# Type of Stochastic Dependence and its Impact on the Performance of Regression Type Classifiers

Olgierd Hryniewicz

Systems Research Institute
Polish Academy of Sciences
Warszawa, Poland
Email: hryniewi@ibspan.waw.pl

***Abstract*—Six regression type binary classifiers based on linear and logistic models have been evaluated using a complex simulation experiment. The classifiers were compared with respect to the robustness to unexpected changes of the models that describe data in training and test sets. The data used for this comparison were generated using different models describing their interdependence. This dependence was modeled by different copulas. The experiments revealed that the performance of considered classifiers strongly depends upon the type of copula. However, the simple logistic regression has appeared to be the best one in these circumstances. Thus, this classifier could be recommended for practitioners when the type of dependence may vary in time.**

***Keywords–Binary classification; Regression type classifiers; Copulas; Simulation of dependent data; Robustness.***

## I. Introduction

This paper is a significant extension of the conference paper "On the Robustness of Regression Type Classifiers" presented in the Proceedings of INTELLI'2015 conference, held in St. Julians, Malta [1], and is focused on rarely discussed aspects of classification problems. Classification algorithms are probably the most frequently used tools of data mining. The methods of their construction in the Artificial Intelligence (AI) community is known under the name of *supervised learning*. There are thousands of books and papers devoted to their theory and applications. Thomson Reuter's scientific database Web of Science displays information (as on 2016 February 24th) on nearly 400 papers with the phrase "classification algorithms" in the title, and nearly 7000 papers with this phrase in the topic. The information about theoretical foundations of classification algorithms can be found, e.g., in books by Duda et al. [2] and Hastie et al. [3]. Comprehensive description of application aspects of classification algorithms can be found in the book by Witten et al. [4].

The main problem with the evaluation of each, from among hundreds of already proposed, classifier is estimation of its quality characteristics. Japkowicz and Shah in their excellent book [5] write about two general approaches to this problem: *de facto* approach based on computing of many different quality characteristics, and *statistical* approach, in which unavoidable randomness of classification results is taken into account. The *de facto* approach can be used for any type of testing procedure, and is predominately used by the AI community. The applicability of the statistical approach is somewhat restricted, as the analyzed data should fulfill some requirements precisely described in terms of the theory of probability. These requirements are easily verified if we use for testing purposes artificially generated data. However, the usage of such data is not appreciated by the AI community, who prefers to use real-life benchmarks for evaluation purposes. When we use benchmark data for evaluation, the data used for the construction of an algorithm and the data used for its evaluation come from the same set of real-life values. In order to assure validity of comparisons different schemes of randomization, e.g., cross-validation techniques, are used. This approach is commonly accepted, and valid for the great majority of potential applications. It is, usually rightly, assumed that a classifier (in fact, the method of its construction) is of good quality if it performs well on many different benchmarks. However, in nearly every case (see, e.g., Hand [6]) it is assumed that the classifier is constructed and further used on *the same population* of classified objects. In some cases, however, this assumption may be questioned.

Robustness is well defined in statistics. According to Wikipedia, robust statistics "is a statistical technique that performs well even if its assumptions are somewhat violated by the true model, from which the data were generated". This definition of robustness can be directly applied to these methods of classification, which are based on well established statistical methodology, such as, e.g., regression. In general, however, many classification methods, such as, e.g., neural networks or decision trees, are not based (at least, directly) on statistical models. Therefore, in the machine learning community robustness is often understood somewhat differently, as the ability to perform well for many different sets of real data. David Hand, one of the most renowned researchers in the area of machine learning, in his overview paper [6] discusses consequences of breaking the assumption that the data in the design (training) set are randomly drawn from the same distribution as the points to be classified in the future. He gives references to some works related to this problem, and presents examples of problems encountered in the area of the credit scoring and banking industries. It has to be noted, however, that the number of papers devoted to the problem of robustness, understood as in [6], is rather small. For example, Japkowicz and Shah [5], while discussing this type of the robustness of classifiers, cite only the paper by Hand [6]. One can consider the concept of robustness in even more general sense, as to perform reasonably well when data are described or generated using different mathematical models. This understanding of robustness is close to the one used by the AI community, but is different to it as takes into account the knowledge about the mathematical models of analyzed data. In this paper we understand the concept of robustness in this, more general, sense.

Hryniewicz [7] [8] considers the case when binary classifiers are used for quality evaluation of items in production processes. In many cases of such processes, quality characteristics cannot be directly evaluated during production time. Sometimes it is impossible, when a testing procedure is destructive or impractical, or when a testing procedure is costly or lasts too long. In such cases, an appropriate classifier, which labels monitored items as "good" or "bad" is constructed using the data coming from specially designed (and usually costly) experiments, and then used in production practice. The situation does not rise any objections if the process, from which items used in the construction phase of a classification algorithm are taken is *the same* as a process, in which obtained classifiers are used. Hryniewicz [7], [8] has demonstrated that deterioration of such process may have detrimental effects on the quality of classification. Similar problems may be also encountered in other fields of applications. Consider, for example, a classifier that is used for the prediction of cancer recurrence who may change its quality characteristics when future patient will undergo a treatment, which was not used at the moment when this classifier was built.

The problems described in the previous paragraph may suggest that in the evaluation of classifiers we should add another dimension, namely the robustness to the change of population understood as the change of probability distributions that describe input variables in the classification process. It was the topic of the paper by Hryniewicz [1] whose work was focused on the analysis of robustness understood in this way. In our analysis we take into account in a more comprehensive way the impact of the type of stochastic dependence on the performance of classifiers. Therefore, we supplement the results already presented in [1] with new results whose aim is to present this impact.

At the moment the analysis of the robustness of classifiers, understood in a general way, can be achieved using artificially generated data, because appropriate, and widely known, benchmarks seem not to exist. Hryniewicz [1] analyzed the problem of robustness using software designed for the generation of complex nonlinear processes with statistically dependent data. This software has also been used for obtaining new results described in this paper. A detailed description of this software can be found in Section II. Similarly as in [1] we have evaluated binary classifiers whose construction is based on generalized linear models and regression techniques. In particular, we have analyzed classifiers based on

- Simple linear regression,
- Linear regression with interactions,
- Simple logistic regression,
- Logistic regression with interactions,
- Linear Discrimination Analysis with a symmetric decision criterion,
- Linear Discrimination Analysis with an asymmetric decision criterion.

We have assumed that the dependence between variables in our simulation model may be described by different copulas, characterized by different strength of dependence. The main goal of the research was twofold. First, as it is presented in [1], we have evaluated the robustness of the considered classifiers to shifts of the expected values of input variables (attributes). Second, we have tried to find if the strength and type of dependence influences performance of the considered classifiers. In contrast to the results published by other authors, we present the results of experiments performed in a strictly controlled environment that simulates conditions, which are significantly different from those usually assumed for the considered classification models.

