Machine learning for sensor-based manufacturing processes

Conference Paper · September 2017

DOI: 10.1109/ICCP.2017.8116997

4 authors:

Dorin Moldovan
Universitatea Tehnica Cluj-Napoca
35 PUBLICATIONS **73** CITATIONS
SEE PROFILE

Tudor Cioara
Universitatea Tehnica Cluj-Napoca
102 PUBLICATIONS **647** CITATIONS
SEE PROFILE

Ionut Anghel
Universitatea Tehnica Cluj-Napoca
100 PUBLICATIONS **646** CITATIONS
SEE PROFILE

Ioan Salomie
Universitatea Tehnica Cluj-Napoca
193 PUBLICATIONS **1,195** CITATIONS
SEE PROFILE

Machine Learning for Sensor-Based Manufacturing Processes

Dorin Moldovan, Tudor Cioara, Ionut Anghel, Ioan Salomie
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
Emails: { Dorin.Moldovan, Tudor.Cioara, Ionut.Anghel, Ioan.Salomie} @cs.utcluj.ro

***Abstract*—The increasing availability of relevant information, events and constraints in the environment of the modern factories due to deployment of IoT sensor technologies on the production line has led to an "explosion" in contextual big data. At the same time the advancements in the machine learning field from the last years opened new approaches for the analysis of the manufacturing processes datasets that are characterized by noisy data, a large number of features and an imbalanced classification of the samples. In this paper we investigate the applicability and the impact of machine learning techniques for managing production processes considering the data from a semiconductor manufacturing process (SECOM dataset). We have applied algorithms such as Boruta and MARS for the selection of the most relevant features and the Random Forest and the Gradient Boosted Trees for the samples classification. The results show better values for precision when the features are selected using Boruta and MARS rather than PCA and better values for accuracy when the data is unsampled and classified using Random Forest and Logistic Regression rather than Gradient Boosted Trees.**

***Index Terms*—Machine Learning, Features Selection, Classification, Imbalanced Datasets, SECOM.**

I. Introduction

Driven by globalization, competing markets, tightened compliance regulations, rising complexity due to mergers and acquisitions as well as fast-changing customer requirements, today's business world is continuously changing. The deployment of smart components and novel IoT sensors at shop floor level has made the management of manufacturing processes increasingly complex. This is due to the escalating number of registered events taking place inside factories and outside their business environment across the whole supply chain. In this highly-competitive and frequently-changing environment manufacturing processes are exposed to the uncertainties generated by rapid changes in suppliers, logistics providers and materials procurement as well as in market conditions and customer requirements demanding on the fly adjustment of production and accelerating the product time to market. In response to these pressuring conditions factories have to rapidly adjust, adapt, optimize and control their production processes always considering external (outside factory) and internal (shop floor level) stimuli. The manufacturing companies have to dissolve themselves from the rigid structures of their manufacturing processes so that they are able to react dynamically to the changes of the context situations (both outside and inside the factory) through the utilization of advanced sensing, embedded systems, smart control and machine learning techniques.

Big data and machine learning [1] open new opportunities for manufacturing processes management and optimization leveraging on collecting of the distributed data, the cleaning of the data, the extraction of meaningful information from noisy data and the refreshing of the optimization insights from the flowing data in real time. The complexity of the analysis of the manufacturing processes [2] is given by the fact that the production line equipment is augmented by a large number of sensors which generate hundreds and sometimes thousands of signals. Also, the sampling rates of the sensor readings have seen important increments recently [3]. These developments led to huge amounts of data that have to be analyzed.

Several impacts of using the big data and the machine learning technologies [4] in the manufacturing processes management are the early detection of the problems related to the quality, the better detection of the defects associated to the products, the quality boosting and the performing of predictions for the product failure. The analysis of the sensors data has as benefits the avoidance of the costly human interventions and the cut of the waste. The products vendors and manufacturing companies benefit very much from the analytics which are used to optimize the semiconductor manufacturing [5]. The big data analytics can thus be used for proactive maintenance and for forecasting [6]. Some of the most important challenges in managing the manufacturing processes are the high number of dimensions, the imbalance between the pass and the fail cases and the construction of a predictive model. The outlier detection is a common characteristic of the high-dimensional data [7]. The selection of the features [8] for the high-dimensional data has to deal with many problems such as the class imbalance, the dataset shift, the incremental learning, the noisy data and the budget constraints. The curse of dimensionality, a term introduced by Richard Bellman in [9], is associated with a decrease in the generalization ability of the classification algorithms. The imbalanced datasets may appear not only in classification problems but also in regression problems [10]. One special issue associated with the imbalanced datasets is the generation of sub-optimal classification models [11]. The efficiency and the performance of the clustering algorithms [12] can be influenced very much by the imbalanced datasets.

