International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 14, No.1 (Dec-23)

http://dx.doi.org/10.12785/ijcds/1401117

Predictive Approach To The Degree Of Business Process Change

Chahira CHERIF[1], Mustapha Kamel ABDI[1], Adeel AHMAD[2] and Mohammed MAIZA[3]

[1] *LRIIR Laboratory, Faculty of Exact and applied Sciences, Ahmed Benbella Oran 1 University, Algeria*
[2] *Laboratoire d'Informatique Signal et Image de la Côte d'Opale, Université du Littoral Côte d'Opale, Calais, France*
[3] *Faculty of Mathematics and Computer Science, USTO, Algeria*

Received 14 Nov. 2022, Revised 2 Nov. 2023, Accepted 30 Nov. 2023, Published 28 Dec. 2023

Abstract: Computer systems must respond to frequently changing user needs in order to remain operational. Their increasing size and operational complexities intend to make them difficult to maintain. A change in a business process is a complicated task especially during the process execution, where a small change can significantly affect the rest of the system with undesirable impacts. In this work, we focus on studying the problem of change impact propagation in Business Process Management (BPM). We propose in this paper an approach that can predict the level of change (LC) in business models. There are three level of changes (Low, Medium, and High) based on structural metrics as used in the predictive model. Five different machine learning (ML) algorithms are used in this model to show their comparison analysis. This issue is important as the LC in business process before implementing any changes helps the organization to make decision in prior. To validate the purposed approach, the experiments conducted in this study show the improved performance using the SVM and Guassian Naïve Bayes algorithms.

Keywords: Business Process, BPM, BPMN2.0, Business Process Evolution, Change Impact Propagation, Metrics, Machine Learning.

1. Introduction

Computer systems must respond to frequently changing user needs in order to remain operational. Their increasing size and operational complexities intend to make them difficult to maintain. A change in a business process is a complicated task especially during the process execution, where a small change can significantly affect the rest of the system with undesirable impacts. In this work, we focus on studying the problem of change impact propagation in Business Process Management (BPM). We propose in this paper an approach that can predict the level of change (LC) in business models. There are three level of changes (Low, Medium, and High) based on structural metrics as used in the predictive model. Five different machine learning (ML) algorithms are used in this model to show their comparison analysis. This issue is important as the LC in business process before implementing any changes helps the organization to make decision in prior. To validate the purposed approach, the experiments conducted in this study show the improved performance using the SVM and Guassian Naïve Bayes algorithms. Currently, enterprises are changing the way they deal with their business processes to better exploit the BPM technologies and thus facilitating the management of their services. In the past decade, the enterprises have experienced a significant evolution, especially in the area of business process management [1]. These processes are confronted with several changes without knowing their consequences. The validation of a change before its application (implementation) is a crucial step for decision makers. This is why the study of the analysis and prediction of the impact of a change in a business process is important in order to avoid undesirable changes whose consequences can be disastrous or unknown. At this stage, the change is only at the proposal stage and therefore its impact has not yet been effectively realized. As the structural metrics relate to static properties which influence process performance. For this reason, we have chosen these metrics to estimate changes, where each measurement corresponds to a specific type of change in a BPM. In this work, our major objective is to find a solution for an *a priori* prediction of the level of change in business models. There are three levels of changes (Low, Medium, and High) in accordance to the structural metrics used in the predictive model. For this purpose, five different classification algorithms (Gaussian Naïve Bayes, Adaboost, Random Forest, SVM, and KNN) are used in this model to show their comparative analysis. This issue is important as the level of changes in business processes may help to make early decisions regarding the change implementation The rest of the paper is organized as follows: in section 2, we briefly review the relevant work in

E-mail address: cherif.chahira@univ-oran1.dz, abdimk@yahoo.fr, adeel.ahmad@univ-littoral.fr, mohammed.maiza@univ-usto.dz

the available literature. Then, in section 3, we describe our working methodology, which we follow in two main phases of our proposed approach; where the first phase defines the collection of data composed by the different versions of a BPMN 2.0 process (Business Process Modeling Notation), and in the second phase, we deal with prediction of the change impact using supervised ML. The experiments and results we have achieved, as well as the performance evaluation of our model, are discussed in sections 4 and 5 Finally, the section 6 concludes the content of this paper and discusses some short-term perspectives of the current work.