This paper is organized similarly to its predecessor [1]. In Section II we describe considered models of data dependence, simulation software, and evaluated classifiers. Then, in Section III we describe used methods of evaluation. The most important results of experiments will be illustrated with examples in Section IV. Finally, in Section V we will conclude the experiments, and present the original results of this research.

## II. Description of simulation experiments

Except for few particular cases the problems described in the previous section cannot be solved analytically. Therefore, statistical simulations are widely accepted by the AI community as a sufficient tool for solving different problems of classification.

### A. Simulation software

Realization of the task formulated in Section I requires an implementation of a complex mathematical model in a form of simulation software. On the most general level, let us assume that a general mathematical model that describes dependence of input variables (predictors) with an output binary variable is a simple one. Let $Z_1, \ldots, Z_p$ be $p$ output characteristics whose values are not directly observed in an experiment. Assume now that these values should be predicted using observations $X_1, \ldots, X_k$ of $k$ predictors. This problem is easy to solve if we assume that we know the joint probability distribution of input and output variables, i.e., the probability distribution of a combined vector $(Z_1, \ldots, Z_p, X_1, \ldots, X_k)$. According to the famous Sklar's theorem this distribution is unequivocally described by a $(p+k)$-dimensional copula, and marginal probability distributions of $Z_1, \ldots, Z_p$ and $X_1, \ldots, X_k$, respectively. Such a general model is hardly applicable, as only two-dimensional copulas $C(u, v)$ are widely used in practice. Therefore, our simulation software should be based on a model, which is simpler and more easy for practical interpretation. In this research we have used a hierarchical 3-level model, originally proposed in [7]. On the top level of this model there is an auxiliary one-dimensional real-valued variable $T$. This value is transformed to a binary one (in which we are interested in) by means of the following transformation

$$Z_t = \begin{cases} 0 & , \quad T \geq t \\ 1 & , \quad T < t \end{cases} \tag{1}$$

The instances with the value $Z_t = 1$ we will call "positive cases" or "Positives", and the instances with the value $Z_t = 0$ we will call "negative cases" or "Negatives". This model has a direct interpretation in the case considered by Hryniewicz [7] who modeled a monitoring of a production process with indirectly observable quality characteristic. The first level of our model describes the predictors $X_1, \ldots, X_k$. In order to simplify simulations we assume that consecutive $k-1$ pairs of predictors $(X_i, X_{i+1}), i = 1, \dots, k-1$ are described by $k-1$ copulas $C_{i,i+1}(F_i(X_i), F_{i+1}(X_{i+1})), i = 1, \dots, k-1$, where $F_1(X_1), \dots, F_k(X_k)$ are the cumulative probability functions of the marginal distribution of the predictors. In order to simulate the input variables we have to assume the type of the proposed copulas, and the strength of dependence between the pairs of random variables whose joint two-dimensional probability distributions are described by these copulas. In the AI community Pearson's coefficient of correlation $r$ is often used as the measure of dependence. Unfortunately, its applicability is limited to the case of the classical multivariate normal distribution, or - in certain circumstances - to the case of the multivariate elliptic distributions (for more information see [9]). When dependent random variables cannot be described by such a model, and it is not an unusual case in practice, we propose to use Kendall's coefficient of association $\tau$ defined, in its population version in terms of copulas, as (see [10])

$$\tau(X, Y) = 4 \int\int_{[0,1]^2} C(u, v) dC(u, v) - 1. \tag{2}$$

Numerical comparisons of the values of Pearson's $r$, Kendall's $\tau$, and - another popular nonparametric measure of dependence - Spearman's $\rho$ are presented in [11], and show that the usage of Pearson's $r$ in the analysis of data that cannot be described by the normal distribution may lead to wrong conclusions, especially in the case of negative dependence. Therefore, Kendall's $\tau$ is, in such cases, a much better measure of dependence.

In order to have a more realistic model for simulation purposes, it was proposed in [7] to use an in-between second level of latent (hidden) variables $H_1, \dots, H_k$ described by cumulative probability functions $F_{H1}(h_1), \dots, F_{Hk}(h_k)$. Each hidden variable $H_i$ is associated with the predictor variable $X_i$, and its fictitious realizations are measured on the same scale as the predicted continuous random variable $T$. The dependence between $H_i$ and $X_i$ is described by a copula $C_{i,i}(F_{Hi}(H_i), F_i(X_i))$. Moreover, in our model we assume that there exists a certain linear relationship between the expected value of $H_i$ and the expected value of $X_i$. This assumption is needed if we want to model the effects of the shifts in the expected values of the predictors on the expected value of the predicted auxiliary variable $T$, which is related to the hidden variables by a certain, possibly nonlinear, function

$$T = f(H_1, \dots, H_k). \tag{3}$$

In real circumstances, such as those described in [7], the probability distribution of $T$, and hence the probability distribution of $Z_t$, can be observed only in specially designed experiments. The results of such experiments can be viewed upon as data sets coming from supervised learning experiments. In our research we simulate similar experiments, and we use actual (i.e., generated by our software) and predicted (i.e., the results generated by classifiers) binary outputs for constructing and testing, several, say $s$, classifiers, $K_1, \dots, K_s$, each of the form

$$Z'_t = K(X_1, \dots, X_k). \tag{4}$$

The mathematical model described above was implemented in a software system written in FORTRAN. The reason for using this old programming language was twofold. First, because of a great amount of needed computations the usage of popular among statisticians interpreted languages like R is completely inefficient. Second, because of the long history of the usage of this programming language in statistics many numerically effective procedures are widely available.

### B. *Description of the experiment*

In this paper, we describe the results for only four input variables. This limited number of input variables may be justified by findings of Hand [6] who noticed that in many real-life problems of classification only few predictors (attributes) have real impact on the results of classification. Another reason for making this restriction is limited time of computations. One has to note that even in this restricted model one run of Monte Carlo simulations may last several days of continuous work of a fast PC computer. The simulation process described in this paper consists of three parts. First, a stream of data points, i.e., the values of predictors, the values of hidden variables, the value of the unobserved auxiliary output variable, and the observed output binary variable are generated. Next, these simulated data serve as training data sets for building several classifiers. Finally, test data sets are generated, and used for the evaluation of considered classification (prediction) algorithms.