In this paper we have addressed the above presented challenges by studying the applicability and the impact of the machine learning techniques for managing production processes considering the data from a semiconductor manufacturing process (SECOM data set). In more details: (i) we have investigated the state of the art approaches for using machine learning techniques on the SECOM data set, (ii) we have cross compared three different algorithms for relevant features selection (Boruta [13], the Multivariate Adaptive Regression Spline (MARS) [14] and the Principal Component Analysis (PCA)) and (iii) finally we have compared three sample classification algorithms (Logistic Regression (LR) [15], Random Forest (RF) [16] and Gradient Boosted Trees (GBT) [17]). The evaluation of different algorithms has been performed using five metrics [18]: the False Positive Rate (FPR), the precision, the recall, the F-measure and the accuracy.

The rest of the paper is organized as follows: Section II presents the related work, Section III proposes a sequence of machine learning techniques that can be applied on large, imbalanced and noisy datasets, Section IV presents the experimental results and Section V presents the conclusions.

II. Related Work

SECOM dataset [19] is generated by a semiconductor manufacturing process [20] consisting of more than one hundred steps and it is representative for a larger class of modern sensor based manufacturing processes. The final products that are the outcomes of the manufacturing process might fail the tests if there are defects which occur in the process. The quality of the products, the increase of the equipment uptime and the reliability are crucial and they can be improved by using appropriate classification techniques.

The dataset consists of features or variables which are collected from sensors installed on process measurement points and contains both useful information and irrelevant information such as noise or constant values. The number of samples is 1567 and each sample has 591 features. Each test point is associated with a data timestamp and is labeled with 1 to indicate a fail and -1 to indicate a pass. The null values are described using 'NaN'.

The SECOM dataset has been widely used in literature for studying the impact of three major problems related to the analysis of the data generated by hundreds of signals or sensors. The first major problem is represented by the selection of the most representative features that characterize the measurement data by taking into consideration the possible drawbacks that might exist such as the large number of features or signals, the existence of noise and the presence of irrelevant information. The second major problem is represented by the imbalance between the pass and the fail cases considering that each measurement point is characterized by either a pass or a fail. The third major problem is represented by the selection of the best classification algorithm.

In [21] the authors use several approaches for the selection of the features such as the removal of the features which have constant values, the removal of the features which have many missing values, the Chi-Square statistical analysis, the Principal Component Analysis (PCA) and the Gain Ratio (GR). The tested algorithms are the k-Nearest Neighbor (k-NN), the Logistic Regression (LR), the Naïve Bayes (NB) and the Decision Trees (DT). The metrics which are used to assess the performance of the model are the true positive rate (TPR), the precision, the F-measure and the False Positive Rate (FPR). The selection of the features is realized using the WEKA software [22], a machine learning workbench which supports many features such as data pre-processing, classification, clustering, attribute selection and data visualization.

In [23] the authors define predictive models for the detection of the equipment fault in the semiconductor manufacturing processes. The approach used by the authors consists of several steps which are applied on the SECOM dataset such as: the data preparation, the data cleansing, the features scaling, the features reduction, the features selection, the variable component analysis and the model selection.

In [24] considering the problem of data imbalance which occurs in the pass or fail classes various techniques are studied. The Synthetic Minority Oversampling Technique (SMOTE) is the proposed method for the solving of the data imbalance problem. The imbalance between the pass and the fail class is solved by oversampling the fail class. The four algorithms which are compared are Random Forest (RF), Artificial Neural Network (ANN), Logistic Regression (LR) and Decision Tree (DT).

The problem of incomplete data is the main topic of [25]. The data completion methods are derived from different types of classifiers such as the Decision Tree (DT), the Naïve Bayes (NB) and the Nearest Neighbor (NN). The missing data is replaced so that it does not cause problems in classification. The causes that lead to missing values are various and they range from the cost of the acquisition to the timeliness aspect which is characterized by the unavailability of the data even if the data is obtainable. A common approach that is used for the substituting of the missing values is the use of the mean or of the mode. Even if the SECOM dataset is very imbalanced compared to the other datasets that are used in the experiments, when the data is classified using the Naïve Bayes, both classes are evenly predicted.

In [26] the problems of class imbalance, class overlap and lack of density are investigated. The imbalance occurs when one class out-represents another class severely. The class overlap is a problem that occurs when the classes from a dataset cannot be separated linearly. The lack of density appears when the data required by a classifier is insufficient for making generalized rules. In this case the number of features of the SECOM dataset is reduced using two feature selection techniques: Principal Component Analysis (PCA) and brute force (w-SimpleCART).

Table I summarizes the main machine learning techniques used in [21], [23]–[26] for the analysis of the SECOM dataset.