2. Related Work

Maintenance is the most expensive phase of the software life cycle and it is still an active research domain to reduce the cost and risks of modifications on the system. Several works have been carried out in this direction. We refer, for instance, an approach in [2] which proposes a model based on mathematical equations making it possible to estimate the cost of the change of a given object-oriented software with regard to these measurements through a modeling of the relation between the change impact and the metrics known as coupling. In the same way, the authors in [3] propose a model which shows the importance of the metrics through their bonds with the attributes of the object-oriented software quality. Moreover, it is significantly practical and flexible for all types of changes. It allows quality estimation and validation. For the verification of the proposed probabilistic model (automaton), they use model-checking and the prisme tool. The authors in [4] have developed a decision-support tool to reduce the cost of change, using coupling metrics to diagnose the software maintenance process. The motivation behind the work cited above is to improve the maintenance of object-oriented systems, and to intervene more precisely in the task of analyzing the impact of change. Among these studies, we have chosen to focus our research using metrics, but on a study of the evolution of business process management. On the other hand, managing the evolution of business processes requires a thorough understanding of the causes of changes, their application levels and their impacts on the rest of the system. Over the past decade, several efforts have been made to manage complex processes that they must ensure reliability and adapt to changes, where the work of [5] proposes a multi-level model of abstraction, expanding the scope of work towards a new research theme on process modelling. Another recent example of the development of process modelling research is the work of [6] which links BPM to the field of risk management. As stated in [7] a robot mimics a human's manual path through Robotic Process Automation (RPA) that contains “software agents” which perform tasks in a business process. Another study realized in [8] consists in proposing a set of contributions allowing a verification of the coherence and conformity of the business process models after each change, and then to elaborate on *a priori* evaluation of the structural and qualitative impact of the modifications. In the same sense, the objective of [9] is to combine the quantitative and non-quantitative management of the change in the ERP projects (Enterprise Resource Planning) in order to lift their limits. The proposed approach is based on the application of the data-mining process. The authors are interested in three main concepts, which are the magnitude, the management effort and the change impact. The idea is to exploit the information collected from previous software development projects to establish quantitative and objective relationships between these three concepts. In [10], the authors propose a model-driven approach, and a model in SCA (Service Component Architecture) language that improve change management by promoting independence between BPMN and SCA model changes. The authors in [11] propose a graph-based approach and graph rewriting systems to model and simulate the propagation of the change impact in the various components of an application based on the implementation of business process models. The objective of [12] is to analyze the business processes of an enterprise from the dependencies of its activities, data and roles of the actors and to store them in a matrix format. Then the analysis is extended on several versions in order to create a learning base that will be exploited for probabilistic (Bayesian) inferences. Another proposal in [13] is based on BPMN2.0 dependency ontologies in order to analyze the different changes in the history of a business process evolution. The second part of the related works are all focused on improving BPM, and we have taken into account their strengths and perspectives, leading us to our study path. Finally, we highlight the importance of present work in studying the change impact propagation in the business processes structure.

3. Proposed Approach

Our objective is to propose a predictive approach of LC for a given ”current” version *i* of a business process into another ”in-future” version *i+1*. This approach is based on the classification of business processes from various domains, following the extraction of comparison data between successive versions allowing us to compute the considered metrics in order to classify any instance of the dataset according to the degree of change (High, Medium or Low). Then, by exploiting ML algorithms, we can predict the LC of a pair of versions (version *i* and version *i+1*) in input. As a result, the business process designer can be helped to be able to make a decision about whether or not to apply the change in the later version (version *i+1*). The Figure 1 presents the general architecture of our approach: Calculating the derived metrics, in this case the structural metrics of change and variation, the second phase is concerned with classifying the business processes according to the degree of change undergone by the later as they evolve from one version to another. The result of this phase is the classification of business processes according to their degree of change.