In our simulation experiment the probability distributions of predictors defined by a user on the first level of the model can be chosen from a set of five distributions: uniform, normal, exponential, Weibull, and log-normal. For the second level of the model a user can choose the probability distributions of the hidden variables from a set of distributions, that are defined on the positive part of the real line: exponential, Weibull, and log-normal. The information about these probability distributions can be found in any textbook on probability and statistics.

The dependence between the pairs of predictors, and between predictors and associated hidden variables, can be described by the following copulas:

- independent

$$C(u, v) = uv, \tag{5}$$

- Normal (Gaussian)

$$C(u, v; \rho) = \Phi_N(\Phi^{-1}(u), \Phi^{-1}(v); \rho) \tag{6}$$

where $\Phi_N(u, v)$ is the cumulative probability distribution function of the bivariate normal distribution, and $\Phi^{-1}(u)$ is the inverse of the cumulative probability function of the univariate normal distribution (the quantile function). Parameter $\rho$ is equal to the well known Pearson's coefficient of linear correlation $r$ only in the case of normal marginal probability distributions,

- Clayton

$$C(u, v) = \max\left(\left[u^{-\alpha} + v^{-\alpha} - 1\right]^{-1/\alpha}, 0\right), \alpha \in [-1, \infty) \setminus 0, \tag{7}$$

- Frank

$$\begin{aligned} C(u, v) = -\tfrac{1}{\alpha} \ln\left(1 + \tfrac{(e^{-\alpha u}-1)(e^{-\alpha v}-1)}{e^{-\alpha}-1}\right), \\ \alpha \in (-\infty, \infty) \setminus 0, \end{aligned} \tag{8}$$

- Gumbel

$$C(u,v) = \exp\left(-\left[(-\ln u)^{1+\alpha} + (-\ln v)^{1+\alpha}\right]^{\frac{1}{1+\alpha}}\right), \quad \alpha \in (0,\infty), \tag{9}$$

used only for positive dependencies, and

- Fairlie-Gumbel-Morgenstern (FGM)

$$C(u_1,u_2;\theta) = u_1u_2 + \theta u_1u_2(1-u_1)(1-u_2), |\theta| \leq 1, \tag{10}$$

used for modeling only weak dependencies. The detailed description of these copulas can be found, e.g., in [10]. The strength of this dependence is defined by the value of Kendall's coefficient of association $\tau$, calculated for each of the considered copulas using (2). The expected values of the distributions of the hidden variables in this simulation model depend in a linear way on the values of its related predictors. At the next stage of simulation, hidden random variables are transformed to the auxiliary output random variable $T$. The relation between the hidden variables and $T$ is strongly non-linear, and is described by operators of a "min-max" type. Finally, the auxiliary output random variable $T$ is transformed to the binary output variable, which is used for classification purposes. The proposed model allows to generate data with great variety of properties (non-linear dependence of a different strength, different probability distributions, etc.) that are significantly different from those usually assummed for linear regression models.

The scheme of the simulation of a data point, for an exemplary set of input parameters (probability distributions, copulas, and values of Kendall's $\tau$), is presented in Figure 1. The values of four input attributes are generated, respectively, from the normal, exponential, logarithmic normal, and Weibull distributions. The generated values are statistically dependent, and the dependencies are described, respectively, by the following copulas: Clayton (with $\tau = 0.8$), Normal (with $\tau = -0.8$), and Frank (with $\tau = 0.8$). Then, for each input attribute the system generates an unobserved (hidden) value. These hidden values are generated, respectively, from the logarithmic normal, exponential, exponential, and Weibull distributions. The parameters of these distributions depend in a linear way upon the values of the respective input attributes (this dependence is not depicted in Figure 1). Moreover, they are also statistically dependent upon the values of the generated input attributes, and these dependencies are described, respectively, by the following copulas: Normal (with $\tau = -0.8$), Frank (with $\tau = 0.9$), Gumbel (with $\tau = -0.9$), and Normal (with $\tau = -0.8$), and Clayton (with $\tau = -0.8$). Finally, the real-valued output is calculated using the formula depicted in Figure 1, and this value is transformed, by using (1), to the binary output variable. The generated 5-tuple (4 input attributes, and a binary output value) describes one point in the training data set. The points of the test set are generated similarly, with the same or different (when robustness is evaluated) parameters of the model. The number of input variables (four) has been chosen in accordance with the opinion presented in [6] that in real situations the number of attributes, which really influence quality characteristics of a classifier is usually small.

Several types of classifiers have been implemented in our simulation program. The classifiers are built using samples of size $n_t$ of training data consisted of the vectors of the values of predictors $(x_1, x_2, x_3, x_4)$, and the actual value of the assigned class. In this paper, we consider only six of them, which represent three different general approaches to the classification problem.

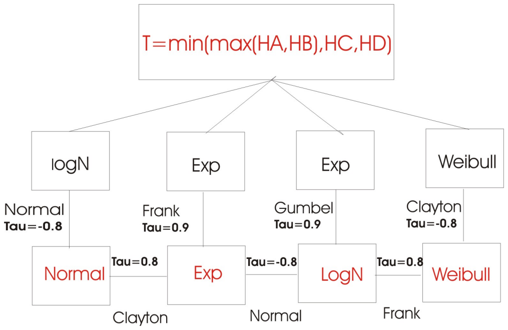

Figure 1. An exemplary scheme of the simulation of a data point

*Binary linear regression.* The first considered classifier is a simple (of the first order) binary linear regression (LINREG4). We label the considered classes by 0 and 1, respectively, and consider these labels as real numbers, treating them as observations of a real dependent variable in the linear regression model of the following form:

$$R_4 = w_0 + w_1 * X_1 + w_2 * X_2 + w_3 * X_3 + w_4 * X_4, \tag{11}$$

where $R$ is the predicted class of an item described by explanatory variables $X_1, X_2, X_3, X_4$, and $w_1, w_2, w_3, w_4, w_0$ are respective coefficients of the regression equation estimated from a training set of $n_t$ elements. The value of $R$ estimated from (11) is a real number, so an additional requirement is needed for the final classification (e.g., if $R < 0,5$ an item is classified as belonging to the class 0, and to the class 1 otherwise). The second considered classifier is also a linear one, but with additional variables describing interactions of the second order between the input variables (LINREG14). The regression function (of the second order) in this case is the following

$$\begin{aligned} R_{14} = \quad & w_0 + w_1 * X_1 + \cdots + w_5 * X_1^2 + \cdots + \\ & w_9 * X_1 * X_2 + \cdots + w_{14} * X_3 * X_4. \end{aligned} \tag{12}$$

The main advantage of these two classifiers is their simplicity. Moreover, the classical linear regression is implemented in all spreadsheets, such as, e.g., MS Excel. For this reason we have chosen these classifiers as the easiest to implement in practice without any specialized software.

