Feature selection-based approach for generalized contradiction recognition

Naser GHANNAD1[0000-0001-8491-9150], Roland DE GUIO1[0000-0002-8201-9551] and Pierre PARREND2[0000-0002-1680-1182]

[1] Icube INSA de Strasbourg, Strasbourg, France
[2] ECAM Strasbourg-Europe, Schiltigheim, France
Naser.Ghannad@insa-strasbourg.fr

Abstract.

The objective of this paper is to improve a machine learning based methodology for recognizing the features of a Generalized Physical Contradictions (GPC) before knowing the contradiction itself when the system to be improved can be described by a simulated model based on design parameters and performance parameters. The paper starts with the background about identifying contradictions from data. It focuses on physical contradiction parameters identification with quantitative data and machine learning techniques. Although previous approaches are promising, they still have several drawbacks that require to be fixed. For instance, they do not propose any metric to inform the user about the quality of the result, which depends, among others, on the sample size. These drawbacks mainly appear in case of imbalanced data or complex relation between variables. To address these issues, we first tested different feature importance variable provided by decision tree methods (with the XGBOOST library) and retain the total gain. Second, we compared the XGBOOST methods with the previous proposed SVM based approach to see which one better describes the feature importance of variables involved in a GPC. As result XGBOOST was more robust to the noise from non-important variables. Third, we defined a set of measures for helping the user to know which is the sample size required to get good results with the tested methods.

Keywords: inventive problem solving , feature selection algorithm, generalized physical contradiction, XGBOOST

1. Introduction

The notion of contradiction is a cornerstone of the TRIZ [1]. Multiple tools and methods deal with how to solve invention problems formulated with the help of contradictions [2]–[13]. Fewer works attempt to identify the contradictions underlying an invention problem. Identifying contradictions is an important, but sometimes tricky step in the problem formulation-solving process. The difficulty may be such that one sometimes observes in problem-solving seminars the inability of some people to formulate a contradiction. However, more often an inverse situation is observed, i.e. The expression of numerous contradictions among which one has to choose the contradiction(s) to be dealt with. To respond to this last practical situation, some authors and experts propose concepts of classification of contradictions (core contradiction, system contradiction, etc.) or classification criteria relating to a set of expressed contradictions. In the vast majority of cases, the search for contradictions is a qualitative process carried out by experts. The difficulties mentioned above and the increasing complexity of the representation and behaviour of certain systems have led us to seek digital alternatives.

The research presented in this paper is part of a research that aims at improving the theoretical and practical continuum between design approaches based on optimization and those based on invention [14]. It enriches the work concerning the automated or semi-automated expression of a constraint system based on the results of physical experiments or numerical simulations whose interest has been validated by previous work [15]–[17].In this paper, we focus on the development of numerical tools and algorithms facilitating the identification of the design parameters involved the concepts and contexts of generalized physical contradictions associated with a system of technical contradictions with the aim of constituting a generalized system of contradictions.

In the following at first, we describe how finding the important action parameters that are involve in the contradiction helps the expert to understand the problem inside of the system and give him a clue to improve the process of the system with add or remove the action parameters that are involve in contradiction. Second, we describe the algorithms that can helps us to find the most important action parameters. Third, we test different feature importance of different algorithm to see which one can give better and more robust results to find the important variable. Fourth, by doing some experiments, we showed that, how we can get the right number of sample size based on the accuracy of the model.

2. Background

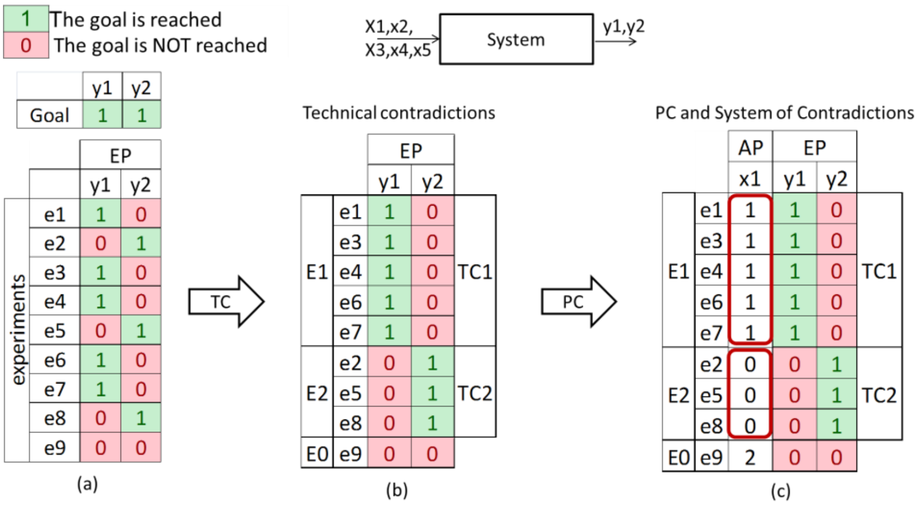
2.1. TRIZ system of contradictions

Commonly, the TRIZ identifies 3 types of contradictions: the administrative contradiction, the technical contradiction and the physical contradiction. The OTSM-TRIZ has made explicit a form of link between technical contradictions and a physical contradiction which is called a system of contradictions. In the research related to this paper, we are interested in the identification of systems of contradictions by using numerical tools and experimental or simulation data. To deal with this issue, it was necessary to represent the concepts of technical, physical and system of contradictions so that these identifications could be made. The following paragraphs recall the definitions and representations of the TRIZ contradictions in a data table.

Consider a system whose performance is evaluated using two evaluation parameters (EP) y1 and y2. To do so, 8 experiments or simulations were performed by acting on the design parameters x1 to x5 of the considered system, which we will henceforth call

Action Parameters (AP) in the remainder of the paper. We give the value 1 to the evaluation parameter, for a given experiment if the simulated system meets the objectives for the evaluation parameter. Otherwise, the evaluation parameter takes the value 0. The result is given in **Fig 1** (a) below. When analyzing it, it can be seen that sometimes one objective can be satisfied but never both objectives at the same time. According to the TRIZ, situations that satisfy y1 but not y2 characterize under these conditions a so-called technical contradiction noted TC1 in **Fig 1** (b). In the same way, situations that satisfy y2 but not y1 have under these conditions a second technical contradiction noted TC2 in **Fig 1** (b). The situation e9, on the other hand, does not bring a contradiction because none of the objectives is achieved; there is no conflict of objectives in this "solution".

