_testing_machine