*Logistic regression.* The next two classifiers are built using a generalized linear regression model, namely the logistic regression. The logistic regression is recommended by many authors (see, e.g., [3]) as the best regression tool for the analysis of discrete data. In this model the dependence of the output $R_L$ upon the input variables is modeled by the logistic function

$$R_L = \frac{1}{1 + exp(-f(X_1, \ldots, X_4))}, \tag{13}$$

where the function $f(X_1, \ldots, X_4)$ is described either by the right side of (11) of the LOGREG4 model, or by the right side of (12) of the LOGREG14 model. Unfortunately, the implementation of the logistic regression is not as simple as in the case of the linear regression. The estimation of its parameters requires the usage of numerical procedures that are implemented in specialized software (available, e.g., in the WEKA package).

*Linear Discriminant Analysis (LDA)*. The last two classifiers implement the LDA introduced by Fisher, and described in many textbooks on multivariate statistical analysis and data mining (see, e.g., [3]). This method is historically the first classification method used in practice, and according to [6] its efficiency has been proved empirically by many authors. In the LDA statistical data are projected on a certain hyperplane estimated from the training data. New data points, projected on this hyperplane, which are closer to the mean value of the projected on this hyperplane training data representing the class 0 than to the mean value of training data representing the remaining class 1 are classified to the class 0. Otherwise, they are classified to the class 1. The equation of the hyperplane is given by the following formula:

$$L = y_1 * X_1 + y_2 * X_2 + y_3 * X_3 + y_4 * X_4 + y_0, \qquad (14)$$

where $L$ is the value of the transformed data point calculated using the values of the explanatory variables $X_1, X_2, X_3, X_4$, and $y_1, y_2, y_3, y_4, y_0$ are respective coefficients of the LDA equation estimated from a training set. If $Z_L$ denote the decision point, a new item is classified to the class 0 if $L \leq Z_L$, and to the class 1 otherwise. The LDA may not perform well in the case of imbalanced data. Therefore, in our simulation we implemented two methods of the calculation of $Z_L$. First, the classical one (LDA-SYM), when this point is just the average of the mean values of the transformed data points from the training set that belonged to the class 0 and the class 1, respectively. Second, an asymmetric one (LDA-ASYM), recommended for the analysis of imbalanced data sets, where $Z_L$ is located asymmetrically between the two mean values mentioned above, depending upon the number of items belonging to each class in the test set. The calculation of the LDA equation (14) is not so simple. However, it can be done using basic versions of many statistical packages such as SPSS, STATISTICA, etc. Moreover, the LDA problem can be reformulated in terms of a simple linear regression, so the statistical tools available in spreadsheets may also be used for computations.

## III. Evaluation of binary classifiers

Proper evaluation of binary classifiers is not as simple as it looks like. If we do not consider any costs of misclassification the whole information about the quality of classifiers is contained in the so called confusion matrix, presented in Table I [5].

TABLE I. Confusion matrix

| | Pred_Negative | Pred_Positive | |
|---|---|---|---|
| Act_Negative | True negative (TN) | False positive (FP) | N=TN+FP |
| Act_Positive | False negative (FN) | True positive (TP) | P=FN+TP |

All measures of the quality of classifiers are built using the information contained in this matrix. A comprehensive overview of these measures can be found in many sources such as, e.g., Chapter 3 of the book by Japkowicz and Shakh [5]. The most frequently used measure is *Accuracy*

$$Acc = \frac{TN + TP}{N + P} \qquad (15)$$

It estimates the probability of correct classification. However, in certain circumstances (e.g., when classes are imbalanced) this measure does not let to discriminate the quality of different classifiers. This happens to be the case in experiments described in this paper.

Other popular and important measures, such as

- *Precision*

$$Prec = \frac{TP}{TP + FP}, \qquad (16)$$

- *Sensitivity* or *Recall*

$$Sens = \frac{TP}{TP + FN}, \qquad (17)$$

- *Specificity*

$$Spec = \frac{TN}{FP + TN}, \qquad (18)$$

describe only certain features of binary classifiers. For example, high values of *Precision* in statistical terms are equivalent to low values of type I classification error when "Positives" are considered as the relevant class. Similarly, high values *Sensitivity* in statistical terms are equivalent to low values of type II classification error. When quality of the classification of "Negatives" is also worth of consideration, one has to take into account the value of *Specificity*.

In the performed experiment we used all these measures for the evaluation purposes. However, in this paper we present the analysis of two aggregate measures recommended for the evaluation of performance especially in presence of imbalanced data. First of these measures is *F1 score* (or *F1 measure*), defined as the harmonic average of *Precision* and *Sensitivity*, and calculated using the following formula

$$F1 = \frac{2TP}{2TP + FP + FN}. \qquad (19)$$

Low values of this measure indicate that a classifier has a large value of at least one of type I or type II errors. Second aggregate measure is known as *G-mean*, defined as the geometric mean of *Sensitivity* and *Specificity*, and calculated as

$$G = \sqrt{\frac{TP * TN}{((TP + FN)(FP + TN)}}. \qquad (20)$$

This measure is recommended for the evaluation of classifiers for highly imbalanced data when percentages of "Positives" and "Negatives" are significantly different, as it was the case in our experiments. It has to be noted that a popular among AI specialists measures such as ROC or AUC cannot be applied in our comparisons, as all considered classifiers are based on the same linear model, and are characterized by the same (or nearly the same) ROC characteristics.

## IV. Results of experiments

The simulation system described in Section II was used in many experiments with the aim to evaluate different binary classifiers. In this paper, we describe only one of them. In each instance of this particular experiment we simulated 50 runs, each consisted of one training set of 100 elements and 100 test sets of 1000 elements each. This small size of a training set was chosen in order to compare the results of simulations with those described in [7], [8], where it had a particular practical meaning. In each instance of the experiment, we used the same type of a copula for the description of all dependent random variables (in other experiments, not described in this paper, we used different copulas in one considered model). The strength of dependence was categorized into 6 categories: strong positive (Sp), medium positive (Mp), weak positive (Wp), weak negative (Wn), medium negative (Mn), and strong negative (Sn). For the Sp category the value of Kendall's $\tau$ was randomly chosen for each training set from the interval $[0.7, 0.9]$. The respective intervals for the remaining categories were the following: $[0.4, 0.6]$ for Mp, $[0., 0.2]$ for Wp, $[-0.2, 0.]$ for Wn, $[-0.6, -0.4]$ for Mn, and $[-0.9, -0.7]$ for Sn. For each of the simulated 50 training sets the expected values of input variables (predictors) varied randomly in certain intervals. The simulated training sets were used for the construction of six classifiers described in Section II. For all test sets in one simulation run the description of the dependence between considered random variables (i.e., the copula, and the set of the values of Kendall's $\tau$) was the same as in the respective training set. However, in choosing the expected values of the input variables (predictors) we considered two cases. In the first case, these expected values were the same as in the training set. Thus, the test sets were simulated using the same model as the respective training set. In other words, the considered classifiers were evaluated, in this case, on data generated by the same model as it had been used for their construction. In the second case, the expected values of the input variables used in the generation of test sets were *different* than the values used in the generation of the respective training sets. Those different values were chosen randomly around the values used for the generation of the training sets (by maximum ±30%).