To obtain a physical contradiction in the sense of the TRIZ, it is necessary to be able to assign to the contradictions TC1 and TC2 an action variable (AP) which allows one to move from one technical contradiction to the other by modifying its value as shown in the example in **Fig 1** (c). In this example, x1=1 characterizes contradiction TC1 and x1=0 characterizes contradiction TC2. In the TRIZ the physical contradiction of the example is stated: the variable x1 must take both the values 1 and 0 to reach the objectives. The system of contradictions described by OTSM-TRIZ is the coherent set of the two technical contradictions TC1 and TC2 and the physical contradiction that "explains" or "causes" TC1 and TC2.



🟩 1	The goal is reached
🟥 0	The goal is NOT reached

	y1	y2
Goal	1	1

	EP	
	y1	y2
e1	1	0
e2	0	1
e3	1	0
e4	1	0
e5	0	1
e6	1	0
e7	1	0
e8	0	1
e9	0	0

Technical contradictions

		EP		
		y1	y2	
E1	e1	1	0	TC1
	e3	1	0	
	e4	1	0	
	e6	1	0	
	e7	1	0	
E2	e2	0	1	TC2
	e5	0	1	
	e8	0	1	
E0	e9	0	0	

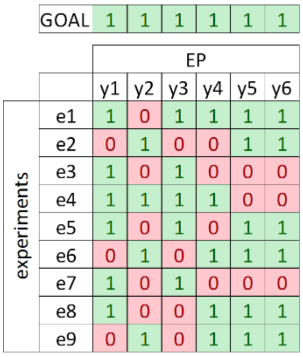
PC and System of Contradictions

		AP	EP		
		x1	y1	y2	
E1	e1	1	1	0	TC1
	e3	1	1	0	
	e4	1	1	0	
	e6	1	1	0	
	e7	1	1	0	
E2	e2	0	0	1	TC2
	e5	0	0	1	
	e8	0	0	1	
E0	e9	2	0	0	

***Fig 1.** TRIZ system of contradiction out of experimental data*

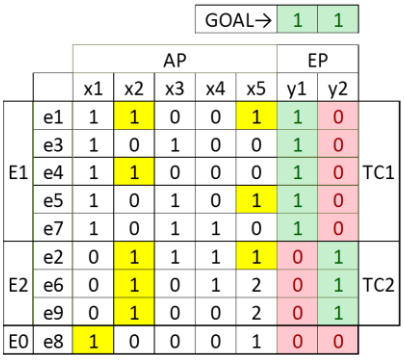
2.2. Limitation on Technical contradiction of the TRIZ models

The example in **Fig 2**, which deals with a problem where six objectives are to be satisfied, illustrates a limitation of the classical TRIZ model of technical contradiction. Indeed, it can be seen from this example that two of the six objectives can always be met at the same time, but that the six objectives can never be met at the same time. In other words, in the sense of the classical TRIZ, there is no contradiction but the problem of conflict between objectives exists. It was therefore necessary to find another way of expressing the technical contradiction. Several proposals called generalized technical contradictions (GTC) have been made to this effect in the literature. They consider at least two or even all objectives simultaneously [18]. We do not elaborate further on this point, which is not the subject of this paper. We will retain for our purpose that as for the system of contradictions of the classical TRIZ, a system of generalized contradictions is composed of two generalized technical contradictions and one generalized physical contradiction. The two generalized technical contradictions GTC1 and GTC2 of a system of contradictions define a partition of the experiments into three sets E1, E2, E0 where E1 represents the experiments for which we have the contradiction GTC1, E2 represents the set of experiments for which we have the contradiction GTC2, and where E0 is constituted by the complementary experiments of the partition. The reader interested in the definitions and the methods of identification of the GTC from the data can refer to the references [19], [20].



	GOAL	1	1	1	1	1	1
		EP					
		y1	y2	y3	y4	y5	y6
experiments	e1	1	0	1	1	1	1
	e2	0	1	0	0	1	1
	e3	1	0	1	0	0	0
	e4	1	1	1	1	0	0
	e5	1	0	1	0	1	1
	e6	0	1	0	1	1	1
	e7	1	0	1	0	0	0
	e8	1	0	0	1	1	1
	e9	0	1	0	1	1	1

Fig 2. TC model limitation example



						GOAL→	1	1	
		AP					EP		
		x1	x2	x3	x4	x5	y1	y2	
E1	e1	1	1	0	0	1	1	0	TC1
	e3	1	0	1	0	0	1	0	
	e4	1	1	0	0	0	1	0	
	e5	1	0	1	0	1	1	0	
	e7	1	0	1	1	0	1	0	
E2	e2	0	1	1	1	1	0	1	TC2
	e6	0	1	0	1	2	0	1	
	e9	0	1	0	0	2	0	1	
E0	e8	1	0	0	0	1	0	0	

Fig 3. PC model limitation example

2.3. Physical contradiction model limitation

To explain the limitations of the TRIZ physical contradiction model consider the example in **Fig 3**, when processing data and look for physical contradictions associated with the TC1 and TC2 technical contradictions. We can see that none of the action parameters alone can explain the TC1 and TC2 contradictions with a conflict on its values. Indeed, for x1, when x1=0 we have the situations of the set E2 which characterize TC2 but when x1=1, we have the situations of E1 which are characteristic of TC1 and situation of E0 namely e8 where none of the objectives is reached. To get around this difficulty two alternative but coherent ways to express the contradiction with the action parameters have been proposed. These contradictions are called by their authors generalized physical contradictions and contextual physical contradictions [21].

2.4. The Generalized physical contradiction

A first approach consists in determining a logical expression of the action parameters C1 which discriminates the situations of E1 from those of E2 and E0 on the one hand, and in determining a logical expression C2 which discriminates the situations E2 from those of E1 and E0 on the other hand. The two expressions thus obtained are then the equivalent of the conflict between parameters of the physical contradiction of the TRIZ. In the case of the example of **Fig 3**, the expressions C1 and C2 below allow to express a generalized physical contradiction (GPC):

C1: (x1=1).(x2=1).(x3=0).(x4=0)

C2: (x1=0).(x2=1).(x3=0).(x4=0)

The generalized physical contradiction can then be expressed with the concepts C1 and C2: "to achieve objectives y1 and y2, both C1 and C2 must be satisfied, which is not possible". The association of a generalized physical contradiction with two generalized technical contradictions is called a system of generalized contradictions (SGC).

2.5. Contextual physical contradiction

The second approach to express the generalized physical contradiction is to add the notion of context [21]. To illustrate this notion, let's take the previous example and observe the expressions C1 and C2. We see that C1 and C2 are distinguished by the value of x1 and share the same values for x2, x3 and x4. Let us then note C the logical expression such as is true if (x2=1).(x3=0).(x4=0). We can then reformulate the previous contradiction by introducing the expression C. In situations where the expression C is true, the action parameter x1 must take both the value 1 and the value 0 to meet the objectives. Expression C is called the context of the generalized physical contradiction and the system of contradictions.