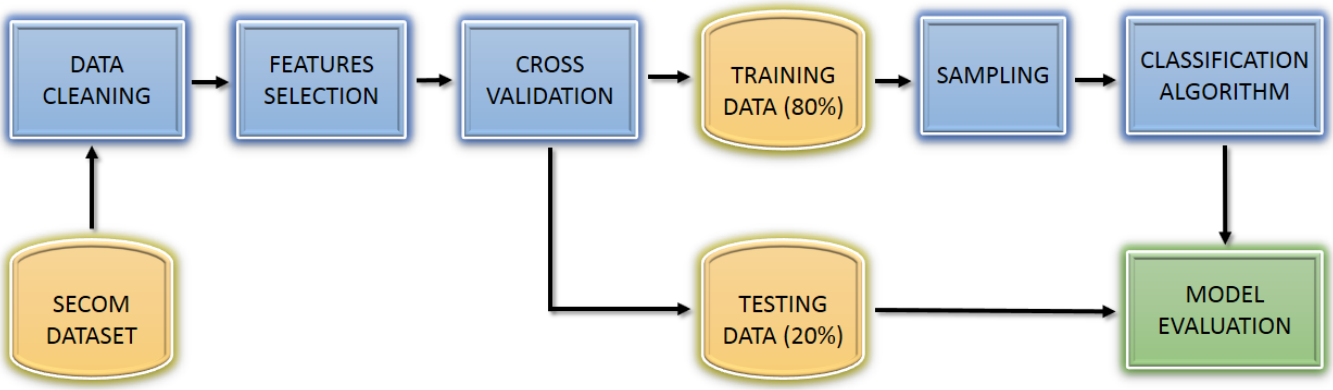

Fig. 1. Machine Learning Steps for the Analysis of the SECOM Dataset

TABLE I
A LITERATURE OVERVIEW OF THE MACHINE LEARNING TECHNIQUES USED FOR THE ANALYSIS OF THE SECOM DATASET

Data Cleaning	Features Removal [21], Data Cleansing and Features Scaling [23], Substitution of the Missing Values [25]
Features Selection	Chi-Square and Gain Ratio [21], Principal Component Analysis [23], [26], Correlation Analysis and Variable Component Analysis [23], Brute Force (w-SimpleCART) [26]
Sampling	Rare Case Boosting [21], Synthetic Minority Oversampling Technique (SMOTE) [26], Random Over-Sampling [26], Bordeline SMOTE [26], Condensed Nearest Neighbor (CNN) [26], Neighborhood Cleaning Rule [26], K-medoid [26]
Classification Algorithms	k-Nearest Neighbor [21], [23], [25], Logistic Regression [21], [23], [24], Naïve Bayes [21], [23], [25], Decision Trees [21], [23], [24], Support Vector Machine [23], Artificial Neural Network [23], [24]
Evaluation Metrics	True Positive Rate, Precision, F-measure and False Positive Rate (FPR) [21], Matthews Correlation Coefficient [23], Receiver Operating Characteristic Area [23], Precision and Recall Curve Area [23]

III. RESEARCH APPROACH

The problem of using machine learning for extracting meaningful information from the manufacturing processes is a complex one and has to deal with many existing challenges such as the missing data, the existence of the noise, the very large number of features, the imbalance of the samples classes, and so on. Our proposed sequence of steps to investigate the above problems using the SECOM dataset is described in Figure 1. The next subsections present details for each step.

A. Data Cleaning

The objective of the *Data Cleaning* is to prepare the data for the processing. The transformations that are performed in the *Data Cleaning* phase are the removal of the features that have a large number of missing values, the replacement of the undefined values which are notated with 'NaN' with numerical values and the normalization of the values so that they have values from the interval $[0, 1]$.

The *features removal* has as main objective the removal of those features for which more than 55% of the data is missing. Some possible causes for missing data are the different rates at which the measurements from the sensors are transmitted or the characteristics of the transmission media. The threshold 55% for the removal of the features is used in [21] and [23], while the threshold 60% is used in [24]. Figure 2 presents the distribution of the features considering the number of missing values. This step removes 24 features from the initial 591 features and thus only 567 features will be considered in the next steps.

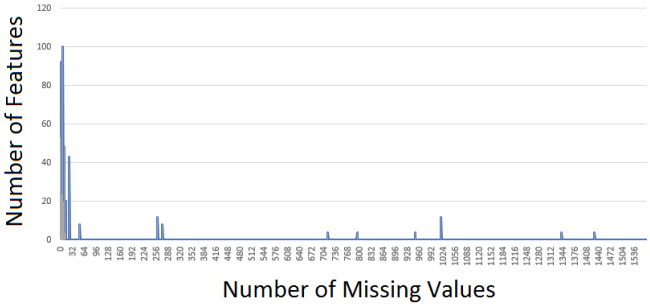

Fig. 2. Features Distribution by the Number of Missing Values

The *undefined values* are replaced by using the mean heuristic and the nearest neighbor heuristic. If the 'NaN' value is between two numerical values then the 'NaN' value is replaced by their mean, otherwise if the 'NaN' value is surrounded by a 'NaN' value and by a numerical value then the 'NaN' value is replaced by the numerical value.

The removal of the *constant values* deals with the elimination of those features from the dataset that have constant values for all the samples from the training data. After the removal of 116 features which consist of constant values, the number of the features that will be considered in the next steps becomes 451.

Finally the data is normalized so that it takes values from the interval $[0, 1]$ using the following equation:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where x_{norm} is the normalized value of x, x_{min} is the minimum value that x can have and x_{max} is the maximum value that x can have.