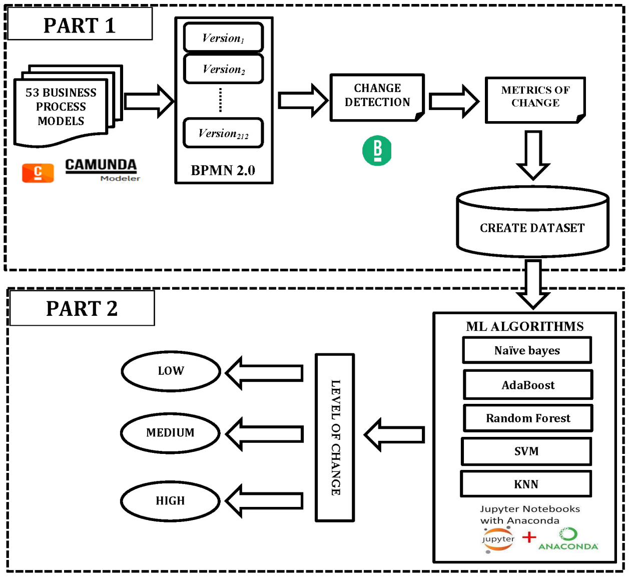


Figure 1. Architecture Of The Proposed Approach

A. *Part 1: Creation Of The Dataset*

Given the unavailability of the dataset in the domain (benchmark), with its history (different versions), we were enforced to create our own dataset. Thence, we collected 53 business processes in the Net, from which we created 212 versions of business processes carrying changes between the different versions. Then, as shown in Figure 1, we take each pair of successive versions (version *i* and version *i+1*) as input for each individual iteration and compare them to detect changes between the two versions. To do this, we used the BPMN Model Diffing[1] tool, an open source tool that allows us to compare two versions of a business process as input and to detect the different changes made between the two versions as output. In order to make a good comparison, we extract and calculate the basic metrics (basic change metrics) as defined in Table I and II. Moreover, in order to quantify the changes, we also compute the structural change metrics, more precisely those related to flow objects, connection objects, artifacts or area of responsibility (swimlane).

1) *Basic Metrics*

The basic metrics are calculated by counting the different elements that compose the business process. It is useful to note that the basic metrics considered in the present study are those that concern only the two categories: flow objects category and connection objects category [14], [15], [16] also as listed in Table I and II.

2) *Derived Metrics*

Derived metrics are some more complex metrics, which result from the aggregation of several basic metrics, and they are divided into two categories: structural metrics of change and structural metrics of variation.

The structural metrics of change defined here are inspired by the basic metrics defined in structural process analysis [17]. The structural metrics of variation answer questions related to the objective of measuring variation in process properties. These metrics are determined on the basis of structural metrics derived from structural process analysis. They refer to more complex properties of the process, which are obtained by differentiating between the values of the corresponding derived metrics of the ”current” and ”in-future” (or ”implemented”) processes [18].

In this study we choose metrics that correspond to the addition and deletion of activities, events to be supervised/-generated and information to be produced/treated, as well as a metric that measures the complexity of the process control flow and another that estimates the depth of the activities. Furthermore, we considered 12 more metrics whom the detailed abbreviations and descriptions are listed in Table III. Broadly, these are defined as follows:

- Structural metric of change in deleted activities to be executed (DA): It returns the rate of deleted activities in the different zones of responsibility, where the responsibility relationship is represented by the fact that the activity is inside the area of responsibility called lanes which are currently used to assign activities to roles, systems or services of the organization in the BPMN2.0 diagrams.

$$DA = \frac{RRCO \times 100}{RRC} \tag{1}$$

- Structural metric of change in added activities to be executed (AA): It returns the rate of added activities in the different areas of responsibility (Lanes).

$$AA = \frac{RRI \times 100}{RRC} \tag{2}$$

- Structural metric of change in deleted actionable information (DAO2A): It returns the rate of deleted association arcs from data objects to activities.

$$DAO2A = \frac{DO2ACO \times 100}{DO2AC} \tag{3}$$

- Structural metric of change in added information to be processed (AO2A): It returns the rate of added association arcs from data objects to activities.

$$AO2A = \frac{DO2AI \times 100}{DO2AC} \tag{4}$$

- Structural metric for change in deleted output information (DAA2O): It returns the rate of deleted association arcs from activities to data objects.

[1] https://demo.bpmn.io/diff

TABLE I. Flow Objects Category

Activities	TNA	Total Number of Activities in the process
Events	TNE	Total Number of Events in the process
Gateway	TNG	Total Number of Gateways in the process

TABLE II. Connection Objects Category

Message flow	NSFM	Number of Message Flows between process participants
Sequence flow	NSFA	Number of Sequence Flows between Activities in the process
	NSFE	Number of Sequence Flows from Events in the process
	NSFG	Number of Sequence Flows from Gateways in the process