In the case of our example, we find in the context C a physical contradiction of the classical TRIZ. The generalized contradiction within a given context specifies the domain of validity of the contradiction of the classical TRIZ. This information, which is not easy to find in general, is not evoked in the classical TRIZ. However, it can be important for understanding the problem and interesting for solving a problem.

This paper concerns the improvement of numerical techniques allowing the fast and reliable identification of the concepts C1 and C2 and the context C of generalized physical contradictions.

2.6. Seeking for the causes and context of the TC

The work of identifying generalized physical contradictions through numerical data analysis techniques is based on the observation that TC1 and TC2 can be considered as discriminant functions between the E1 and E2 sets of experiments. In [22] an exhaustive search algorithm for all generalized physical contradictions associated with two generalized technical contradictions is proposed by modeling this problem as an integer optimization problem. This problem is recognized as NP-hard, which limits the application of this algorithm to problems the number of action variables of which is lower than 13 when those action parameters have only 2 levels. To bypass this problem, the same author proposes to first identify the variables involved in the searched contradictions using SVM learning techniques[23] and the associated discriminant analysis [19]. The same technique was used to identify the parameter-value pairs of the action parameters involved in the physical contradictions [19]. Taking advantage of this last possibility, Bach proposes an approach for interpreting the weights of the SVM discriminant analysis with respect to selected technical contradictions in the Pareto set of the binary matrix of EPs. This last approach avoids the use of the exhaustive algorithm.

The methods mentioned above have been used successfully in real cases. The exhaustive search method may give many contradictions, but only a few of them are relevant. The practical question of the choice of the contradiction then arises. The direct identification method based on learning does not have this drawback [24]. It provides the parameters and values involved in the pair of technical contradictions but it lets the user conclude on the expression of C1 and C2 and does not provide the context C. Moreover, their application requires knowledge and experience in data analysis to correctly interpret the results of the SVM learning algorithm. In some cases, the interpretation of SVM results remains difficult even for an expert in data analysis.

3. Problematic and expected contribution(s) of the paper

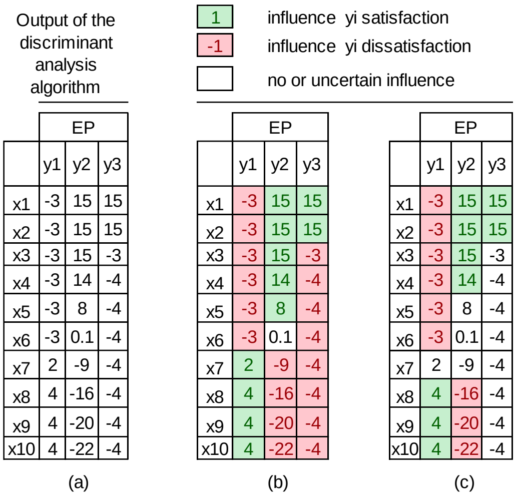
The objective of our current research is to construct a heuristic that quickly provides a user who is not an expert in data analysis with only the relevant generalized physical contradictions and their context when the number of action parameters and values is large. As discussed in Section 2, one disadvantage of SVM based method is the ambiguity that may arise in the interpretation of the weights of the discrimination function provided by the SVM method.

3.1. *Practical limitations of SVM use*

To explain these drawbacks, we recall with the example of **Fig 4** the principle of use and interpretation of SVM outputs. In this example, we consider three evaluation parameters y1, y2, and y3 and an action parameter that can take 10 values x1 to x10. We want to know which values xi explain the achievement or non-achievement of the objective yj. To do this, the SVM discrimination function provides weights (see **Fig 4** (a)) which can be interpreted as follows: if the value of the weight is "enough" positive, then xi explains the achievement of the objective for yj and if it is "enough" negative, then xi explains the unsatisfactory achievement of the objective for yj.

The first practical problem concerns the interpretation of "enough". Indeed, the values defining the limit between a positive or negative action and no discriminatory action are not fixed by the method. Moreover, they may vary according to the evaluation parameter yi. To illustrate this point, it is provided in **Fig 4** (b) and (c), two different interpretations of the weights in **Fig 4** (a). Knowledge about the system can help in the interpretation, but we would like to be able to decide on the basis of the data.

The second practical question concerns the sample size of data needed to ensure that the order of values is stable because with different sample size we see different order of variables and we don't understand which one is more accurate and trustable, we would like to know the necessary sample size to be able to interpret the data.



Output of the discriminant analysis algorithm

1	influence yi satisfaction
-1	influence yi dissatisfaction
	no or uncertain influence

	EP		
	y1	y2	y3
x1	-3	15	15
x2	-3	15	15
x3	-3	15	-3
x4	-3	14	-4
x5	-3	8	-4
x6	-3	0.1	-4
x7	2	-9	-4
x8	4	-16	-4
x9	4	-20	-4
x10	4	-22	-4

(a)

	EP		
	y1	y2	y3
x1	-3	15	15
x2	-3	15	15
x3	-3	15	-3
x4	-3	14	-4
x5	-3	8	-4
x6	-3	0.1	-4
x7	2	-9	-4
x8	4	-16	-4
x9	4	-20	-4
x10	4	-22	-4

(b)

	EP		
	y1	y2	y3
x1	-3	15	15
x2	-3	15	15
x3	-3	15	-3
x4	-3	14	-4
x5	-3	8	-4
x6	-3	0.1	-4
x7	2	-9	-4
x8	4	-16	-4
x9	4	-20	-4
x10	4	-22	-4

(c)

***Fig 4.** Limitation of SVM weights interpretation*

3.2. Problematic of the paper

The main question in this paper is to define which action parameters are involved in generalized physical contradiction (i.e we do not seek for the contradiction itself). In the SVM-based approaches presented in previous works, the idea is to exploit a learning method and its recognition function. We can imagine using other learning algorithms to recognize the variables involved in the contradictions. Each method can produce different results. The question then arises to compare their performance and to know which one is more accurate or robust.

As soon as we know how to measure the quality of a solution provided by a method we can compare the methods. In this paper, we present an alternative method to the SVM approach named XGBOOST [25](eXtreme Gradient Boosting) , which will be compared to SVM [26].

The main problem that we deal with is imbalanced data that exists in some problems, it's means that number of data for each class or group of data is not balanced and that the number of data that exist in one group is very few. Moreover, when the method can deal with imbalanced data, the usual metric like accuracy can't show overfitting or under-fitting for this kind of dataset. That is the reason why we need measures that allow us to deal with these questions. Some of them are proposed in the literature of machine learning like AUC, and PRC curve. We will test them also in the method we propose bellow because this situation often happens in innovation. Our objective is finding accurate and robust models that can show the importance of features involved in the contradiction. Also we looking for an algorithm that can (1) easily deal with multi-class problems and have a way to delete ineffective variables and (2), in a next research step, might provide the concepts C1 and C2 of the GPC).The latest point is among main reasons why we explored the use of XGBOOST.