B. Features Selection

The objective of the *Features Selection* is to eliminate those features that do not have any influence on the classification results. In this subsection we cross compare three approaches: Boruta algorithm, MARS algorithm and Principal Component Analysis (PCA). As far as we know, no other paper has reported the use of the Boruta and the MARS algorithms on the SECOM dataset. We have compared these algorithms with the PCA, a widely used approach which appears in at least three other references. The impact of the features selection algorithms on the performance of the classification algorithms is detailed in Section IV. A comparison of the three features selection algorithms is presented in Table II.

TABLE II
COMPARISON OF THE FEATURES SELECTION TECHNIQUES

Features Selection Algorithm	Number of Selected Features	Characteristics
Boruta	22	by default it uses the Random Forest (RF) algorithm
MARS	10	it is a form of regression analysis
PCA	111	it uses orthogonal linear transformations

Boruta Algorithm [13] identifies the relevant variables by using a classification algorithm which is wrapped around the Random Forest classification algorithm. The package Boruta is implemented in R, a language which is used widely in statistics and data mining.

Figure 3 shows the results that are obtained using the Boruta algorithm which has performed 76 iterations in 7.3 minutes. From the total of 451 features, 22 were confirmed as important while 428 were confirmed as unimportant or potentially important. The feature which does not appear in the results $(451 - 22 - 428 = 1)$ is the label.

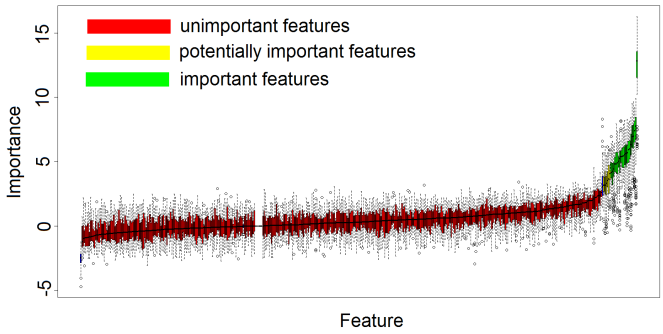

Fig. 3. Features Selection (Boruta)

The Multivariate Adaptive Regression Spline (MARS) [14] selects the most important features based on the Residual Sum of Squares (RSS) (an amount which indicates the amount of difference between an estimation model and the actual data), the Generalized Cross Validation (GCV) (a modified version of cross-validation that uses a formula for approximating the error that is determined by the leave-one-out validation) and the number of sets (the number of model subsets which include the variable). Figure 4 presents the selected features after running MARS in R. The number of selected features is 10.

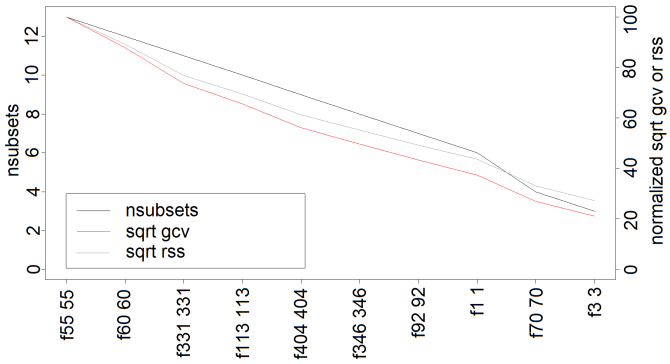

Fig. 4. Variable Importance (MARS)

The Principal Component Analysis (PCA) [28] converts a set of uncorrelated features or variables into a set of correlated ones which are called principal components. Three approaches can be used for selecting the number of components: the selection of the components that have the eigenvalues greater than 1, the selection of the factors which explain at least 80% of the variance and the selection of the factors until a break occurs in the graph. The method used in the experiments is the first one. The number of principal components that have the eigen values greater than 1 is equal with 111, thus the PCA algorithm will use the first 111 principal components. Figure 5 (TOP) presents the Variables Factor Map, a map that describes the projections of the observed variables on the plane which is described by the first two principal components, while Figure 5 (BOTTOM) presents the Individuals Factor Map, a map that describes the principal component scores of the individuals projected on the first two principal components.

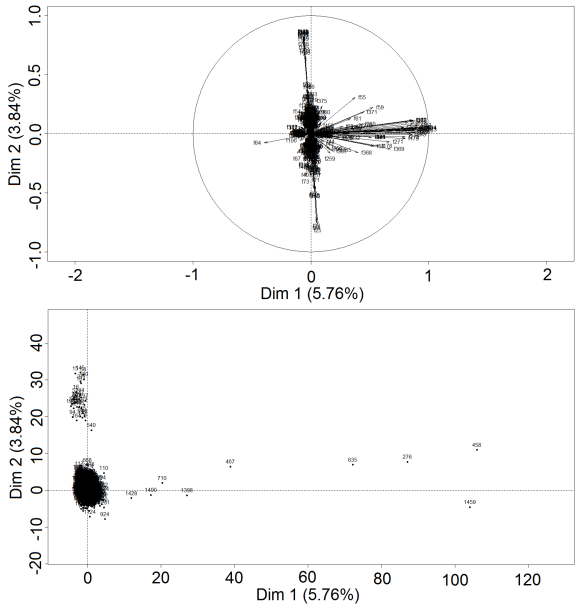

Fig. 5. (TOP) Features Factor Map (PCA) and (BOTTOM) Individuals Factor Map (PCA)