TABLE III. Table Of Abbreviations

Equations	Abbreviation	Description
(1) (2)	DA	Deleted Activities
	AA	Added Activities
	RRCO	Responsibility type relationship in ”current only”
	RRI	Responsibility type relationship in ”implemented”
	RRC	Responsibility type relationship in ”current”
(3) (4)	DAO2A	Deleted association arcs from data objects to activities
	AO2A	Added association arcs from data objects to activities
	DO2ACO	Association data objects to activity in ”current only”
	DO2AI	Association data objects to activity in ”implemented”
	DO2AC	Association data objects to activity in ”current”
(5) (6)	DAA2O	Deleted association arcs from activities to data objects
	AAA2O	Added association arcs from activities to data objects
	A2DOCO	Association activities to data objects in ”current only”
	A2DOI	Association activities to data objects in ”implemented”
	A2DOC	Association activities to data objects in ”current”
(7) (8)	DSF2A	Deleted monitorable events (sequence flows from events to activities)
	AE2A	Added supervised events
	SFE2ACO	Sequence flow event to activity in ”current only”
	SFE2AI	Sequence flow event to activity in ”implemented”
	SFE2AC	Sequence flow event to activity in ”current”
(9) (10)	DA2E	Events generated by deleted activities
	AA2E	Events generated by added activities
	SFA2ECO	Sequence flow activity to event in ”current only”
	SFA2EI	Sequence flow activity to event in ”implemented”
	SFA2EC	Sequence flow activity to event in ”current”
(11)	SMCFC	Control flow complexity variation
	$CFC_{to\ be}$	The number of decisions in the ”implemented” process
	$CFC_{as\ is}$	The number of decisions in the ”current” process
(12)	SMMDV	Maximum nesting depth variation
	$PIM_{to\ be}$	The maximum nesting depth in the ”implemented” process
	$PIM_{as\ is}$	The maximum nesting depth in the ”current” process

$$DAA2O = \frac{A2DOCO \times 100}{A2DOC} \quad (5)$$

- Structural metric of change in added production information (AAA2O): It returns the rate of added association arcs from activities to data objects.

$$AAA2O = \frac{A2DOI \times 100}{A2DOC} \quad (6)$$

- Structural metric of change in deleted supervised events (DSF2A): It returns the rate of deleted monitorable events (sequence flows from events to activities).

$$DSF2A = \frac{SFE2ECO \times 100}{SFE2AC} \quad (7)$$

- Structural metric of change in added supervised events (AE2A): It returns the rate of added events to be monitored.

$$AE2A = \frac{SFE2AI \times 100}{SFE2AC} \quad (8)$$

- Structural metric of change in events generated by deleted activities (DA2E): It returns the rate of events generated by deleted activities (sequence flows from activities to events)

$$DA2E = \frac{SFA2ECO \times 100}{SFA2EC} \quad (9)$$

- Structural metric of change in events to be added (AA2E): It returns the rate of events generated by added activities.

$$AA2E = \frac{SFA2EI \times 100}{SFA2EC} \quad (10)$$

- Structural metric of control flow complexity variation (SMCFC): It returns the rate of decisions in the process flow.

$$SMCFC = \frac{(CFC_{to\ be} - CFC_{as\ is}) \times 100}{CFC_{as\ is}} \quad (11)$$

- Structural metric of maximum nesting depth variation (SMMDV): It returns the rate of decisions in the control flow required to execute the activity with the greatest nesting.

$$SMMDV = \frac{(PIM_{to\ be} - PIM_{as\ is}) \times 100}{PIM_{as\ is}} \quad (12)$$

The example in Figure 2 illustrates how to calculate the two metrics DA and AA of the business process ”Online Shopping Process” by applying the formula (1) and (2) by using the BPMN Model Diffing tool :

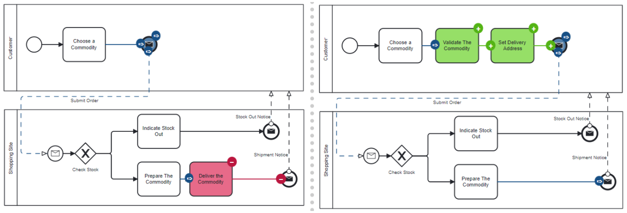


Figure 2. Comparison Of Two Successive Versions Of The Business Process : ”Online Shopping Process”

As illustrated in Figure 2, we observe that there is only one activity deleted among four activities in total and there are two activities added. In this context, the calculation of the two metrics is as follows.

$$DA = \frac{Number\ of\ activities\ deleted}{Total\ number\ of\ activities} \times 100 = \frac{1}{4} \times 100 = 25.00\%$$

$$AA = \frac{Number\ of\ activities\ added}{Total\ number\ of\ activities} \times 100 = \frac{2}{4} \times 100 = 50.00\%$$

B. Part 2: Change Impact Prediction

Following the comparison of two successive versions of a business process, we update the already calculated metrics in a 12-element vector. The dependent variable change level NV_CHANG is defined as the average of the metrics of each individual instance. In order to use the classification algorithms, we need to discretize this variable. To do this, we use the IBM SPSS (Statistical Package for the Social Sciences)[2] software, a tool used for statistical analysis, which we used to discretize this dependent variable (variable to be predicted). Table IV shows the discretization ratio of the dependent variable provided by SPSS, describing the clusters of assignment thus giving 3 values to this variable (LC) which are ”Low”, ”Medium” and ”High”. Table V gives the number of observations for each cluster.