.

4. Material and method

4.1. Building the artificial data set

The problem of identifying functions C1, C2, and C or the action parameters involved in one of these functions can be put in a generic form illustrated in **Table 1**. In this example, we have three action parameters A1, A2 and A3 which can take two values 1 or 0. The class column of the table corresponds in practice either to a column of the binary matrix or to the selection of the sets E1 (values 1) versus E2 +E0 (zeros values) or E2 versus E1+E0. This column can also be interpreted as the value of the logical function C1, C2, or C to be found. Thus, for example on **Table 1**. if the column "class" corresponds to an EP of the binary matrix we will have the objective reached (EP=1) if (A1=1) OR (A2=1) AND (A3=1). What can still be written EP = A1+A2.A3.

This remark allows us to build a set of test functions that enable us to check to what extent an algorithm provides the right parameters of the contradiction and later if an algorithm finds functions C, C1, or C2.

At the preprocessing stage of our research methodology, we build different datasets with different numbers of action parameters (10, 15, 20) and also with different sizes of data samples (100, 300, 500, 1000).

***Table 1.** Data sample*

A1	A2	A3	Class
1	0	0	**1**
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**1**
0	0	0	**0**
0	1	0	**0**

The functions used for the test part are given below, '.' means AND between two variables, and '+' means OR between two variables.

$$Function\ 1 = A1.A2 \tag{1}$$

$$\text{Function } 2 = A1.A2 + A1.A3 \tag{2}$$

$$\text{Function } 3 = A1.A2 + A3.A7 + A5 \tag{3}$$

$$\text{Function } 4 = A1.A2.A3\ldots A10 \tag{4}$$

$$Function\ 5 = A1{+}A2.A3{+}A2.A4{+}A3.A5.A6{+}A4.A5.A7.A8 \tag{5}$$

They were chosen for their degree of difficulty in treatment. In the following, you can see imbalance ratio (IR) and table of imbalance ratio of these functions.

$$IR = \frac{number\ of\ data\ in\ Majority\ Class}{number\ of\ data\ in\ Minority\ Class} \qquad (6)$$

In this formula, for example if we have 2 outputs for the defined function and if the number of zero outputs in our dataset is equal to 1000 and the number of one output in our dataset is equal to 8, then class 1 or data with output 1 is minority class because it has lower number of data inside the dataset and class 0 is majority then IR will be 1000 divide by 8 is 125. It shows that this dataset is highly unbalanced, and we need a plan to pass this problem.

Table 2. *imbalance ratio of different function with different sampling*

#Sample	F1	F2	F3	F4	F5
100	4.26	1.77	0.51	99.0	0.49
200	4.40	1.81	0.43	99.0	0.42
300	3.54	1.67	0.48	149.0	0.38
400	3.25	1.54	0.45	199.0	0.37
500	3.62	1.68	0.48	124.0	0.39
1000	3.08	1.61	0.42	249.0	0.37

It can be seen in the **Table 2** that Function 4 has the highest imbalance ratio. Among the other functions, F1 and F2 are most challenging ones because the number of data in majority class is much more than number of data in minority class and this makes learning harder because in learning stage, model focus on majority class and neglect the minority class.

Once the test functions are defined, we preprocess the data, it means we binarize the input and output and split the data to training, validation, and test data sets by using 5-fold stratified data sampling. Indeed, as we are dealing with some imbalanced data sets these actions allow having enough data of each value (0 and 1). A stratified sampling ensures that subgroups (strata) of a given class are each adequately represented within the whole sample of classes. In this way, we can see different aspects of the data, and later we can evaluate and check the confidence and robustness of our model in different conditions.

4.2. Global process to find the important variables involved in contradiction

In this paper, we want to evaluate the performance of the "extreme gradient boosting” algorithm to extract the important features involved in physical contradiction and compare it performances to SVM approach. The methodology to identify the features is provided below. The two first stages 1) and 2)consist in getting data and preprocessing them as described in the previous section.

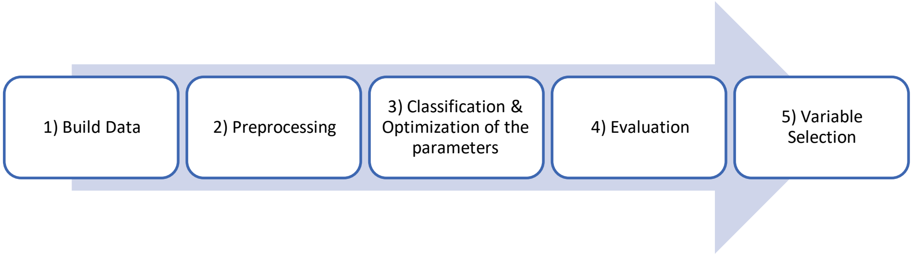


Fig 5. *Our proposed methodology to detect the important inputs involved in contradiction*

In the next step 3) we classify the data, with a gradient boost tree algorithm (XGBOOST). During the training of step 3), we check the model via a Log loss metric in order to prevent as much as possible overfitting and underfitting phenomenon.

After training the model, we evaluate it in step 4), to see how much it might be successful in classifying the data by using the validation set. For evaluating the result, as we deal with imbalanced data, we used metrics that are good when dealing with imbalanced datasets like AUC curve, PRC curve. We consider that in very high-level imbalance dataset PRC curve is a better metric than AUC curve [27], [28]. If model underfits or overfits then our metric on test data will give very different result than on training data and the model and output of model on test data will be useless. The measures used so far in step 4) allow the user of the classification method to "decide" whether the classification model obtained in step 3) has identified the Action Parameters without knowing them beforehand as in a real situation.

In step (5), the user of this algorithm should then decide from the measures which variables will retains as important features.

But, as in our evaluation examples we already know the important features, it is also possible to check afterward the algorithm did find the right features or not. Thus, we added a new evaluation metric, that we call correctness: It is a measure that helps to evaluate the feature selection process and compare them.

$$Correctness = \frac{f(choice\ first\ N(sort\ M))}{number\ of\ actual\ important\ features} \qquad (7)$$

In the formula of correctness, M is equal to the value of the importance of each variable extracted from the model, it could be weight in SVM or gain of each feature extracted from the decision tree, and sort M means, we sort these M from highest value to lowest value, because feature with highest value means that it has more contribution than others to classify the output of dataset. N is equal to the number of important features in the function, for example, if we are looking to evaluate function 1, then only 2 variables are involved in the function, and N value is 2 for this function. Function f in the formula is comparing these selected variables with actual variables that existed in actual function and return number of matching between these two lists, for example, if we have 10 APs and actual function that builds the output is like A1+A2A4 with 3 important variables. Let us imagine that after training and extraction of the feature importance of model, the provided ranking of feature importance is A1, A3, and A4. Then correctness for this situation is equal to two divides by three, because we can predict 2 of 3 important variables of the actual function and correctness will be 66%, this formula only useable now because we already know the actual function and we use it to compare and understand the accuracy of the algorithm.

