Predict Failures in Production Lines

A Two-stage Approach with Clustering and Supervised Learning

Darui Zhang, Bin Xu, Jasmine Wood
International Center for Automotive Research
Clemson University
Greenville, USA
e-mail: daruiz@g.clemson.edu

***Abstract*— The implementation of advanced technologies in manufacturing has created large amounts of data. The data can be utilized to create predictive models for quality control, which allows manufacturers to produce higher quality products at a lower cost. Bosch has provided a large-scale data set of a production line and hosted a challenge on Kaggle aiming to predict the manufacturing failures using the anonymized features. We proposed a two-stage method first to cluster the data into groups based on the manufacturing process and then use supervised learning to predict the failed product in each cluster. This approach reduces the sparsity of the data set. Various algorithms were compared. The random forest algorithm achieved the highest performance score and was chosen as the final model.**

Keywords—manufacturing; quality control; clustering; supervised learning

I. Introduction

Quality control has long been the heart of manufacturing. The introduction of more advanced sensor technologies, Radio-Frequency Identification (RFID), and the Internet of Things (IoT) [1], allows for data collection at every point of the manufacturing process. The rapidly increasing availability of data has created a new paradigm in product quality control. The challenges of big data come in volume, variety, and velocity [2]. How to turn data into useful information has been the key to increasing the productivity and eventually bring the high quality and low-cost product to the end user.

Quality engineers have been leveraging data to improve product quality for decades [3]. The traditional approach often relies on extensive domain knowledge of the process to identifying the critical features [4]. A Bayesian network approach was proposed for manufacturing process monitoring [5]. Each part was represented by the Bayesian network model, and parts models were combined to form a process model. A fault diagnosis method for machinery was implemented using Principal Component Analysis (PCA) to reduce the data dimension and decision tree for diagnosing [6]. A tool breakage detection method is presented in the study [7] using the Support Vector Machine (SVM) algorithm and the data from multiple sensor signals such as cutting forces and power consumptions. A random forest algorithm and wavelet packet decomposition (WPD) approach was developed for gearboxes fault diagnostics [8].

The recent trend has shifted to harnessing the power of large-scale data set using machine learning and reducing the effort of manual feature engineering. To cope with large data sets, clustering and PCA have commonly been used to reduce data size and select important features [9][10][11]. In this paper, a two-stage approach is proposed aiming to predict failed products using the large-scale anonymized features.

The rest of the paper is organized as follows: Section II explores the Bosch production line data set. Section III presents an overview of product failure prediction process. Section IV describes the clustering stage, which is followed by the supervised learning stage in Section V. The final model and is presented in Section VI, and future work is discussed in Section VII.

II. DATA EXPLORATION

A. Production Line Dataset

Bosch hosted a challenge on Kaggle aiming to predict the failed products in a production line [12]. The production line dataset is one of the largest public manufacturing datasets. Each observation represents a product, which moves through a production line. The features are anonymized. The naming of the feature follows the convention of "L#_S##_F####", which represents the line, station, and feature number. The ground truth of whether a product is a failure is provided as a binary class, with 1 representing failure and 0 representing pass. The goal is to use the features to predict the occurrence of a failed product.

The Bosch dataset includes numerical, categorical and time data. The categorical data is extremely sparse (more than 99%) and thus not included in the paper. The time data required the label of the preceding data, which may not be availed in a production line and therefore is not considered in the scope of this paper. Only the numerical dataset is used in this paper. A summary of the dataset is shown as follows:

- Number of features: 968.
- Number of observations: 1183747
- Percentage of a failed product: 0.58%
- Percentage of missing values: 78.5%

Several challenges are present in this dataset:

- Large scale: the dataset is large with about one thousand features and more than one million observations. And thus, the algorithms need to be scalable with a large dataset.
- Highly imbalanced: the amount of passed and failed products are highly imbalanced because only a small proportion of the products are failures. And thus, the machine learning algorithms need to be able to handle highly imbalanced data.

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

- Lack of domain knowledge: the features are anonymized. Therefore we do not have domain knowledge of the manufacturing process. The importance of features needs to be discovered by the algorithms.
- High sparsity: the dataset aggregates features from all the stations in the production line. Since many stations serve similar purposes, a product only passes through a fraction of the stations. As a result, a large portion of the features has missing values, as shown in Fig. 1. The frequency represents the number of features. Most of the features endure missing values. And nearly 450 features miss more than 95% of all the values.