TABLE IV. Cluster Assignment Of Observations

Final cluster centers	1	2	3
NV_CHANG	14,027	49,312	36,661

TABLE V. Number Of Observations In Each Cluster

Cluster	1	30,000
	2	104,000
	3	77,000
Valid		211.000
Missing		0.000

[2] https://www.ibm.com/products/spss-statistics

This part is concerned with the classification of business processes according to the degree of changes undergone by the later version in evolving from one version to another. It thus qualifies as a ML problem and more precisely a classification problem.

Our dataset is composed of the comparisons of different successive versions of the BPMN2.0 processes (i.e., two successive versions: V_i - V_{i+1}) and their structural metrics. The dataset contains the data of each pair of versions in rows while the columns gives the structural metrics that we have already calculated along with a last column indicating the class which is the LC of each process (as shown in Table VI). The result of this phase is the classification of the business processes according to their degree of change.

TABLE VI. Dataset Overview

Versions	DA	AA	...	NV_CHANG
V_0-V_1	25.00	50.00	...	Low
V_1-V_2	75.00	0.00	...	Low
V_2-V_3	50.00	25.00	...	Low
V_3-V_4	0.00	0.00	...	Medium
...	...	...	...	...

For this purpose, we initially considered opting for the Naïve Bayes classifier for several reasons:

- It is a supervised learning classifier and our problem consists in classifying business processes according to their LC (High, Medium and Low) from quantitative variables (metrics considered) characterizing these processes.

- Classification is possible even with a small dataset, which is the case in this study.

- The Naïve Bayes Classifier algorithm assumes the independence of the variables, but this is violated in the majority of real cases [19]. The authors in [19] state that despite the violation of the variable independence constraint, Naïve Bayes gives good classification results. Moreover, the Gaussian approach works with continuous values as it is also the case with all the metrics we considered for the construction of our dataset.

For best result, research work comparing several algorithms is carried as follows [20] :

- Adaptive Boosting (Adaboost): This is a ML approach based on the idea of creating very accurate prediction rules by combining several relatively weak and imprecise rules.
- Random Forest: This is an ensemble approach for classification and regression that works by building a multitude of decision trees at the time of training.
- Support Vector Machine (SVM): SVMs are algorithms that use a nonlinear transformation of the training data. They project the training data into a space of higher dimension than their original space. In this new space, they search for the hyperplane that allows an optimal linear separation of the training data using the support vectors and the margins defined by these vectors.
- K-Nearest Neighbors (KNN): The basic idea of the KNN algorithm is to classify a new unlabeled page P based on the dominant class of K-Nearest Neighbors in the training space.

4. Experiments And Results

In this work, following a search in the Net, we collected 53 types of processes from which we were able to create 212 versions following changes that we made using the BPMN2.0 process editor the open source Camunda Modeler[3]. We have as input the two successive versions and following the comparison between them using the BPMN Model Diffing tool (3-A), we obtain the changes. In this study, we are interested in the addition and deletion of components and data in the business processes, then we calculate the 12 metrics using the number of changes obtained following the comparison. After the extraction of comparison data between successive versions, we calculate the considered metrics to classify each instance of the dataset using the SPSS tool (see 3-B) according to the degree of change (high, medium or low). The Table VI gives an overview of the obtained dataset. For the classification task, we used the Python language (version 3.5.3)[4]. It includes the APIs and packages that support most of the classification algorithms. We exploited the Anaconda Navigator 1.10.03 tool as an environment manager and the Jupyter Notebook (6.1.4) [20] tool as a notebook environment to edit and execute Markdown text and Python code. We prepared our data before providing it to the machine for learning purpose. We have to specify the dataset that is used for training and the dataset for the test. Thus, we imported from the module sklearn.model_selection the function train_test_split which allows splitting the dataset and creating our trainset and testset. For our case, we have opted for 80% for the trainset and 20% for the testset.

After importing the learning model, we proceed to its instantiation and we train this model on the trainset with the fit() function. Then we proceed to the test of the model on the test-set with the predict() function by providing it with the x_test (descriptors of the test-set), and on display we get the results of the prediction.