The presentation of the obtained results let us start with the analysis of the influence of the type of a copula describing the type of dependence on the *Accuracy* (i.e., fraction of correctly classified objects) of considered classifiers, which is the most frequently used quality characteristics of classification. In Table II we present the obtained average values of *Accuracy* for 4 different copulas, and the strength of dependence belonging to the category Mp. We can see that the quality of the considered classifiers for a given copula is similar. Only the asymmetric LDA classifier is visibly worse. However, this quality is different for different types of copulas. This seems to be a very important finding, as the type of dependence is rarely (if ever) considered in the evaluation of classifiers. In the case described in Table II the observed (marginal) probability distributions are the same, and the estimates of the strength of dependence are also the same. Nevertheless, the accuracy of classification is visibly different, depending upon the type of dependence defined by the respective copula.

TABLE II. Average *Accuracy*. The same model for training and test sets. Medium positive dependence

| *Classifier* | *Normal* | *Clayton* | *Gumbel* | *Frank* |
|---|---|---|---|---|
| LINREG4 | 0.769 | 0.835 | 0.741 | 0.752 |
| LINREG14 | 0.769 | 0.833 | 0.735 | 0.751 |
| LOGREG4 | 0.789 | 0.849 | 0.757 | 0.773 |
| LOGREG14 | 0.769 | 0.832 | 0.736 | 0.751 |
| LDA-SYM | 0.741 | 0.765 | 0.729 | 0.729 |
| LDA-ASYM | 0.697 | 0.732 | 0.683 | 0.685 |

From Table II we can see that the highest fraction of correctly classified objects appears when data are generated by the Clayton copula, and the LOGREG4 classifier is the best one for all considered types of dependency, described by four considered copulas.

Now, let us consider the case of the similar (of medium strength) dependence, but a negative one. In Table III we compare the values of Accuracy for three copulas: Normal, Clayton, and Frank (the Gumbel copula does not allow negative dependence).

TABLE III. Average *Accuracy*. The same model for training and test sets. Medium negative dependence

| *Classifier* | *Normal* | *Clayton* | *Frank* |
|---|---|---|---|
| LINREG4 | 0.356 | 0.279 | 0.321 |
| LINREG14 | 0.432 | 0.411 | 0.404 |
| LOGREG4 | 0.424 | 0.333 | 0.379 |
| LOGREG14 | 0.436 | 0.422 | 0.408 |
| LDA-SYM | 0.428 | 0.438 | 0.460 |
| LDA-ASYM | 0.483 | 0.472 | 0.486 |

From Table III we see that in the case of negative dependence the situation is totally different in comparison to the case of positive dependence of the same (in absolute values) strength. For classifiers based on linear and logistic regression models the best results of classification are observed when data are described by the Normal copula. Moreover, classifiers that use the linear model with interactions perform much better than the simple ones. However, when data are analysed using classifiers based on linear discriminant analysis the best results are observed when they are described by the Frank copula. It is worth to note that in the considered case of negative dependence the LDA-ASYM classifier is visibly the best one.

It is a well known fact that for imbalanced classes (i.e., when objects belonging to one of the two considered classes, usually the "Positives", appear much less frequently than the objects belonging to the second class) *Accuracy* may not be a good quality characteristic. In such a case, an aggregate characteristics, such as, e.g., *F1 score*, are used for evaluation purposes. The results of such evaluation (averaged for the same data!) are presented in Table IV.

TABLE IV. Average *F1 score*. The same model for training and test sets. Medium positive dependence

| *Classifier* | *Normal* | *Clayton* | *Gumbel* | *Frank* |
|---|---|---|---|---|
| LINREG4 | 0.358 | 0.561 | 0.266 | 0.324 |
| LINREG14 | 0.490 | 0.623 | 0.419 | 0.472 |
| LOGREG4 | 0.537 | 0.662 | 0.464 | 0.509 |
| LOGREG14 | 0.500 | 0.630 | 0.430 | 0.487 |
| LDA-SYM | 0.101 | 0.061 | 0.072 | 0.089 |
| LDA-ASYM | 0.549 | 0.598 | 0.484 | 0.562 |

It is evident that the situation in this case becomes quite different. First of all, we can see unacceptably low values of the *F1 score* for the symmetric LDA classifier. Despite its quite good accuracy (see Table II) classification errors of this classifier are completely imbalanced. As the matter of fact, the precision of this classifier was good, but its sensitivity was really very low. The variability of the *F1* score observed in Table IV is much greater than the variability of the *Accuracy*. It means that for different copulas the quality of considered classifiers measured by the *F1 score* may be significantly different. Moreover, if we look simultaneously on Tables II and IV, we can see that the simple logistic regression classifier seems to be quite visibly the best when it classifies data generated by the same model as it had been used for the generation of the training set.

The situation completely changes when consider the case of negative dependence, presented in Table V.

TABLE V. AVERAGE *F1 score*. THE SAME MODEL FOR TRAINING AND TEST SETS. MEDIUM NEGATIVE DEPENDENCE

| *Classifier* | *Normal* | *Clayton* | *Frank* |
|---|---|---|---|
| LINREG4 | 0.356 | 0.279 | 0.321 |
| LINREG14 | 0.432 | 0.411 | 0.404 |
| LOGREG4 | 0.424 | 0.333 | 0.379 |
| LOGREG14 | 0.436 | 0.422 | 0.408 |
| LDA-SYM | 0.428 | 0.438 | 0.460 |
| LDA-ASYM | 0.483 | 0.472 | 0.486 |

The observed values of the *F1 score* behave similarly to the case observed for *Accuracy*. Both classifiers built using the discriminant analysis (LDA-SYM and LDA-ASYM) perform much better than classifiers built on linear and logistic regression. What is interesting, however, that the values of the *F1 score* of the LDA-ASYM classifier (the best one!) observed for different copulas are similar. It means that in case of negative dependence this classifier is robust against possible variations of the type of dependence.