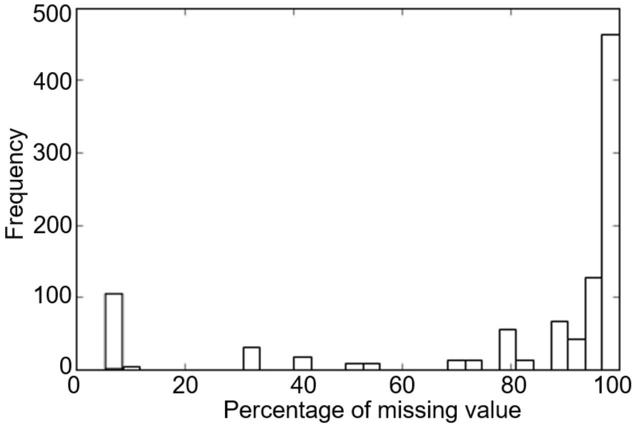


Fig. 1. The distribution of features with missing values

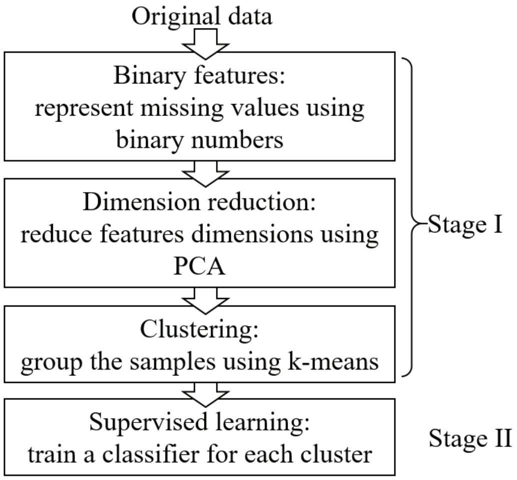
The high sparsity in the dataset brings problems. It introduces noises when replacing the missing values. And, it often creates a larger data size which slows down the machine learning algorithm. We propose a two-stage approach to reduce the sparsity of the dataset. In the first stage, clustering will be conducted to divide data into similar process groups. In the second stage, supervised learning will be used to predict the failed product in each cluster.

III. Product Failure Prediction Overview

The prediction process is conducted in two stages as shown in Fig.2

- Stage I clustering: this step clusters data with similar processes together into process groups. In every cluster, the empty and constant features are deleted to reduce the data size.
- Stage II supervised learning: this step uses supervised learning to predict the failed products. Each cluster is treated as an independent dataset and has its own classifier. During the prediction step, the data is first classified with regard to which clusters it belongs to. Then the classifier in the corresponding cluster is used to predict whether the data is a failed product or not.

The advantage of the two-stage approach is the data within each cluster is similar in their manufacturing process. The clustering process reduces the data size by eliminating irrelevant features, which speeds up the supervised learning step.



Original data

Binary features:
represent missing values using binary numbers

Dimension reduction:
reduce features dimensions using PCA

Clustering:
group the samples using k-means

Stage I

Supervised learning:
train a classifier for each cluster

Stage II

Fig. 2. Schematic flowchart of the two-stage prediction approach

A. Data Preprocess

The practice of three-fold cross-validation was followed. Before the data set was processed, it was randomly shuffled and divided into training, cross-validation, and test sets. The training and the cross-validation sets were used for building and evaluating the models. The test set was used only for evaluating the final model. The percentage of the three sets out of the total dataset are as follow:

- Training set: 50%
- Cross-validation set: 25%
- Test set: 25%

IV. Stage I: Clustering

The goal of this step is to cluster data into groups by manufacturing process. The values of the features are first converted into binary, where 1 indicates a value and 0 a missing feature.

The dataset contains 968 features. However, many of the features are correlated because they are measured in the same production stations or production lines. The Principal Component Analysis (PCA) is conducted to reduce the dimension of the features. The PCA transforms the original correlated features into linearly uncorrelated principal components in reduced dimension, while preserving the largest variation [13].

The variance explanation rate of the PCA is shown Fig.3. Although the dataset has about 1000 features, the processes are largely correlated. 95% of the variance can be explained by 22 principal components. The reduced dimension is chosen to be two for the purposes of data visualization.