5. Evaluation Of The Model And Results Discussion

This step consists in evaluating the performance of our model using several evaluation metrics (precision, recall,

[3] https://camunda.com

[4] https://anaconda.org/anaconda/python

and F1-score). These are measures calculated from the elements found in a confusion matrix (Table VII).

TABLE VII. Confusion Matrix

Class	Y	$\overline{Y}$
Y	TP	FP
$\overline{Y}$	FN	TN

This matrix is a table that presents the different predictions and test results, comparing them with real values where :

- TP (True Positive) : Number of well-predicted processes in class Y.
- FP (False Positive) : Number of processes predicted to be in class Y when they should not be.
- FN (False Negative) : Number of processes that are predicted to be of the $\overline{Y}$ class when in fact they are not.
- TN (True Negative) : Number of correctly predicted processes in the $\overline{Y}$ class.
- Precision: minimizes the rate of False Positives in predictions. For example, it avoids incorrectly classifying processes belonging to one of the High or Medium classes as Low.

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

- Recall : minimizes the false-negative rate in predictions :

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

- F1-score: the ratio between precision and recall.
- Accuracy : It is the rate of success or recognition.

$$Accuracy = \frac{TP + TN}{N} \tag{15}$$

Where N is the total number of business processes.

The results are shown in Table VIII that are achieved with the Gaussian Naïve Bayes classifier, according to the evaluation report generated.

A. The Prediction Of A New Example

Let us take an example of a new process not belonging to the dataset. As explained in section 3-A, after performing the comparison of two successive versions of this process, we pass the result vector of the comparison as an input parameter to the Gaussian.predict() prediction function.

B. Comparison Of The Naïve Bayes Model With Other Models

For validation purposes, we make a comparison between the chosen model (Gaussian Naïve Bayes) and other supervised learning models, which are as follows [21]: We start by importing the packages of the different classifiers as discussed above, and then instantiate the corresponding algorithms. For learning and testing, we proceed in the same way as the Naïve Bayes model for these four models. Then, we launch the evaluation, model by model. The comparative results of these five models, implied in this study, are presented in Table VIII.

TABLE VIII. Evaluation Results Of The Learning Algorithms

LM	LC	Precision	Recall	F1-score
Gaussian Naïve Bayes	High Low Medium	0.74 0.98 1.00	1.00 0.95 0.77	0.88 1.00 0.80
Adaboost	High Low Medium	0.15 0.80 0.48	0.17 0.73 0.89	0.25 0.75 0.69
Random Forest	High Low Medium	1.00 0.55 0.84	0.82 0.88 0.55	0.84 0.66 0.71
SVM	High Low Medium	0.95 0.97 1.00	0.98 0.96 1.00	0.94 0.98 1.00
KNN	High Low Medium	0.96 0.69 1.00	1.00 1.00 0.55	0.88 0.68 0.64

According to the evaluation results shown in Table VIII, we obtained a prediction value equal to 95% by the SVM model. Furthermore, as a performance evaluation of SVM, we have an average accuracy equal to 97.33%, which means that the sensitivity of the SVM model is strong and that it is able to select a certain class in our dataset (the recall value is higher, so the TP -True positive- value is also higher than the FN -False Negative- value). The validation of these results is encouraging as these are well predicted in the dataset. On the other hand, we also obtained convergent values of Random Forest and KNN, with successive mean values of 0.71 and 0.72 and good accuracy of high and medium level of change. As shown in Table VIII, the average prediction of Gaussian Naïve Bayes is equal to 0.75, with excellent high and medium level of change accuracy. Also, the adaboost model gave us a prediction of 0.59 and poor accuracy of high class level of change. This indicates that adaboost gave more weight to poorly classified observations. The motivation behind our choice of the Gaussian Naïve Bayes model was based on theoretical analysis. This does not preclude the possibility of other models performing equally well, as in the case of SVM, for example, which demonstrated its performance with a competing recall rate and F1-score.

In order to verify this premier evaluation and obtained result through the comparison of the five models, and given that the size of our dataset is yet small, we repeat our experimentation (further evaluation) by considering the cross-validation (90% of the dataset for learning and 10% for testing). Then, we compared the results of these evaluations by focusing during this iteration on the average of the Average of Accuracy. Table IX shows the result of this comparison.

We used cross-validation in ML to estimate the skill of machine learning models on a new dataset [22]. We can see that the result displayed in the Table IX confirms our earlier result (see Table VIII) in this study. More precisely, the SVM algorithm gives the best classification results followed by the Gaussian Naïve Bayes algorithm. While the AdaBoost model gave low results.