In the next section, we provide the results obtained with this method. The details of the measure and their interpretation for the user are provided.

5. Illustration and evaluation of the approach

For understanding the problem and to understand which variables play a more important role in contradiction we need to find them inside of the inputs (Action parameters), for that reason after the classifying stage we can check the variable importance from the simulated model. In SVM we can see it via the weights of the kernel and in XGBOOST we can see it via different parameters of the XGBOOST method like gain, cover, weight[29], total gain, total cover, and total gain. weight is the number of times a feature is used to split the data across all trees, gain is the average gain across all splits the feature is used in, cover is the average coverage across all splits the feature is used in, total gain is the total gain across all splits the feature is used in, and total cover is the total coverage across all splits the feature is used in. In this paper, we first explored which one of the XGBOOST measures better explains the important features and later compare XGBOOST measures with SVM measures to see which one is more robust in imbalance situations and which one can give the more accurately the important feature with the lowest sample size.

To compare the different feature importance provided by XGBOOST we used AUC, PRC curve to check how much results are usable and model can simulate the actual function.

In the following we compare the result for different functions, all the functions have 20 inputs with the name of A1 to A20 and one output. But all the input variables are not necessary to describe the function; each of them can be described with one of the 5 functions provided above (equations (1) to (5))

5.1. Comparing different feature importance of XGBOOST

Because we build our dataset, we know which variables are important and which are useless we can now check if our algorithm can find these important variables or not. But we have a problem, XGBOOST give multiple feature importance and we need to figure out which one of them is more suitable to detect the important features. To find out which one of the feature importance of the XGBOOST (weight, Cover, Gain, Total gain, Total cover) can better describe the feature important of actual functions, and because we already know the variables ranking based on the functions we propose a new way to find the best feature important of XGBOOST. For example $Function\ 5 = A1 + A2.A3 + A2.A4 + A3.A5.A6 + A4.A5.A7.A8$ we have 20 variables in our dataset, but only 8 of them are useful in this function. Then we propose a general ranking model for this type of functions. From the function formula most important variable we can see is A1, because output of the function is depending on A1 value and with only A1 we can change the output and it's independent from other variables, second one is A2 because it exist in 2 part of function (A2.A3 and A2.A4) and these two parts are only

depending on one variable that why we can understand second most important variable is A2, third is A3 because it's exist in A2.A3 and A3.A5.A6, fourth is A4 because we find A2 and A3 then A4 will be remain on A2.A4, fifth is A5 because it's exist in two part of function A3.A5.A6+A4.A5.A7.A8, sixth is A6 because it's remain on A3.A5.A6, and seventh and eighth is A7,A8 but they seem to be same important in the function because they do not exist in another part of the function.
From our experimental data set we know the actual function, the useless variables and we can compute a proposed ranking of the features thanks to previous described ranking model. Now we need an algorithm that show us the same situation as described above. It means that we need an algorithm that has the ability to show the A9 to A20 as useless variables and shows A1 as the most important variables and after that A2 to A8 etc. For that reason, we plot different bar chart of 5 feature importance measures of XGBOOST (weight, Cover, Gain, Total gain, Total cover) with different sample size to see which one can show us the uptrend in value when we increase the sample size from 100 to 2000 and also show the downtrend from A1 to A8, because A1 is most important variable and A8 is lowest important variable in the function and most values of A1 to A8 are bigger than A9 to A20.

As can be seen in **Fig 6**, total gain can give A1 to A8 values greater than A9 to A20 values, and when the sample size increases, the value of total gain also increases. It means that when we give more data to the model, the model confidence in important features also increases. And we can also see the downtrend from A1 to A8 which shows that the model is ranking A1 as most important feature and A8 as lowest important feature inside of our function.

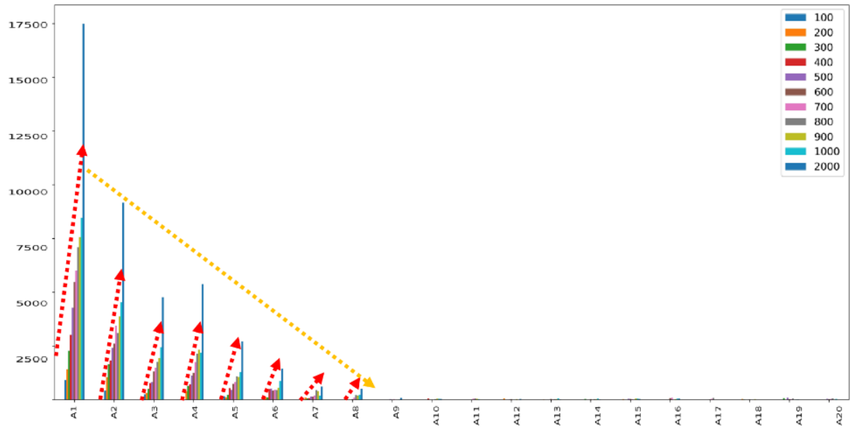


***Fig 6**. Total gain of each variable of XGBOOST with different sample size of function 5*

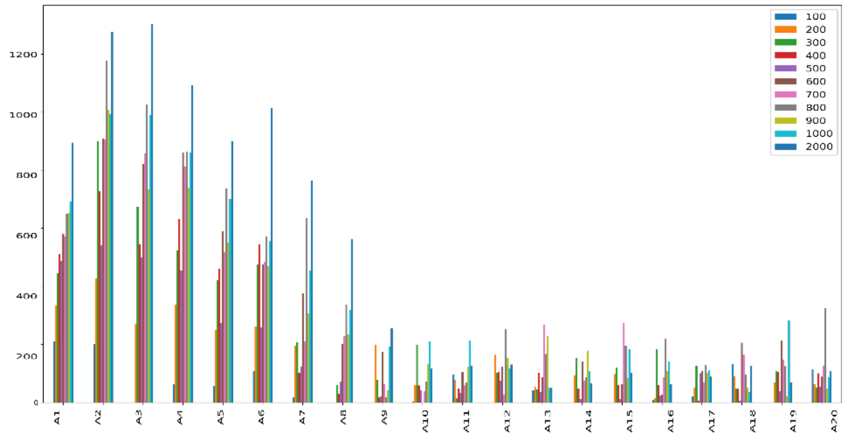


Fig 7. *Weight of each variable from XGBOOST with different sample size of function 5*

One can see an example of inconsistency of our method when we used weight as importance function in Fig 7, because there is no sign of uptrend when we increase the sample size and also there is no sign of downtrend from A1 to A8, and value of A1 to A8 are not higher than the values of A9 to A20 in most of case.