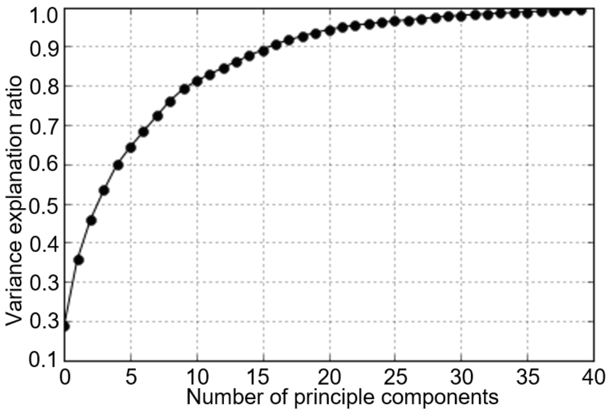


Fig. 3. Variance explained ratio with different numbers of principle components

The next step is to use the unsupervised algorithm to divide the data into clusters. The K-mean algorithm [14] is selected. The algorithm iteratively calculates the distance between the data to centroids and assigns the data to the nearest centroid, until it converges to a local optimum.

The value of K is determined by measuring the inertia within the clusters. As shown in Fig. 4, the mean inertia decreases rapidly at first, then the rate of decrease slows after six clusters. Therefore, the number of clusters is chosen to be six.

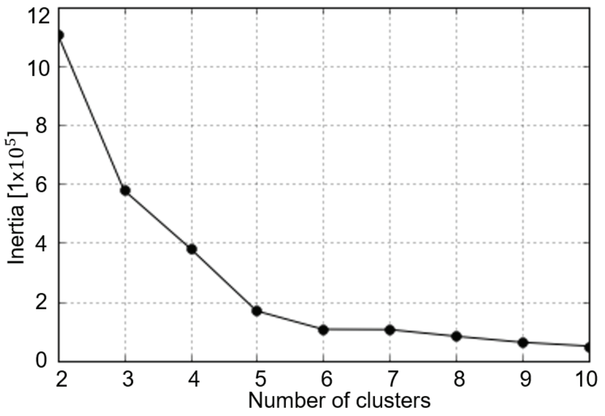


Fig. 4. Inertia within the clusters for various K

Fig. 5 shows the scatterplot of the data in the first two principal dimensions. Each dot represents a unique process, which involves a different combination of features. The transparency of the data indicates the amount of data in each process. The darker the color, the more data is in the process. The centroids of the clusters are shown as the cross marks. The regions of the different clusters are denoted with different colors.

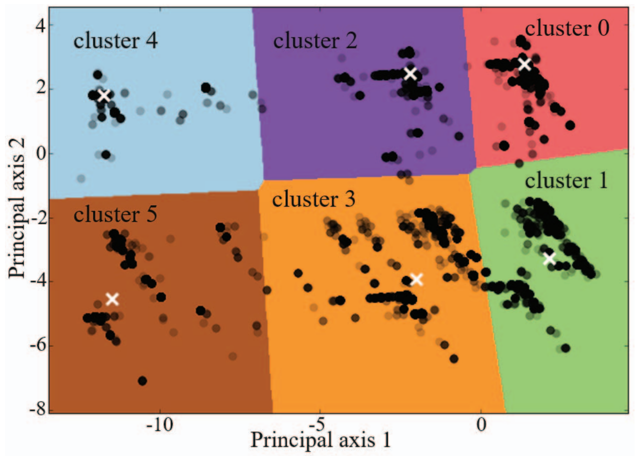


Fig. 5. Clusters of the process groups in the dimensions of the first two principal components

The features which have all missing or constant values in each cluster are eliminated. 15.8% of the total dataset is removed. The properties of each cluster are shown in Table I. The amount of data in each cluster varies greatly. The first two clusters consist of nearly 70% of the total data.

TABLE I. PROPERTIES OF THE CLUSTERS

Cluster #	Centroid	Number of samples	Number of features
0	1.37, 2.78	230831	813
1	2.14, -3.28	177562	882
2	-2.19, 2.48	88781	692
3	-1.99, -3.93	65106	874
4	-11.7, 1.81	17756	737
5	-11.49, -4.55	11837	697

The following step is to replace the values which are still missing. Since the missing value is caused by different manufacturing processes and is not missing at random, they are replaced with the special value “-1”.