C. Cross Validation

The objective of *Cross Validation* [27] is to split the data in order to estimate the performance of the tested algorithms. A part of the data (training) is used for training the algorithms while the remaining part (validation) is used for estimating the performance of the algorithms. For Cross Validation we have used a 5-fold approach in which the data is split randomly in 5 subsets that have the same number of samples. The steps described in the next subsections are repeated five times and each time the testing data will be one distinct fold from the set of the 5 folds and the training data will consist of the remaining 4 folds. The training data is processed further in the sampling phase and the classification algorithm phase. The obtained model is evaluated using the testing data and the following metrics: the accuracy, the F-measure, the recall, the precision and the False Positive Rate (FPR). The experimental results are detailed in Section IV.

D. Sampling

The objective of the *Sampling* phase is to treat the imbalance problem between the samples of the minority class and the samples of the majority class. Out of the 1567 samples from the SECOM dataset, 104 are fails and 1463 are passes. There is an evident imbalance between the fails and the passes. The imbalance ratio is approximately $1:14$ (see Figure 6). The sampling is performed on the samples from the training dataset. The testing dataset is left unchanged in order to avoid problems such as overfitting.

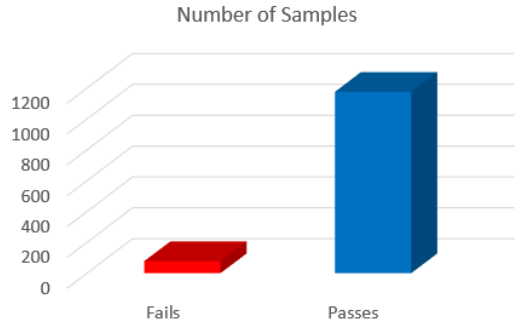

Fig. 6. Initial Distribution of the Fail Samples and the Pass Samples

The first approach for improving the imbalance ratio is the random *Undersampling of the Majority Class*. The number of instances of the majority class becomes equal with the number of instances of the minority class. The filter used in the undersampling is the SpreadSubsample filter from WEKA [29]. The second approach is the *Oversampling of the Minority Class*. The oversampling of the minority class is realized by using the Synthetic Minority Oversampling Technique (SMOTE). The SMOTE filter is applied several times until the dataset is balanced. After the oversampling of the minority class, the order of the samples is randomized in order to avoid the overfitting.

E. Classification Algorithms

The training data is used as input for three *Classification Algorithms* which are briefly described below. The Logistic Regression (LR) algorithm [15] was used in at least three research papers that analyzed the SECOM dataset. It is one of the simplest classification algorithms. The dependent variable is binary and it can be either 1 or 0. The LR applied in the experimental results section uses the Limited-Memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) optimization algorithm in order to use only a limited amount of the computer memory. We have used LR as a reference algorithm to evaluate the performance of the other two classification algorithms tried in this paper, the Random Forest (RF) and the Gradient Boosted Trees (GBT) which to the best of our knowledge were not applied before on the SECOM dataset.

The Random Forest (RF) [16] algorithm is an ensemble classifier which is composed of decision trees. Starting from the same dataset, the idea behind the RF is to construct many decision trees. The trees are constructed using a divide-and-conquer approach. The adjustable parameters that are used to initialize the RF algorithm and their values are presented in Table III. A detailed description of the parameters can be found in [30].

TABLE III
RANDOM FOREST PARAMETERS

Parameter	Value	Description
number of classes	2	the number of classification classes
number of trees	20	a parameter that describes how many trees are in the forest
features subset strategy	"auto"	how many features to use as candidates for splitting
impurity	"giny"	a parameter that describes the homogeneity of the labels
maximum depth	20	the maximum depth of the trees from the forest
maximum number of bins	100	the maximum number of bins that are used to discretize the features

The Gradient Boosted Trees (GBT) are ensembles of Decision Trees (DTs). The DTs are trained iteratively by the GBT in order to minimize a loss function. Even if the GBT have fewer parameters to tune than the RFs, a study performed in [17] and which studied the performance of the supervised learning algorithms concluded that the GBT are the best learning algorithm. The adjustable parameters and their values are presented in Table IV.

TABLE IV
GRADIENT BOOSTED TREES PARAMETERS

Parameter	Value	Description
number of classes	2	the number of classification classes
number of iterations	3	each iteration creates a tree
maximum depth	5	the maximum depth of the trees

IV. EXPERIMENTAL RESULTS

The metrics that are used for the evaluation of the results are briefly described next:

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

The TP, FP, FN and TN represent the confusion matrix and are described in Table V.