TABLE IX. Results Of The Five Algorithms With Cross Validation

Model	Average Accuracy
Gaussian Naïve Bayes	0,77
AdaBoost	0,60
Random Forest	0,71
SVM	0,97
KNN	0,72

6. Conclusion And Future Work

In this work, we are interested in predicting the rate or LC of business processes. To do so, we proposed a predictive approach to classify a business process according to its High, Medium or Low LC. The unavailability of business process datasets led us to search on web to collect open source BPMN2.0 business process systems from which we created new versions. Then, we calculated structural metrics of change (12 metrics considered) following comparisons of successive versions of the business processes. This attributed us to create our own dataset, consisting of 212 versions. For classification, we used five ML algorithms, which are Gaussian Naïve Bayes, Random Forest, AdaBoost, SVM and KNN. The results of our experiments show the performance of the SVM and Guassian Naïve algorithms, expressed by the values of the evaluation metrics i.e. precision, recall and F1-score. These results have shown the best values of the evaluation metrics (precision, recall and F1-score) using the SVM model, because once we find good decision frontiers (a hyperplane of separation), the data points are closer to each class. Consequently, the frontier can adapt to the new sample, and evidently it corresponds to our experiments. The performance of these results is followed by the Gaussian Naïve Bayes model after the SVM. While, these empirical data showed the lowest values of these metrics by using the AdaBoost, which is sensitive to noisy data and outliers model. These results are subsequently confirmed by re-running our experiments considering cross-validation. The performance evaluations show that our prediction results are encouraging and this allows our prototype to help the organization to make decisions in prior and with success and be exploited to avoid errors and risks on the process functioning.

In perspectives of this work, we mainly aim at automating the process of calculating the structural metrics of change instead of doing the calculation through the excel tabular sheets. We then aim to import the final database to our prototype Moreover, as we have found it difficult to collect and create historical versions of BPMN2.0 processes we are inspired to enrich the dataset with a large number of business processes. In order to further improve the accuracy of the prediction of the LC, we are also aiming to collect the database from another domain. It shall allow an adaptation of metrics estimating the various changes, as well as the integration of other learning algorithms for better predictions. Consequently, this can help making the right decision when faced with a new implemented version.

Acknowledgment

The authors would like to acknowledge the support of the LISIC Laboratory of ULCO, Calais, for this study.

References

[1] A.Ahmad, M.Bouneffa, and H.Basson, “A declarative approach for change impact analysis of business processes,” *Enterprise Interoperability IX*, pp. 169–180, 2023.

[2] M.Dahane, M.K.Abdi, M.Bouneffa, A.Ahmad, and H.Basson, “Using design of experiments to analyze open source software metrics for change impact estimation,” *International Journal of Open Source Software and Processes (IJOSSP)*, vol. 10, pp. 762–781, 2019.

[3] M.Bouslama and M.K.Abdi, “Towards a formal approach for assessing the design quality of object-oriented systems,” *International Journal of Open Source Software and Processes (IJOSSP)*, vol. 12, pp. 1–16, 2021.

[4] M.Z.Dinedane and M.K.Abdi, “Towards group decision support in the software maintenance process,” *International Journal of Decision Support System Technology (IJDSST)*, vol. 14, pp. 1–22, 2022.

[5] G.Polančič, “Bpmn-l:a bpmn extension for modeling of process landscapes,” *Computers in Industry*, vol. 121, p. 103276, 2020.

[6] E.Lamine, R.Thabet, A.Sienou, D. Bork, F.Fontanili, and H.Pingaud, “Bprim: An integrated framework for business process management and risk management,” *Computers in Industry*, vol. 117, p. 103199, 2020.

[7] R.Syed, S.Suriadi, M.Adams, W.Bandara, S. Leemans, C.Ouyang, and H.AReijers, “Robotic process automation: contemporary themes and challenges,” *Computers in Industry*, vol. 115, p. 10362, 2020.

[8] O.M.Kherbouche, A.Ahmad, M.Bouneffa, and H.Basson, “Robotic process automation: contemporary themes and challenges,” *11th International Conference on Frontiers of Information Technology*, vol. 115, pp. 235–240, 2013.

[9] A.El-Mhamedi, L.Kermad, and M.Camara, "Proactive management of organizational change using bayesian networks," *9th International Conference on the Modern Information Technology in the Innovation Process of the Industrial Enterprise (MITIP 2007)*, vol. 115, p. 1, 2007.

[10] K.Dahman, F.Charoy, and C.Godart, "Alignment and change propagation between business processes and service-oriented architectures," *IEEE International Conference on Services Computing*, pp. 168–175, 2013.