V. STAGE II: SUPERVISED LEARNING

After the first step, the data is divided into several clusters. The next step is to utilize the supervised learning algorithms to predict the failed product in each cluster. Each cluster is treated as an independent dataset, where different algorithms and parameters can be chosen independently of that of other clusters.

A. *Performance Metrics*

The performance metrics need to accommodate for the highly imbalanced data. The Matthew's Correlation Coefficient (MCC) was chosen as the metric, which is defined as:

$$MCC = \frac{TP\times TN - FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \qquad (1)$$

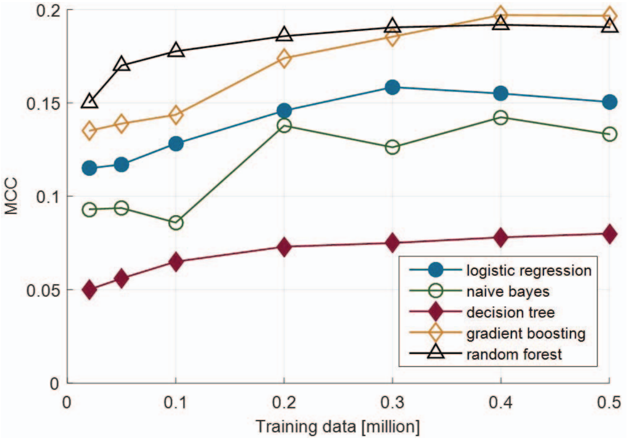
Where TP, TN, FP, FN represent True Positive, True Negative, False Positive and False Negative values in the confusion matrix. The MCC score usually ranges from zero (random guess) to one (perfect classification). MCC is calculated directly from the binary prediction results. A threshold is required to convert the probability into the binary result. The threshold was optimized toward the highest MCC score on the cross-validation set.

The Area Under the Receiver Operating Characteristic Curve (AUROC) [15] is used as an additional metric for the robustness of the algorithm. The classification results are provided as the probability of whether or not the data belongs to one class. The ROC curve is created by plotting the true positive (TP) rate against the false positive (FP) rate at various threshold settings. The AUROC usually ranges from 0.5 (random guess) to one (perfect classification.).

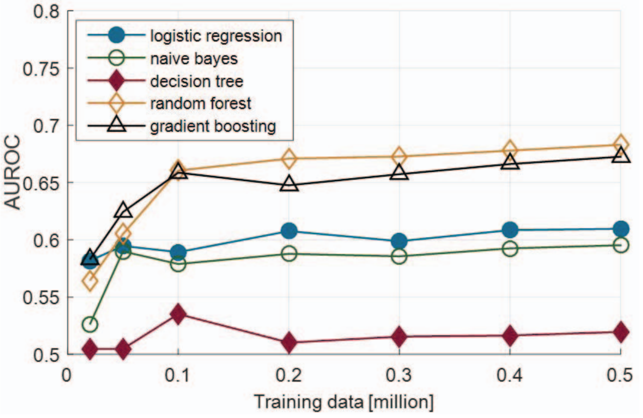
B. *Algorithm Comparison*

Choosing the supervised learning algorithm is based on performance and computation time. We compared the several widely-used algorithms. The machine learning algorithms were implemented using the Scikit-Learn package [16].

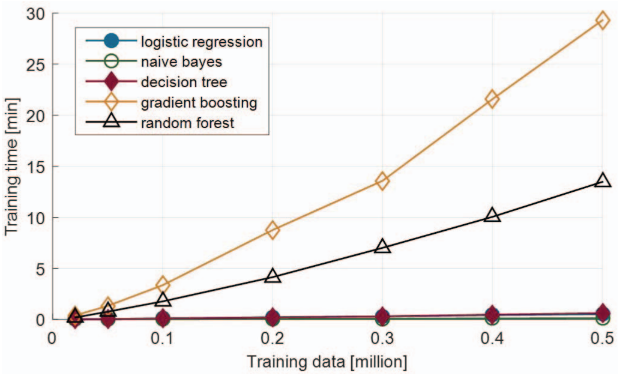
MCC and AUROC are used as the performance metric. The training time is used as the metric of computation time. The results of the total data set are shown in Fig. 6. Both the MCC and AUROC scores increase as the training data size increases; however, the rate of increase slows down after 100,000 observations. The ensemble methods (random forest and gradient boosting) present better performance than those of simple classifiers (logistic regression, naïve Bayes, and a decision tree). The computation time of the ensemble methods increases linearly with respect to the size of the data, which is desired for the large dataset.