TABLE V
CONFUSION MATRIX

	Actual Class = 1 (Fail)	Actual Class = -1 (Pass)
Prediction Class = 1 (Fail)	True Positive (TP)	False Positive (FP)
Prediction Class = -1 (Pass)	False Negative (FN)	True Negative (TN)

For each algorithm there are $12 = 4 \times 3$ possible configurations that are evaluated depending on how the features are selected and how the sampling is performed. The number of features that is used in the experiments varies depending on the features selection method that is chosen. Table VI presents the number of features used by each features selection method.

TABLE VI
FEATURES SELECTION

Approach	Number of Features
Worst Case	450
Boruta	22
MARS	10
PCA	111

Table VII describes the sampling methods used in testing.

TABLE VII
SAMPLING METHOD

Sampling Method	Description
Unsampled	the number of samples is unchanged
Undersampled Majority Class	the number of samples of the majority class is reduced to 78
Oversampled Minority Class	the number of samples of the minority class is increased to 1176

A summary of the experimental results is presented in Figure 7 and in Figure 8. In the first figure the perspective is the approach in which the features are selected while in the second figure the perspective is the classification algorithm. The best model is the one that has the highest value for the precision and the smallest value for the False Positive Rate (FPR).

When the data is unsampled, the best value for the precision (0.89440) is obtained when the chosen algorithm is the LR and the features are selected using the Boruta algorithm. When the majority class is undersampled, the precision is improved and the best value (0.91541) is obtained when the data is classified using the RF and the features are selected using the Boruta algorithm. The best value for precision that corresponds to the oversampling of the minority class is (0.90506) and this value is obtained when the LR algorithm is used and the features are selected using the MARS algorithm.

The value of the FPR when the data is unsampled has the smallest value (0.75597) when all the features are used and the