[11] M.Bouneffa, A.Ahmad, and H.Basson, "Gestion intégrée du changement des modèles de processus métier," *Congrès INFORSID (INFormatique des ORganisations et Systèmes d'Information et de Décision), Grenoble, France*, 2016.

[12] C.Cherif, "Towards a probabilistic approach for better change management in bpm systems," *23rd International Enterprise Distributed Object Computing Workshop (EDOCW)*, vol. 115, pp. 184–189, 2019.

[13] C.Cherif, M.K.Abdi, A.Ahmad, M. Bouneffa, H.Basson, and M.Maiza, "Predictive study of changes in business process models," *Interoperability for Enterprise Systems and Applications Conference (I-ESA'22)*, 2022.

[14] I.Vanderfeesten, J. Cardoso, J.Mendling, H.A.Reijers, and W.VanderAalst, "Quality metrics for business process models," *23rd International Enterprise Distributed Object Computing Workshop (EDOCW)*, vol. 144, pp. 179–190, 2007.

[15] M.Alanen, J.Lilius, I.Porres, and D.Truscan, "On modeling & techniques for supporting model-driven development of protocol processing applications," *Model-Driven Software Development, Berlin, Heidelberg*, pp. 305–328, 2005.

[16] E.R.Aguilar, F.Ruiz, F.García, and M.Piattini, "Evaluation measures for business process models," *ACM symposium on Applied computing*, pp. 1567–1568, 2006.

[17] L. Jiang, Z.Cai, H.Zhang, and D.Wang, "Naive bayes text classifiers: a locally weighted learning approach," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 25, pp. 273–286, 2013.

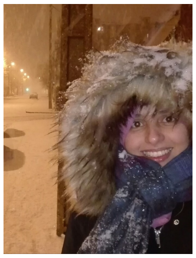
[18] X.Heguy, G.Zacharewicz, and Y.Ducq, "Interoperability markers for bpmn 2.0-making interoperability issues explicit," *Advances in Engineering Research*, vol. 86, pp. 330–333, 2017.

[19] T.Kluyver, B.Ragan-Kelley, P.Fernando, B.Granger, M.Bussonnier, J.Frederic, and C. Willing, "Jupyter notebooks – a publishing format for reproducible computational workflows," *F. Loizides & B. Schmidt (Eds.), Positioning and Power in Academic Publishing: Players, Agents and Agendas*, p. 87–90, 2016.

[20] C.Zhang and Y.Ma, "Ensemble machine learning: methods and applications," *Springer Science & Business Media*, 2012.

[21] V.P.Keon, H.O.Kyoung, J.J.Yong, R.Jihye, S.H.Mun, and W. C.June, "Machine learning models for predicting hearing prognosis in unilateral idiopathic sudden sensorineural hearing loss," *Clinical and Experimental Otorhinolaryngology*, pp. 148–156, 2020.

[22] M.Sahagun and M.Anne, "Machine learning based selection of incoming engineering freshmen in higher education institution," *International Journal of Computing and Digital System*, pp. 325–334, 2021.



Chahira CHERIF is Phd Student at the computer science department, University of Oran 1 Ahmed Ben Bella, Algeria. She received her engineering degree and Master in computer science at the same university. Her main research interests are Artificial intelligence, Business process management and technologies, Software engineering, Software decision support systems and Business rules modeling.

Mustapha Kamel ABDI is a professor at University of Oran 1 Ahmed Ben Bella, Algeria. He holds a master degree and a Ph.D. degree in computer science from Department of Computer Science at the same university. His research interests include the application of artificial intelligence techniques to software engineering, Software quality, Formal specification, Systems analysis and simulations, Data-Mining and Information Research.

Adeel AHMAD is a Research Scientist at the Computing, Signal and Image Laboratory of the Opal Coast (LISIC). His research occurs to add explanations in artificial intelligence systems. His research work are published in the context fourth industrial revolution,(Industry 4.0) to automate the analysis of change impact propagation for qualitative evaluation and improvement of the process models. His research interests include the rules-based business intelligence and ontology, in particular to seek explainability in machine learning. He received his PhD in Computer Science from the University of the Littoral Opal Coast (ULCO), Calais, France.

Mohammed MAIZA received the M.Sc. degree in Computer Science and Engineering from the Department of Computer Science University of Sciences and Technology of Oran (USTO), Algeria. His research interests include Bioinformatics and computational biology, Optimization, Digital image processing, Machine learning, Techniques for microarray data analysis.