(a)



(b)



(c)

Fig. 6. Algorithm comparison (a) MCC (b) AUROC (c) training time

The random forest algorithm [17] is chosen by considering both the performance and training time. The algorithm uses multiple decision trees to construct the classifier to overcome the problem of overfitting by an individual decision tree.

C. *Feature Selection*

The original dataset contains hundreds of features. However, the features contribute differently to predicting the target and most of the features are irrelevant. To further reduce the data size and speed up the supervised learning algorithm, the features with less importance are truncated.

The feature importance can be measured by assessing the frequency of the feature being used as a node in a decision tree. The more frequently a feature is used as a node in a decision tree, the more important that feature is. The ensemble method, like the random forest, is capable of ranking the feature importance by averaging the feature importance of each tree.

Fig. 7 shows the relative feature importance ranked by the random forest classifier for the data in cluster 1. The features are selected until 95% of the cumulative sum of the feature importance has been reached. The same process is conducted for the other five clusters. In addition, the features with relatively high importance often indicate which station is problematic and thus can provide insight for quality assurance.

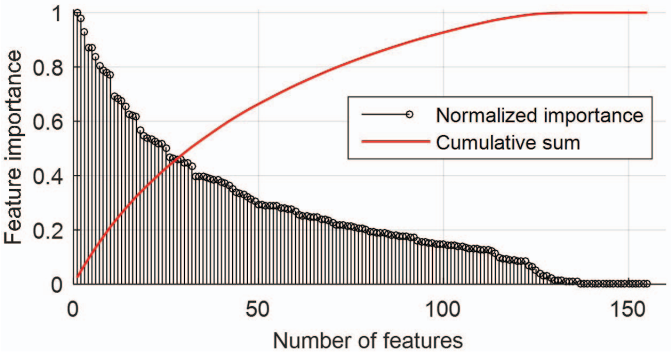


Fig. 7. Feature importance

VI. Results of the Final Model

The practice of three-fold cross validation is followed to train the classifiers. The parameters in the random forest classifiers are optimized using the grid search. The test set is utilized for evaluating the final model.

During the prediction step, the algorithm first determines which cluster the data belongs to, then predicts whether the product is a failure using the classifier for the responding cluster. The final performance score of each cluster and the total data set is shown in Table II and the ROC curve is shown in Fig.8.

TABLE II. THE PERFORMANCE OF THE FINAL MODEL

Cluster #	Classifier Parameters*		MCC	AUROC
	max depth	*min_sample_leaf*		
0	25	6	0.169	0.665
1	25	6	0.308	0.736
2	20	8	0.148	0.664
3	5	6	0.145	0.672
4	20	8	0.184	0.606
5	20	5	0.179	0.568
Total	--	--	0.211	0.692

* The number of estimators in the random forest algorithm is 100 for all the classifiers

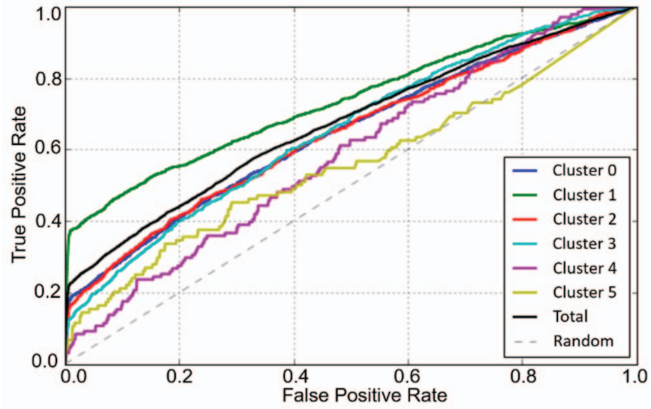


Fig. 8. ROC curve of the 6 clusters and the total data set

The AUROC score ranges from 0.5 (random guess) to one (perfect classification); the score of 0.69 is a relatively low score, which indicates the complexity of this production failure prediction problem. The slope of the ROC curve is steep in the beginning but soon flattens, which reveals that a small portion of the positive data is easy to classify, but most of the data is hard to classify. The performance of each cluster also varies highly. The data in Cluster 1 is the easiest to classify and the data in Cluster 3 is the most difficult to classify.