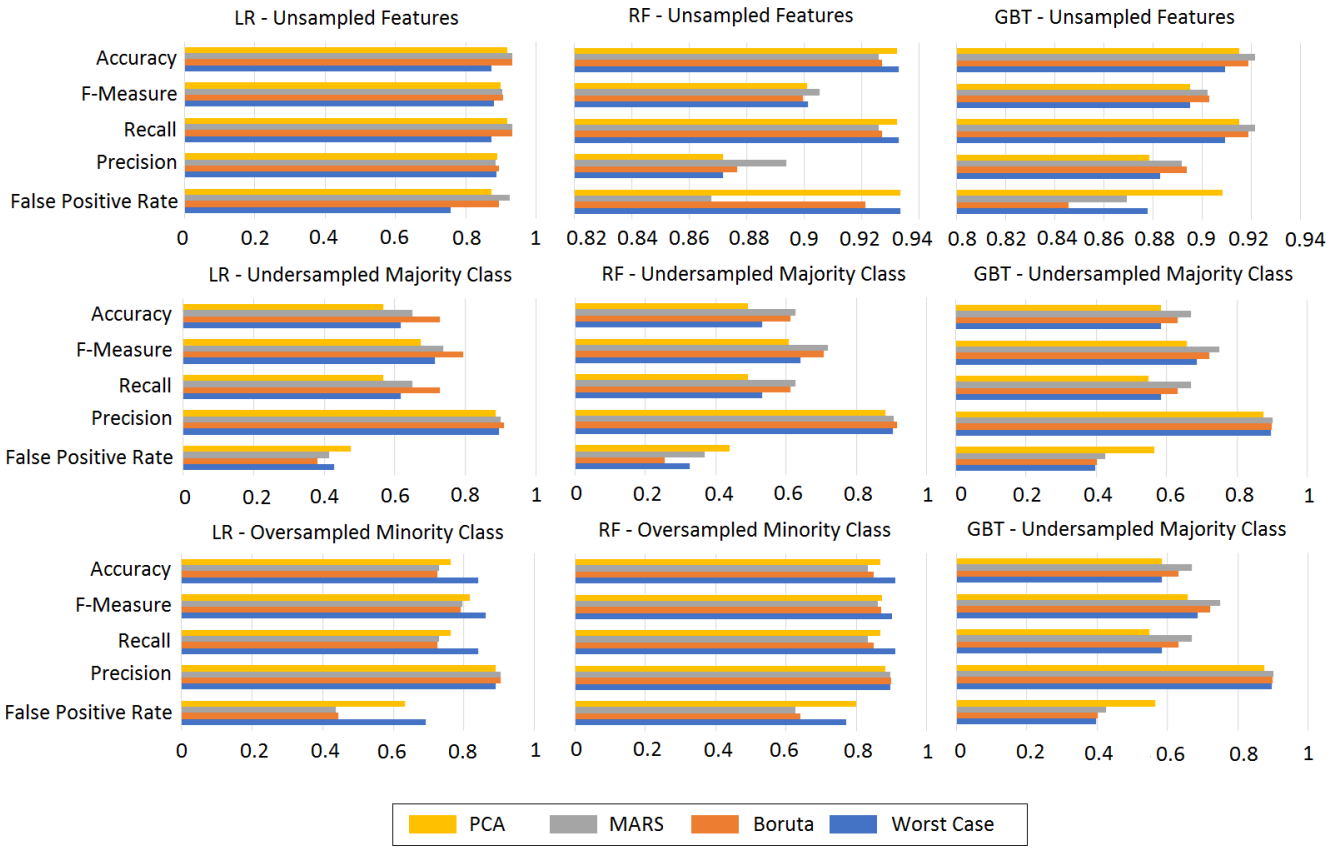

Fig. 7. Experimental Results Summary - Features Selection Perspective

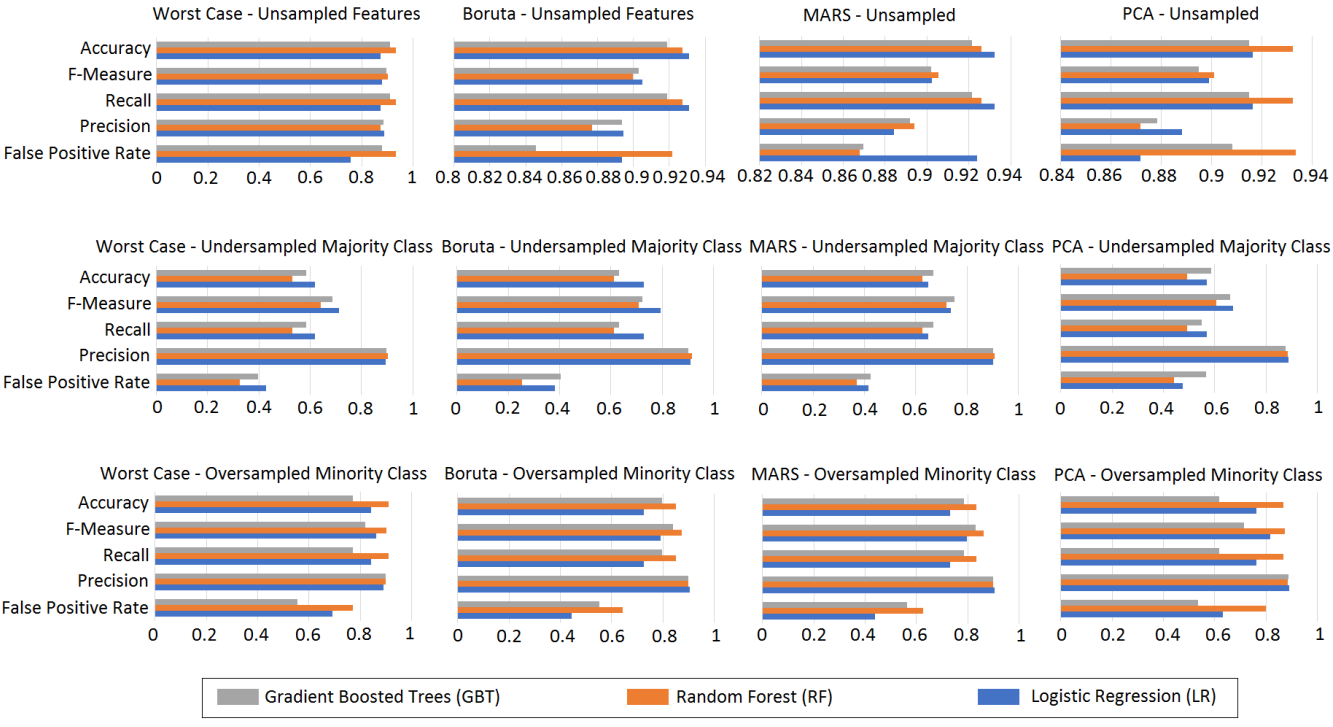

Fig. 8. Experimental Results Summary - Classification Algorithm Perspective

applied classification algorithm is the LR. When the majority class is undersampled, the smallest value of the FPR (0.25475) is obtained when the features are selected using the Boruta algorithm and the data is classified using the RF algorithm. Finally, when the minority class is oversampled, the smallest value of the FPR (0.43741) is obtained when the features are selected using the MARS algorithm and the data is classified using the LR.

V. Conclusions

In this research paper we studied and analyzed machine learning techniques that can be applied on the datasets that are characterized by noisiness, high-dimensional feature space and imbalance in classification. The dataset used in the experiments is the SECOM dataset, a dataset available online. The contributions of this research paper are the critical study of the literature approaches that use the SECOM dataset as experimental support, the selection of the features using two new algorithms that were not used in the other approaches that study the SECOM dataset, the MARS and the Boruta algorithms, and the classification of the data using two ensemble methods from Apache Spark, the Gradient-Boosted Trees (GBT) and the Random Forest (RF). The original dataset was oversampled and undersampled using the WEKA software and the final model was evaluated using five metrics: the False Positive Rate (FPR), the precision, the recall, the F-measure and the accuracy. Under the proposed configuration, the experimental results show that the best model is obtained when the majority class is undersampled, the features are selected using the Boruta algorithm and the data is classified using the RF algorithm.

Acknowledgment

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI UEFISCDI, project number PN-III-P2-2.1-BG-2016-0080, within PNCDI III.