A close look at the definition of the *F1 score* reveals that this quality characteristic is related to the classification of “Positives”, and does not take into account the quality of classification of “Negatives”, which usually form a much more numerous class. So, in the next step of our analysis let us examine the impact of the strength of dependence on the performance of considered classifiers evaluated using the *G-mean*. From the definition of this characteristic one can see that quality of classification of both “Positives” and “Negatives” is taken into account in this case. In Tables VI– IX we present a similar, as above, comparison for four cases: two levels of the strength of dependence of both positive and negative sign.

TABLE VI. AVERAGE *G-mean*. THE SAME MODEL FOR TRAINING AND TEST SETS. STRONG POSITIVE DEPENDENCE

| *Classifier* | *Normal* | *Clayton* | *Gumbel* | *Frank* |
|---|---|---|---|---|
| LINREG4 | 0.806 | 0.860 | 0.796 | 0.820 |
| LINREG14 | 0.834 | 0.893 | 0.831 | 0.847 |
| LOGREG4 | 0.868 | 0.918 | 0.858 | 0.877 |
| LOGREG14 | 0.835 | 0.896 | 0.831 | 0.847 |
| LDA-SYM | 0.055 | 0.037 | 0.058 | 0.063 |
| LDA-ASYM | 0.848 | 0.874 | 0.796 | 0.844 |

TABLE VII. AVERAGE *G-mean*. THE SAME MODEL FOR TRAINING AND TEST SETS. MEDIUM POSITIVE DEPENDENCE

| *Classifier* | *Normal* | *Clayton* | *Gumbel* | *Frank* |
|---|---|---|---|---|
| LINREG4 | 0.513 | 0.665 | 0.434 | 0.496 |
| LINREG14 | 0.625 | 0.725 | 0.567 | 0.617 |
| LOGREG4 | 0.665 | 0.758 | 0.602 | 0.642 |
| LOGREG14 | 0.634 | 0.736 | 0.576 | 0.629 |
| LDA-SYM | 0.281 | 0.195 | 0.261 | 0.287 |
| LDA-ASYM | 0.699 | 0.756 | 0.651 | 0.704 |

TABLE VIII. AVERAGE *G-mean*. THE SAME MODEL FOR TRAINING AND TEST SETS. STRONG NEGATIVE DEPENDENCE

| *Classifier* | *Normal* | *Clayton* | *Frank* |
|---|---|---|---|
| LINREG4 | 0.708 | 0.637 | 0.712 |
| LINREG14 | 0.809 | 0.774 | 0.785 |
| LOGREG4 | 0.815 | 0.731 | 0.790 |
| LOGREG14 | 0.809 | 0.775 | 0.785 |
| LDA-SYM | 0.309 | 0.180 | 0.234 |
| LDA-ASYM | 0.370 | 0.305 | 0.300 |

The results presented in Tables VI– IX show a rather complex picture. In the case of strong positive dependence ($\tau \in [0.7, 0.9]$) between all variables the quality of classification, measured using the *G-mean*, is consistently the highest when dependencies are described by the Clayton copula. However, the differences between considered copulas are not very strong. The best results are observed for the LOGREG4 classifier based on the simple (i.e., without interactions) logistic regression. When the strength of positive dependence is weaker ($\tau \in [0.4, 0.6]$) the data generated by the Clayton copula are still classified in the best way, but in this case the best classifier is the LDA-ASYM, based on Fisher’s linear discrimination model with asymmetric decision criterion. The situation changes dramatically when the dependence is negative. In both considered cases of strong negative ($\tau \in [-0.9, -0.7]$) and medium negative ($\tau \in [-0.6, -0.4]$) dependencies the best results of classification are observed for the Normal (Gaussian) copula. For the Clayton copula (the best in case of positive dependencies) the observed quality is the worst. What is also very important that in the case of negative dependencies none of the considered classifiers is the best one. However, the classifiers based on logistic regression seem to be more stable, as their performance does not depend so visibly on the strength of dependence between observed (predictors) and hidden variables.

Let us now consider an interesting case when the model of data in test sets is *different* from that of training data. In reality, it means that a classifier is used on data described by a different probability distribution than the data used during its construction. In Tables X– XI we present average values of the *Accuracy*, and in Tables XII– XIII we present average values of the *F1 score*, when the expected values of the input variables in the test sets have been randomly shifted around the values used for the generation of the training sets (by maximum ±30%).

Similar results for the case of the *G-mean* are presented in Tables XIV– XV.

As we can expect, the values of quality indices in this case

TABLE IX. AVERAGE *G-mean*. THE SAME MODEL FOR TRAINING AND TEST SETS. MEDIUM NEGATIVE DEPENDENCE

| *Classifier* | *Normal* | *Clayton* | *Frank* |
|---|---|---|---|
| LINREG4 | 0.488 | 0.431 | 0.464 |
| LINREG14 | 0.566 | 0.557 | 0.545 |
| LOGREG4 | 0.553 | 0.482 | 0.519 |
| LOGREG14 | 0.572 | 0.568 | 0.550 |
| LDA-SYM | 0.566 | 0.458 | 0.575 |
| LDA-ASYM | 0.605 | 0.527 | 0.592 |

TABLE X. Average *Accuracy*. Different models for training and test sets. Medium positive dependence

| *Classifier* | *Normal* | *Clayton* | *Gumbel* | *Frank* |
|---|---|---|---|---|
| LINREG4 | 0.730 | 0.773 | 0.705 | 0.728 |
| LINREG14 | 0.678 | 0.741 | 0.670 | 0.687 |
| LOGREG4 | 0.759 | 0.809 | 0.725 | 0.749 |
| LOGREG14 | 0.680 | 0.728 | 0.669 | 0.683 |
| LDA-SYM | 0.745 | 0.775 | 0.730 | 0.731 |
| LDA-ASYM | 0.643 | 0.672 | 0.631 | 0.646 |

TABLE XI. Average *Accuracy*. Different models for training and test sets. Medium negative dependence

| *Classifier* | *Normal* | *Clayton* | *Frank* |
|---|---|---|---|
| LINREG4 | 0.765 | 0.738 | 0.757 |
| LINREG14 | 0.718 | 0.669 | 0.701 |
| LOGREG4 | 0.759 | 0.736 | 0.755 |
| LOGREG14 | 0.717 | 0.666 | 0.693 |
| LDA-SYM | 0.527 | 0.423 | 0.529 |
| LDA-ASYM | 0.537 | 0.501 | 0.544 |