Finally, among the five tested importance features the total gain and total cover could be used for our problem when using bar charts. In order to choose between these two measures, we made more detailed comparisons of them by using line charts to plot them (see **Fig 8** and ***Fig 9***).

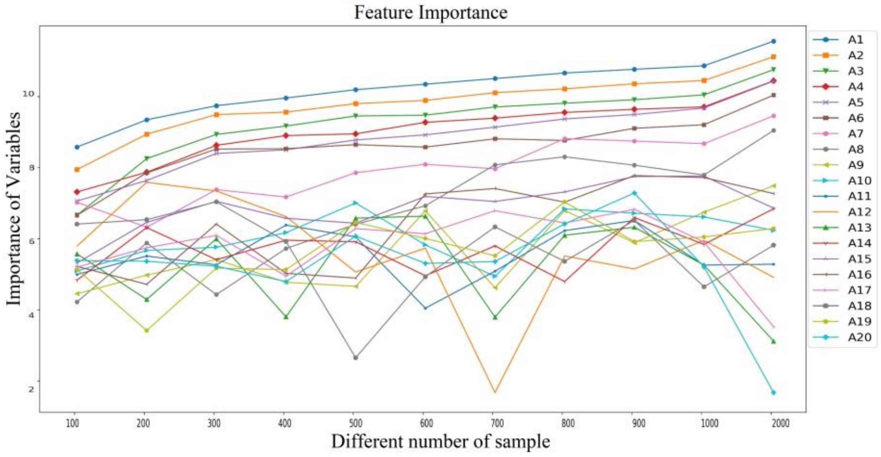


Fig 8. *Total Cover of each variable of XGBOOST with different sample size of function 5*

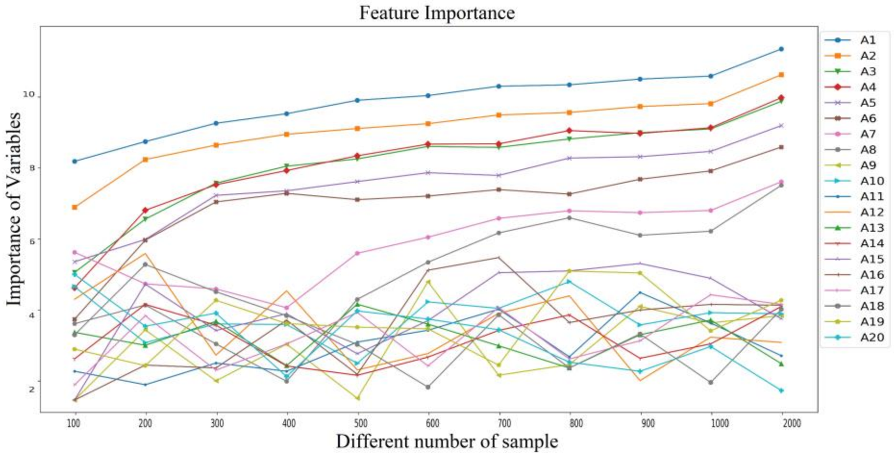


Fig 9. *Total gain of each variable of XGBOOST with different sample size of function 5*

When we compare these two graphs, we see that the total coverage requires larger sample sizes than the total gain to exhibit the important parameters of the example, namely A1 to A8. We also see that there does not seem to be any trend in the rank of the other variables when the sample size is increased. So, we used the total gain as measure to compare the use of XGBOOST and SVM on our problem.

To compare XGBOOST and SVM, we use box plot to see the median, standard deviation, maximum and minimum of feature importance of each algorithm with different sample sizes. We performed this process the 5 functions mentioned above, but here we just show the result for function 5:

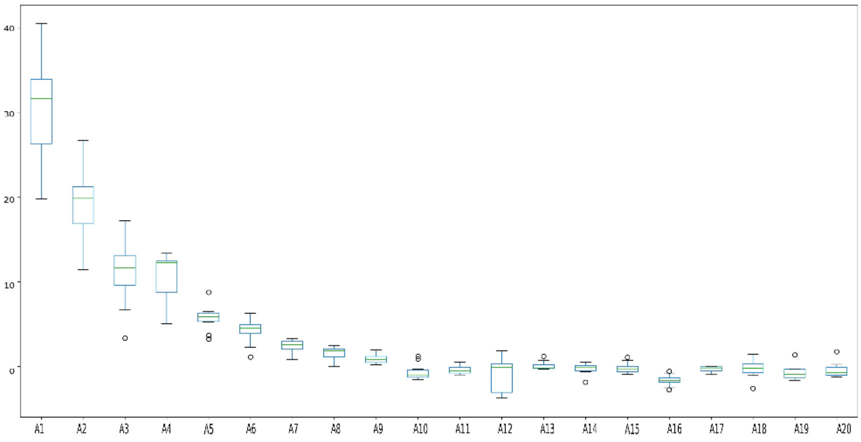


Fig 10. *SVM weights range with different sample size from 100 to 2000 of function 5*

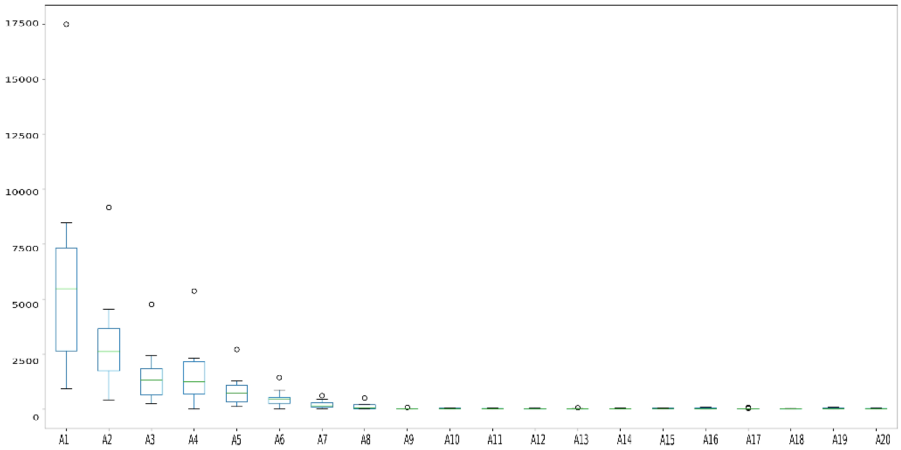


Fig 11. XGBOOST total gain with different sample size from 100 to 2000 of function 5

As you can see in these two figures variable of A1 to A8 have a higher value in compare to the A9 to A20, but for example A8 in XGBOOST doesn't to much overlap to non-important values, but in contrary A8 in SVM has overlap with different non-important variables like A9, A12, A18. This leaves us to speculate that XGBOOST can better distinguish between important and non-important features of our problem (the same king of result is obtained for our 5 examples and you can find more results on our github[1]). Another benefit of XGBOOST for the 5 examples, it was more robust to the non-important features. Because in box plot with different sample size standard deviation of non-important features compared to SVM was in a much smaller range and they were around the zero but in SVM range of non-important feature was not consistent and they change below and above the zero with different sample size.