References

[1] H. Bauer, P. Ranade, and S. Tandon, "Big data and the opportunities it creates for semiconductor players," *McKinsey on Semiconductors*, pp. 46–55, 2012.

[2] A. A. U. Haq, K. Wang, and D. Djurdjanovic, "Feature construction for dense inline data in semiconductor manufacturing processes," *IFAC-PapersOnLine*, vol. 49, no. 28, pp. 274–279, 2016.

[3] J. Dietz and C. Knepfler, "New controller extends lifetime of 200mm tools," *Nanochip Fab Solutions*, vol. 8, no. 2, pp. 14–17, 2013.

[4] L. Wang and C. A. Alexander, "Big data in design and manufacturing engineering," *American Journal of Engineering and Applied Sciences*, vol. 8, no. 2, pp. 223–232, 2015.

[5] R. Hattori, "Big data and analytics for semiconductor manufacturing," *Technical Report, IBM*, pp. 1–25, 2013.

[6] D. Wu, D. W. Rosena, L. Wang, and D. Schaefer, "Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation," *Computer-Aided Des.*, vol. 59, pp. 1–14, 2015.

[7] K. Ro, C. Zou, and Z. Wang, "Outlier detection for high-dimensional data," *Biometrika*, vol. 102, no. 3, pp. 589–599, 2015.

[8] V. Bolon-Canedo, N. Sanchez-Marono, and Amparo-Betanazos, "Feature selection for high-dimensional data," *Prog Artif Intell*, vol. 5, no. 65, pp. 65–75, 2016.

[9] R. E. Bellman, "Adaptive control processes: a guided tour," *Princeton University Press*, vol. 4, p. 5, 1961.

[10] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modelling under imbalanced distributions," *arXiv: 1505.01658v2*, pp. 1–48, 2015.

[11] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[12] L. Xuan, C. Zhigang, and F. Yan, "Exploring of clustering algorithm on imbalanced data," *2013 8th International Conference on Computer Science & Education (ICCSE)*, pp. 89–93, 2013.

[13] M. B. Kursa and W. R. Rudnicki, "Feature selection with the boruta package," *Journal of Statistical Software*, vol. 36, no. 11, pp. 1–13, 2010.

[14] D. S. Kumar and S. Sukanya, "Feature selection using multivariate adaptive regression splines," *International Journal of Research and Reviews in Applied Sciences And Engineering (IJRRASE)*, vol. 8, no. 1, pp. 17–24, 2016.

[15] S. Sperandei, "Understanding logistic regression analysis," *Biochem Med (Zagreb)*, vol. 24, no. 1, pp. 12–18, 2014.

[16] H. Grahn, N. Lavesson, M. H. Lapajne, and D. Slat, "A cuda implementation of random forests - early results," *Third Swedish Workshop on Multi-core Computing*, 2010.

[17] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms using different performance metrics," in *In Proc. 23 rd Intl. Conf. Machine learning (ICML06*, 2005, pp. 161–168.

[18] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, vol. 5, no. 2, pp. 1–11, 2015.

[19] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[20] J. Kumaran, J. Edery, P. Sunkara, and S. Jaswanth, "Analysis of data from a semi-conductor manufacturing process using different classification models and principal component analysis," The Texas A&M University System, Tech. Rep., 2014.

[21] K. Kerdprasop and N. Kerdprasop, "Feature selection and boosting techniques to improve fault detection accuracy in the semiconductor manufacturing process," *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, 2011.

[22] R. R. Bouckaert, E. Frank, M. A. Hall, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "Weka-experiences with a java open-source project," *Journal of Machine Learning*, vol. 11, pp. 2533–2541, 2010.

[23] S. Munirathinam and B. Ramadoss, "Predictive models for equipment fault detection in the semiconductor manufacturing process," *IACSIT International Journal of Engineering and Technology*, vol. 8, no. 4, pp. 273–285, 2016.

[24] J. Kim, Y. Han, and J. Lee, "Data imbalance problem solving for smote based oversampling: Study on fault detection prediction model in semiconductor manufacturing process," *Advanced Science and Technology Letters*, vol. 133, pp. 79–84, 2016.

[25] H. F. Jelinek, A. Yatsko, A. Stranieri, S. Venkatraman, and A. Bagirov, "Diagnostic with incomplete nominal/discrete data," *Artificial Intelligence Research*, vol. 4, no. 1, pp. 22–35, 2015.

[26] A. Kaveri Sharma*, "A study on effects of intrinsic characteristics of datasets on classification performance," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 6, no. 1, pp. 198–204, 2016.

[27] S. Arlot, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, pp. 40–79, 2010.

[28] J. H. Steiger. (2015) Principal components analysis. [Online]. Available: http://www.statpower.net/Content/312/R%20Stuff/PCA.html

[29] E. G. Kulkarni and R. B. Kulkarni, "Weka powerful tool in data mining," *International Journal of Computer Applications*, pp. 10–15, 2016.

[30] Ensembles - rdd-based api. https://spark.apache.org/docs/2.1.0/mllib-ensembles.html.