TABLE XII. Average *F1 score*. Different models for training and test sets. Medium positive dependence

| *Classifier* | *Normal* | *Clayton* | *Gumbel* | *Frank* |
|---|---|---|---|---|
| LINREG4 | 0.317 | 0.440 | 0.259 | 0.300 |
| LINREG14 | 0.443 | 0.519 | 0.356 | 0.420 |
| LOGREG4 | 0.482 | 0.577 | 0.406 | 0.451 |
| LOGREG14 | 0.452 | 0.528 | 0.367 | 0.430 |
| LDA-SYM | 0.138 | 0.126 | 0.101 | 0.124 |
| LDA-ASYM | 0.470 | 0.524 | 0.411 | 0.484 |

TABLE XIII. Average *F1 score*. Different models for training and test sets. Medium negative dependence

| *Classifier* | *Normal* | *Clayton* | *Frank* |
|---|---|---|---|
| LINREG4 | 0.340 | 0.274 | 0.308 |
| LINREG14 | 0.367 | 0.394 | 0.372 |
| LOGREG4 | 0.399 | 0.320 | 0.358 |
| LOGREG14 | 0.365 | 0.399 | 0.373 |
| LDA-SYM | 0.427 | 0.434 | 0.457 |
| LDA-ASYM | 0.470 | 0.473 | 0.486 |

TABLE XIV. Average *G-mean*. Different models for training and test sets. Medium positive dependence

| *Classifier* | *Normal* | *Clayton* | *Gumbel* | *Frank* |
|---|---|---|---|---|
| LINREG4 | 0.540 | 0.632 | 0.494 | 0.525 |
| LINREG14 | 0.626 | 0.685 | 0.561 | 0.610 |
| LOGREG4 | 0.652 | 0.723 | 0.599 | 0.629 |
| LOGREG14 | 0.631 | 0.692 | 0.567 | 0.615 |
| LDA-SYM | 0.326 | 0.287 | 0.294 | 0.325 |
| LDA-ASYM | 0.653 | 0.692 | 0.618 | 0.661 |

TABLE XV. Average *G-mean*. Different models for training and test sets. Medium negative dependence

| *Classifier* | *Normal* | *Clayton* | *Frank* |
|---|---|---|---|
| LINREG4 | 0.499 | 0.461 | 0.473 |
| LINREG14 | 0.551 | 0.583 | 0.554 |
| LOGREG4 | 0.557 | 0.506 | 0.522 |
| LOGREG14 | 0.549 | 0.587 | 0.558 |
| LDA-SYM | 0.572 | 0.479 | 0.581 |
| LDA-ASYM | 0.591 | 0.564 | 0.602 |

are lower in comparison to the case when training and test data are described by the same probability distributions. The relative changes of their values are presented in Tables XVI–XVII for *Accuracy* and *F1 score*, respectively.

TABLE XVI. Relative change of *Accuracy* due to different models for training and test sets

| *Classifier* | *Normal* | *Clayton* | *Gumbel* | *Frank* |
|---|---|---|---|---|
| LINREG4 | 0.950 | 0.925 | 0.950 | 0.968 |
| LINREG14 | 0.883 | 0.889 | 0.911 | 0.914 |
| LOGREG4 | 0.962 | 0.953 | 0.957 | 0.969 |
| LOGREG14 | 0.884 | 0.875 | 0.909 | 0.910 |
| LDA-ASYM | 0.923 | 0.918 | 0.924 | 0.943 |

TABLE XVII. Relative change of *F1 score* due to different models for training and test sets

| *Classifier* | *Normal* | *Clayton* | *Gumbel* | *Frank* |
|---|---|---|---|---|
| LINREG4 | 0.887 | 0.785 | 0.974 | 0.925 |
| LINREG14 | 0.904 | 0.834 | 0.850 | 0.890 |
| LOGREG4 | 0.897 | 0.872 | 0.875 | 0.886 |
| LOGREG14 | 0.904 | 0.837 | 0.853 | 0.884 |
| LDA-ASYM | 0.855 | 0.876 | 0.848 | 0.862 |

In the case of the *G-mean* the results are presented in Table XVIII. In this table we have deleted the LDA-SYM classifier, as its behaviour looks quite random, and thus cannot be compared with the other considered cases.

TABLE XVIII. Relative change of *G-mean* due to different models for training and test sets. Strong positive dependence

| *Classifier* | *Normal* | *Clayton* | *Gumbel* | *Frank* |
|---|---|---|---|---|
| LINREG4 | 0.829 | 0.777 | 0.833 | 0.803 |
| LINREG14 | 0.827 | 0.820 | 0.852 | 0.850 |
| LOGREG4 | 0.906 | 0.845 | 0.906 | 0.884 |
| LOGREG14 | 0.828 | 0.818 | 0.851 | 0.848 |
| LDA-ASYM | 0.791 | 0.773 | 0.842 | 0.790 |

The analysis of the robustness of the considered classifiers to an unexpected change of the underlying model of observed data is not simple and unequivocal. For example, in the case of data described by the Clayton copula the loss of efficiency seems to be the biggest one, so the type of data that yields the best results of classification when training and test data are generated by the same model becomes the worst one when test data are generated from a different model. When we want to compare the quality of considered classifiers, the simple logistic regression classifier (LOGREG4) still seems to be the best. However, its loss of efficiency is the best one only in the case of G-mean. There is also another interesting observation: the robustness of classifiers built on extended models (i.e., that take into account interactions) is generally worse than the robustness of simple models.

Finally, let us consider the problem how the strength of dependence influences the robustness of classifiers to an unexpected change of the underlying model of observed data. We will illustrate this problem on the example of the LOGREG4 classifier, which seems to be the best from among all classifiers considered in this paper. It seems to be quite obvious that there exists a general rule that "the stronger dependence (positive or negative) the better classification". However, the relationship between the strength and type of dependence and the quality of classification may be not so simple. In Table XIX, we show

how the values of the *F1 score* are changing for different copulas and different strengths of dependence.

TABLE XIX. AVERAGE *F1 score* FOR THE LOGREG4 CLASSIFIER. THE SAME MODEL FOR TRAINING AND TEST SETS. DIFFERENT LEVELS OF THE STRENGTH OF DEPENDENCE

| *Dependence* | *Normal* | *Clayton* | *Gumbel* | *Frank* | *FGM* |
|---|---|---|---|---|---|
| Sp | 0.799 | 0.866 | 0.813 | 0.813 | X |
| Mp | 0.537 | 0.662 | 0.463 | 0.509 | X |
| Wp | 0.041 | 0.062 | 0.044 | 0.052 | 0.052 |
| Wn | 0.088 | 0.060 | X | 0.056 | 0.056 |
| Mn | 0.424 | 0.333 | X | 0.379 | X |
| Sn | 0.763 | 0.647 | X | 0.730 | X |

The results displayed in Table XIX reflect the complexity of the stated problem. First of all, the quality of classification strongly depends upon the type of dependence described by a respective copula. Only in the case of the normal (Gaussian) copula (the classical multivariate normal distribution is a particular case of a distribution described by this copula) the relationship between the strength of dependence and the quality of classification (measured by the *F1 score*) is symmetric. For the remaining copulas this relationship is visibly asymmetric (negative dependence leads to worse classification), and the values of the *F1 score* may be quite different despite the same strength of dependence.