VII. FUTURE WORK

Large-scale manufacturing data is being generated as processes becoming more intelligent. The emergence of big data provides both the opportunity for using predictive models in quality assurance and the challenge for data processing. In this paper, we proposed a two-stage method to predict manufacturing failures in a production line. During the first step, the data is clustered into similar process groups. Then, supervised learning is applied to each cluster to predict product failure. This approach reduces the sparsity of the data set by eliminating irrelevant features.

In future work, the parameters of the classifiers can be optimized to increase performance. Meanwhile, additional models will be considered. In addition to the numerical data, the categorical data and the timestamp data can also be used to provide additional information.

ACKNOWLEDGMENT

The authors would like to thank Bosch for providing the data set and Kaggle for hosting the competition.

REFERENCES

[1] A. Zaslavsky, C. Perera, and D. Georgakopoulos, “Sensing as a Service and Big Data,” *Proc. Int. Conf. Adv. Cloud Comput.*, pp. 21–29, 2012.

[2] C. Eaton, D. Deroos, T. Deutsch, G. Laipis, and P. Zikopolos, *Understanding big data:Analytics for enterprise class hadoop and streaming data.*, vol. 11, no. 1. 2011.

[3] J. A. Harding, M. Shahbaz, Srinivas, and A. Kusiak, “Data Mining in Manufacturing: A Review,” *J. Manuf. Sci. Eng.*, vol. 128, no. 4, p. 969, 2006.

[4] A. K. Choudhary, J. A. Harding, and M. K. Tiwari, “Data mining in manufacturing: A review based on the kind of knowledge,” *J. Intell. Manuf.*, vol. 20, no. 5, pp. 501–521, 2009.

[5] E. Wolbrecht, B. D’ambrosio, R. Paasch, and D. Kirby, “Monitoring and diagnosis of a multistage manufacturing process using Bayesian networks,” *Artif. Intell. Eng. Des. Anal. Manuf.*, vol. 14, no. 1, pp. 53–67, 2000.

[6] W. Sun, J. Chen, and J. Li, “Decision tree and PCA-based fault diagnosis of rotating machinery,” *Mech. Syst. Signal Process.*, vol. 21, no. 3, pp. 1300–1317, 2007.

[7] S. Cho, S. Asfour, A. Onar, and N. Kaundinya, “Tool breakage detection using support vector machine learning in a milling process,” *Int. J. Mach. Tools Manuf.*, vol. 45, no. 3, pp. 241–249, 2005.

[8] D. Cabrera, F. Sancho, R. V. Sánchez, G. Zurita, M. Cerrada, C. Li, and R. E. Vásquez, “Fault diagnosis of spur gearbox based on random forest and wavelet packet decomposition,” *Front. Mech. Eng.*, vol. 10, no. 3, pp. 277–286, 2015.

[9] Y. Li, “Building a Decision Cluster Classification Model by a Clustering Algorithm to Classify Large High Dimensional Data with Multiple Classes.” Diss. Hong Kong Polytechnic University, 2010.

[10] D. Feldman, M. Schmidt, and C. Sohler, “Turning Big data into tiny data : Constant-size coresets for k -means , PCA and projective clustering,” *Proc. Twenty-Fourth Annu. ACM-SIAM Symp. Discret. Algorithms*, pp. 1434–1453, 2013.

[11] A. Malhi and R. X. Gao, “PCA-based feature selection scheme for machine defect classification,” *IEEE Trans. Instrum. Meas.*, vol. 53, no. 6, pp. 1517–1525, 2004.

[12] [Online]. Available: https://www.kaggle.com/c/bosch-production-line-performance.

[13] C. Bishop and N. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2007.

[14] D. Arthur and S. Vassilvitskii, “k-means++: The Advantages of Careful Seeding,” vol. 8, pp. 1–11.

[15] C. D. Brown and H. T. Davis, “Receiver operating characteristics curves and related decision measures: A tutorial,” *Chemom. Intell. Lab. Syst.*, vol. 80, no. 1, pp. 24–38, 2006.

[16] L. Buitinck, G. Louppe, and M. Blondel, “API design for machine learning software: experiences from the scikit-learn project,” *arXiv Prepr. arXiv* , pp. 1–15, 2013.

[17] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.