With using Box plot or line chart, we can detect the non-important features also, in box plot we can detect them by removing the narrow boxes around the zero and in line chart with different sample size, non-important feature changes a lot but important features are more robust to these changes and they hold their ranking with different sample size.

Now we make the hypothesis that XGBOOST is more robust on non-important features, then problem that we deal with is that we don't know, which model with how many number of sample we can trust more because we deal with different sample size of the real system or simulated model, and we need a metric for this problem to evaluate the result of classification and also result of features selected from the model [30]. And because of we deal with different kind of data and different ratio of imbalanced dataset, we cannot say exactly how many data we need to have a good performance in the modeling and feature selection before training the model. Our solution for this problem is, at first, we build the model based on different sample size and then, with different

[1] https://github.com/nasergh/TRIZ-contradiction

measures, we check which one of these models can better modelized the function inside of the data.

For checking the performance, we use two measurements, AUC and PRC, and we show different scenarios of sampling that model can successfully find the important variables and also some example that it doesn't, and compare the measurement to find the best sample size.

Because we already know the important variables of each function, we can calculate the correctness of feature selection, that why we calculate the different measure for different function and we show them in ***Table 3.***

Table 3. *result of functions with different sample size (in percent)*

		100	**200**	**300**	**400**	**500**	**1000**
Function 1	AUC	100	100	100	100	100	100
	PRC	100	100	100	100	100	100
	Correct	100	100	100	100	100	100
Function 2	AUC	100	100	100	100	100	100
	PRC	100	100	100	100	100	100
	Correct	100	100	100	100	100	100
Function 3	AUC	97.86	99.97	100	100	100	100
	PRC	98.93	99.99	100	100	100	100
	Correct	80	100	100	100	100	100
Function 4	AUC	Nan	73.48	87.91	90.45	96.47	99.24
	PRC	Nan	4.12	18.09	6.21	63.92	61.30
	Correct	40	50	60	40	70	100
Function 5	AUC	83.70	99.50	99.80	100	99.80	99.90
	PRC	93.50	99.80	99.90	100	99.90	100
	Correct	62.50	87.50	87.50	87.50	87.50	100

With these results we can see that, when we increase the sample sizes, if PRC or AUC becomes stable and changing very smoothly then our sample size is large enough to have a good correctness (more than 70%) or in other word we can more trust to the result of feature importance values of the model and in other hand we have an accurate model of the data and model understand the relation between the variables much better. One can see that when we have higher AUC or PRC value, then value of feature important and ranking of them is also improved. For example, for the function 4, PRC decrease 11.88% when we changed the number of samples from 300 to 400 sample. Also, one can see is the correctness of feature selection decrease 20%, it lets us conjecture that there is a relationship between the model accuracy and features extracted from it.

It can also be seen that when there is a big jump in PRC and AUC values, and PRC value comes to more than 60% then it seems that we are very near to a sufficient number of sample size to find the feature important.

6. Discussion

in this paper our goal was founding the features important that are involved in contradiction and for that purpose we built artificial dataset with already define functions. At first, we compare different output of feature importance that can be extract from the XGBOOST model and we find out, total gain shows the feature important of each function with better ranking and also with small amount of data, total gain can show the feature important better than others. After that we compare the feature importance of SVM with XGBOOST to see which one is more robust in different functions and data's, then we see that XGBOOST result for non-important features is more stable and robust to the noise and they are very close to zero in compare to the SVM. Also, we see that in some functions XGBOOST with a smaller number of samples in compare to the SVM can give the right order of feature important. With this methodology we build an automatic system that extract the most important variable from the data.

In our experiment we see that if we deal with highly imbalance dataset then it's hard to find the feature important inside of the dataset, and also, it's hard to make the decision about how many numbers of samples is enough for model to give us more trustable feature importance. That why we evaluate the model with different measurement and we see that there is a relation between the feature importance extracted from the model and accuracy of the model. But this relation was not very clear, but can be used as a guide for the user to continue the improving the model and correctness of extracted feature important. With defined metrics we know that when model can understand the relation between the variables but we don't know with how many numbers of samples we will have high correctness in feature important. In the future, we will need to look more closely to this problem to find the other relationships between the correctness of the features extracted from the model and the accuracy of the model.

Another problem that we deal with it, was in imbalance dataset. We need at least 4 samples that show the minority class otherwise we cannot have a good feature selection and also accurate model that describe the relation and variable important inside of the dataset.

This methodology can help to find the important features or in other word action parameters that are involved in identifying the contradiction between the technical contradiction, and if we know the action parameters and their values that are involved in Technical contradiction then we can come back to the physical contradiction of the system and find the generalized technical contradictions. In next research, we will try to find the function that describe the technical contradiction, to see the values and relations between the action parameters.

7. Conclusion

In this paper, first we find out which feature importance of XGBOOST describe the important variable better then we compared the performance of SVM and XGBOOST algorithms for getting the important variables involved in generalized physical contradictions. In order to make this comparison and to use XGBOOST, it was necessary to seek for measures providing the important variables for XGBOOST algorithm. The results suggest that, first, XGBOOST is more powerful and robust in noisy dataset and can better detect the non-important features than SVM, second, XGBOOST can show the importance action parameters with a smaller number of data. In the next part, we show that PRC and AUC can give a clue to the user that how much model can understand the actual function and indirectly show how much model can successfully extract the important action parameters.

Acknowledgement

This research is carried out within the framework of the Offensive Sciences project number 13.11 " Virtual Innovative Real Time Factory " (VIRTFac) which benefits from the financial support of the Offensive Sciences programme of the Upper Rhine Trinational Metropolitan Region, the INTERREG V Upper Rhine programme and the European Regional Development Fund (ERDF) of the European Union.

8. References

[1] G. S. Altshuller, *Creativity as an exact science: the theory of the solution of inventive problems.* Gordon and Breach, 1984.

[2] S. Bach, R. De Guio, and G. Nathalie, ‘Combining discrete event simulation, data analysis, and TRIZ for fleet optimization’, *J. Eur. TRIZ Assoc. Innov.*, vol. 04, no. 02, pp. 47–61, 2017.