When the data in the test sets are generated by different models than in the training sets the values of the *F1 score* are changing. This is illustrated in Table XX for the case of the LOGREG4 classifier.

TABLE XX. AVERAGE *F1 score* FOR THE LOGREG4 CLASSIFIER. DIFFERENT MODELS FOR TRAINING AND TEST SETS. DIFFERENT LEVELS OF THE STRENGTH OF DEPENDENCE

| *Dependence* | *Normal* | *Clayton* | *Gumbel* | *Frank* | *FGM* |
|---|---|---|---|---|---|
| Sp | 0.642 | 0.618 | 0.633 | 0.625 | X |
| Mp | 0.482 | 0.577 | 0.406 | 0.451 | X |
| Wp | 0.069 | 0.089 | 0.064 | 0.077 | 0.076 |
| Wn | 0.110 | 0.082 | X | 0.074 | 0.075 |
| Mn | 0.399 | 0.320 | X | 0.358 | X |
| Sn | 0.648 | 0.558 | X | 0.633 | X |

For strong and medium positive dependencies the strongest worsening of quality of classification has been observed when data are described by the Clayton copula. However, when dependencies are negative, the case of the Normal copula seems to be the worse. It is also surprising that for weak dependencies the values of the *F1 score* have even improved. It shows that in such cases this quality index is rather inappropriate as the results of classification to great extent seem to be random.

## V. CONCLUSIONS

In the paper we have evaluated six binary regression type classifiers. For the comparison we used three measures of quality: the *Accuracy* (i.e., the probability of correct classification), the *F1 score*, which is the harmonic average of *Precision* (equal to one minus the probability of type I error) and *Sensitivity* (equal to one minus the probability of type II error), and the *G-mean*, which is the geometric average of *Sensitivity* and *Specificity*. The evaluation was performed using a complex simulation software that allowed to model strongly nonlinear dependencies of different types (described by different copulas) and different strength (measured by Kendall's $\tau$). The distinctive feature of this research is taking into consideration the impact of the type and the strength of dependence between all variables in the considered model. Moreover, we have considered a practical problem when objects classified by a certain classifier are described by a different probability distribution than the objects used for building (training) this classifier.

The performed experiments revealed that the quality of classification is strongly related to the type of dependence (type of the respective copula). This relationship may have different impact on the performance of different classifiers. For example, a simple linear regression classifier is quite robust to the change of the data model when the data are described by the Gumbel copula, but not robust when the data are described by the Clayton copula, even if the strength of dependence is in both cases the same. What is more important, and from a practical point of view quite undesirable, that the impact of the type of a specific copula strongly depends upon the sign of dependence. For example, when the data are generated by the Clayton copula the performance of considered classifiers is the best in the case of (strong) positive dependence, but the worse for the negative one.

The performed experiments do not reveal unquestionable superiority of anyone of the considered classifiers. It is hardly unexpected, as according to the famous Wolperts "no free lunch theorem" such the best classifier cannot exist. However, classifiers based on linear and logistic regressions are generally (with some exceptions) better than those based on Fisher's linear discrimination. If we take into account both the quality of classification and the robustness to the change of the underlying model, *the classifier based on a simple (without interactions) logistic regression is the best one*. This could serve as the general recommendation for practitioners. However, when some additional information is available, other classifiers could be preferred. For example, if we know that input attributes are dependent, and their dependence is described by the Frank copula, then the LDA classifier with an asymmetric decision criterion would be preferred. In practice, however, obtaining such specific information seems to be rather unlikely, so our general recommendation seems to be valid for the great majority of practical cases.

In the research described in this paper we have analysed the case with when considered classes are imbalanced. We have assumed that the class of main interest contains the minority of observations. This is a situation frequently met in practice, e.g., in medicine or in quality control. Moreover, we have analysed only the case of binary classification. The consideration of more realistic, in some applications, cases of multiple classes requires further research. Moreover, the behaviour of more complex classifiers (e.g., based on decision trees) also requires further investigation. Preliminary investigations (see, e.g., [8]) show, however, that these more complex non-linear classifiers may perform much better in the case of the same data model for training and test data (it is a usual case considered in the data mining community), but loose its superiority when the data models for training and test data sets are different, as in the case considered in this paper.

## REFERENCES

[1] O. Hryniewicz, "On the robustness of regression type classifiers," in *Proceedings of INTELLI'2015*, October 11–16, St. Julians, Malta. IARIA, 2015, pp. 16–22.

[2] R. Duda, P. Hall, and D. Stork, *Pattern Classification, 2nd Edition*. Wiley, 2000.

[3] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning. Data Mining, Inference and Prediction, 2nd Edition*. Springer, 2009.

[4] I. Witten, E. Frank, and F. Hall, *Data Mining. Practical Machine LearningTools and Techniques, 3rd Edition*. Morgan Kaufman, 2011.

[5] N. Japkowicz and M. Shah, *Evaluating Learning Algorithms. A Classification Perspective*. New York: Cambridge University Press, 2015.

[6] D. Hand, “Classifier technology and the illusion of progress,” *Statistical Science*, vol. 21, pp. 1–14, 2006.

[7] O. Hryniewicz, “Spc of processes with predicted data: Application of the data mining methodology,” in *Frontiers in Statistical Quality Control 11*, S. Knoth and W. Schmid, Eds., 2015, pp. 219–235.

[8] ——, “Process inspection by attributes using predicted data,” in *Challenges in Computational Statistics*, ser. Studies in Computational Intelligence 605, S. Matwin and J. Mielniczuk, Eds. Springer Int. Publ. Switzerland, 2016, pp. 113–134.

[9] P. Embrechts, F. Lindskog, and A. McNeil, *Modelling Dependence with Copulas and Applications to Risk Management*. Amsterdam: Elsevier, 2015, ch. 8, pp. 329–384.

[10] R. Nelsen, *An Introduction To Copulas*. New York: Springer, 2006.

[11] O. Hryniewicz and J. Karpiński, “Prediction of reliability - pitfalls of using pearsons correlation,” *Eksploatacja i Niezawodnosc - Maintenance and Reliability*, vol. 16, pp. 472–483, 2014.