[3] F. Ben Moussa, R. Benmoussa, R. de Guio, S. Dubois, and I. Rasovska, ‘An algorithm for inventive problem solving coupled with optimization for solving inventive problems encountered in supply chains’, 2016, [Online]. Available: http://icube-publis.unistra.fr/4-BBdD16.

[4] F. Z. Ben Moussa, I. Rasovska, S. Dubois, R. D. Guio, and R. Benmoussa, ‘Reviewing the use of the theory of inventive problem solving (TRIZ) in green supply chain problems’, *J. Clean. Prod.*, vol. 142, pp. 2677–2692, 2017, doi: https://doi.org/10.1016/j.jclepro.2016.11.008.

[5] F. Z. BenMoussa, S. Dubois, R. De Guio, I. Rasovska, and R. Benmoussa, ‘Integrating the Theory of Inventive Problem Solving with Discrete Event Simulation in Supply Chain Management’, in *Automated Invention for Smart Industries*, Oct. 2018, pp. 330–347, doi: 10.1007/978-3-030-02456-7_27.

[6] L. Burgard, S. Dubois, R. de Guio, and I. Rasovska, ‘Sequential experimentation to perform the Analysis of Initial Situation’, 2011, [Online]. Available: http://icube-publis.unistra.fr/4-BDDR11.

[7] H. Chibane, S. Dubois, and R. D. Guio, ‘Automatic Extraction and Ranking of Systems of Contradictions Out of a Design of Experiments’, in *Automated Invention for Smart Industries*, Oct. 2018, pp. 276–289, doi: 10.1007/978-3-030-02456-7_23.

[8] S. Dubois, R. de Guio, and I. Rasovska, ‘From simulation to invention, beyond the pareto-frontier’, 2015, [Online]. Available: http://icube-publis.unistra.fr/4-DDR15.

[9] S. Dubois, T. Eltzer, and R. De Guio, ‘A dialectical based model coherent with inventive problems and optimization problems’, *Comput. Ind.*, vol. 60(8), pp. 575–583, Oct. 2009.

[10] S. Dubois, R. De Guio, A. Brouillon, and L. Angelo, ‘A Feedback on an Industrial Application of the FORMAT Methodology’, in *Automated Invention for Smart Industries*, Oct. 2018, pp. 290–301, doi: 10.1007/978-3-030-02456-7_24.

[11] I. Rasovska, R. de Guio, and S. Dubois, ‘Using dominance relation to identify relevant generalized technical contradictions in innovative design’, Oct. 2017, [Online]. Available: http://icube-publis.unistra.fr/4-RdD17.

[12] L. Lin, S. Dubois, R. de Guio, and I. Rasovska, ‘An exact algorithm to extract the generalized physical contradiction’, *Int. J. Interact. Des. Manuf.*, vol. 9, no. 3, pp. 185–191, 2015, doi: 10.1007/s12008-014-0250-3.

[13] L. Lin, I. Rasovska, R. de Guio, and S. Dubois, ‘Optimization Methods for Inventive Design’, in *TRIZ – The Theory of Inventive Problem Solving*, Springer., D. Cavallucci, Ed. Cavallucci, Denis, 2017, pp. 151–185.

[14] P. Parrend, F. Guigou, J. Navarro, A. Deruyver, and P. Collet, ‘Artificial Immune Ecosystems: the role of expert-based learning in artificial cognition’, in *ICAROB 2018| The International Conference on Artificial Life and Robotics*, Feb. 2018, p. 5, [Online]. Available: http://icube-publis.unistra.fr/4-PGND18a.

[15] S. Dubois, L. Lin, R. De Guio, I. Rasovska, and S. H. Christian Weber, ‘From Simulation to Invention, beyond the Pareto-Frontier’, Milan, Italy, 2015.

[16] S. Dubois, T. Eltzer, and R. De Guio, ‘A dialectical based model coherent with inventive and optimization problems’, *Comput. Ind.*, vol. 60, no. 8, pp. 575–583, Oct. 2009, doi: 10.1016/j.compind.2009.05.020.

[17] S. Dubois, I. Rasovska, and R. De Guio, ‘Interpretation of a general model for inventive problems, the generalized system of contradictions’, 2009.

[18] L. Lin, I. Rasovska, R. De Guio, and S. Dubois, ‘Algorithm for identifying generalized technical contradictions in experiments’, *J. Eur. Systèmes Autom. JESA*, vol. 47, no. 4–8, pp. 563–588, 2013.

[19] L. Lin, I. Rasovska, R. De Guio, and S. Dubois, ‘Optimization Methods for Inventive Design’, in *TRIZ – The Theory of Inventive Problem Solving*, D. Cavallucci, Ed. Cham: Springer International Publishing, 2017, pp. 151–185.

[20] L. Lin, S. Dubois, R. De Guio, and I. Rasovska, ‘An exact algorithm to extract the generalized physical contradiction’, *Int. J. Interact. Des. Manuf. IJIDeM*, vol. 9, no. 3, pp. 185–191, Nov. 2014, doi: 10.1007/s12008-014-0250-3.

[21] L. Lin, ‘Optimization methods for inventive design’, PhD Thesis, 2016.

[22] D. S. Madara, ‘Theory of inventive problem solving (TRIZ): his-story’, *IJISET - Int. J. Innov. Sci. Eng. Technol.*, vol. 2, no. 7, pp. 86–95, Jul. 2015.

[23] C.-W. Hsu, C.-C. Chang, C.-J. Lin, and others, *A practical guide to support vector classification*. Taipei, 2003.

[24] S. Bach, R. De Guio, and N. Gartiser, ‘Combining discrete event simulation, data analysis, and TRIZ for fleet optimization’, *J. Eur. TRIZ Assoc. Innov.*, vol. 4, no. 2, pp. 47–61, 2017.

[25] T. Chen and C. Guestrin, ‘Xgboost: A scalable tree boosting system’, in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[26] U. Malik, M. Barange, N. Ghannad, J. Saunier, and A. Pauchet, ‘A Generic Machine Learning Based Approach for Addressee Detection In Multiparty Interaction’, in *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*, 2019, pp. 119–126.

[27] A. Folleco, T. M. Khoshgoftaar, and A. Napolitano, ‘Comparison of four performance metrics for evaluating sampling techniques for low quality class-imbalanced data’, in *2008 Seventh International Conference on Machine Learning and Applications*, 2008, pp. 153–158.

[28] T. Saito and M. Rehmsmeier, ‘The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets’, *PloS One*, vol. 10, no. 3, 2015.

[29] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1. Springer series in statistics New York, 2001.

[30] N. Cristianini, J. Shawe-Taylor, and others, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.