Journal of KONBiN 2021
Volume 51, Issue 1
DOI 10.2478/jok-2021-0004

Dariusz AMPUŁA
*Military Institute of Armament Technology (Wojskowy Instytut Techniczny Uzbrojenia)*

# BOOSTING TREES IN APPLICATION TO HAND GRENADES FUZES

## Wzmacniane drzewa w zastosowaniu do zapalników do granatów ręcznych

***Abstract:*** *The article in the introduction presents a brief description of the decision tree, and the purpose of the article was defined. Then, the process of building boosting trees was characterized, paying attention to the algorithm of their building. A method of building boosting trees for UZRGM fuzes is described. The assortment of fuzes in which this type of fuze is used is indicated, and the individual features of the fuze are presented, which are checked during laboratory diagnostic tests. The importance classes that were used to classify the revealed inconsistencies were also described. A boosting classification tree for UZRGM fuzes was designed and built. An exemplary graph of the built tree and its structure and also a fragment of specific values predicted in individual analyzed classes are shown. The matrix of incorrect classifications was determined, which determines the accuracy of the incorrect predictions. On selected examples of the analyzed classes, the designed model was assessed on the basis of the lift chart and gains chart.*
**Keywords:** boosting trees, artificial intelligence, feature, prediction, fuze

***Streszczenie:*** *We wstępie przedstawiono krótki opis drzew decyzyjnych oraz określono cel artykułu. Następnie, scharakteryzowano proces budowy wzmacnianych drzew, zwracając uwagę na algorytm ich budowy. Opisano metodę budowy drzew wzmacnianych dla zapalników typu UZRGM. Wskazano asortyment, w którym stosowane są tego typu zapalniki, oraz przedstawiono poszczególne cechy zapalnika, które sprawdzane są podczas prowadzonych laboratoryjnych badań diagnostycznych. Opisano również klasy ważności, które zostały zastosowane do klasyfikacji ujawnianych niezgodności. Zaprojektowano oraz zbudowano wzmacniane drzewo klasyfikacyjne dla zapalników typu UZRGM. Pokazano przykładowy wykres zbudowanego drzewa oraz jego strukturę, a także fragment określonych wartości przewidywanych w poszczególnych analizowanych klasach. Określono macierz błędnych klasyfikacji, która określa trafność błędnych predykcji. Na wybranych przykładach analizowanych klas, oceniono zaprojektowany model na podstawie wykresu przyrostu i wykresu zysku.*
**Słowa kluczowe:** wzmacniane drzewa, sztuczna inteligencja, cecha, predykcja, zapalnik

# 1. Introduction

A decision tree [4] is a popular and effective algorithm, used especially in classification tasks, but it is also well suited for predicting quantitative phenomena. The attractiveness methods based on decision trees is, inter alia, due to the fact that they represent a set of rules.

Although decision tree is a very effective method, unfortunately, like most algorithms, it is not without defects. One of the main problems that we may encounter when basing our prediction on a decision tree may be the instability of the results. This means that the model of the designed and built decision tree is sensitive to the specifics of the data set and even its slight modification may lead to building a completely different tree. Hence, the idea to improve this classic data mining model. One way to solve the problem of instability of the decision tree model is to boost decision trees.

In the case of some difficult estimation and prediction tasks [3], the predictions generated by sequences of relatively simple trees are closer to real values than the predictions of, e.g. neural networks or one complex tree. The technique of using a sequence of simple models, with each subsequent model paying more attention to those observations that have been misclassified by the previous models, is called boosting.

Boosting trees [3] inherit to some extent, some of the benefits of "ordinary" classification trees. Namely, they naturally support different types of predictors (quantitative and qualitative) and are insensitive to the units in which the variables are expressed.

Boosting produces very effective models - these are one of the strongest predictive data mining methods. It is worth noting that the role of a simple model is very well fulfilled by "small" classification or regression trees. Boosting trees are suggested as the best general method for solving data mining tasks.

The aim of this article was to show the applicability of the model of boosting decision trees, and more specifically of boosting classification trees, to the obtained test results of elements of UZRGM hand grenade fuzes. These fuzes have the largest number of test results, thanks to which the designed model of boosting classification trees will be more reliable. In the case of a smaller number of test results, the development of this type of models may prove impossible; then it remains possible to build models based on single decision trees.

# 2. The process of building boosting trees

Boosting trees are a group of many, very simple trees (usually with one or two divisions), which are learned based on data, in which we attach more and more importance to the so-called "difficult" cases (i.e. those for which there has been prediction error).

The software [8] builds a model of boosting trees using the stochastic gradient method and creating a series of straight trees (the complexity of each tree can be determined). The trees designed in the built model of boosting trees can be viewed as a graph or as a structure sheet after entering the appropriate numbers of a given tree.

Boosting [5] of trees generally involves repeating the learning process, with more and more weight being given to those cases that previously caused trouble. More precisely, we start learning by developing a simple model. At each step of building an algorithm, a subset is drawn from the learning sample, for which the tree is matched. This reduces the risk of over-learning, i.e. the model will learn the data by heart. The weights can be determined, for example, by increasing the initial weights using the error value of the entire model. The weights for correctly classified cases do not change.

In the next step, we fit a simple model for the data with corrected weights. We combine the partial models obtained in this way into one model by calculating their weighted sum.

It should be noted that in the classification problem, the components of the tree model is not the value of the dependent variable, but the value of residuals, i.e. the deviations between the indicator variable and the calculated probability.

After entering all the necessary parameters when designing the model, the software automatically selects the best model based on the error for the test sample; you can view the various characteristics of the built model and generate the code necessary to apply this model to the new test results obtained from UZRGM fuzes. Thanks to the "quick implementation" module, it is possible to implement a previously designed and save model from evaluating new test results of elements of hand grenade fuzes.

# 3. A method of building boosting trees

When designing the boosting classification trees [3], the default division of data into a learning and a testing sample, the so-called “set aside” sample, was adopted. If you do not specify a variable with codes for the testing sample, the software automatically selects a random sample of 30% of the observations (cases) from the data and treats them as a testing sample to evaluate the fit of the model in subsequent iterations. The remaining 70% of the observations serve as data for the analysis by the stochastic gradient amplification method (e.g. for the samples selected in the subsequent amplification steps). By default, the software selects a specific solution (a specific number of simple boosting trees) so that the absolute error (rate of incorrect classifications) in all iterations of amplifications is as low as possible.

In boosting classification trees, the software builds independent sets of these trees for each qualitative category of the dependent variable. In our case, we have six different values of this variable, i.e. we will have six independent sets of boosting classification trees.

A logistic transformation is used in the software to calculate the predicted values. It is worth noting that the algorithm does not modify the prediction probabilities for each class and each tree in the case of different a priori probabilities or different costs of incorrect classifications for different classes. The predicted classification probabilities for each class

are modified if a priori probabilities or costs of incorrect classifications are not equal to each other.

The boosting trees module in the software calculates a weighted "additive" expansion of simple trees. The weight with which successive trees are added to the prediction equation is usually constant and is called the learning rate or shrinkage parameter. The best designed model is obtained when the value of this parameter is 0.1 or less (then the prognostic accuracy is better).

The implementation of the stochastic gradient amplification algorithm in the software [8] can account for missing data in the forecasts. If, while building the model, the algorithm encounters no data, then a prediction is made for such a case based on the last node of the tree. The prediction is the average of the node values. This way, there is no need to remove missing predictors from the analysis. The prepared database of test results of elements of UZRGM fuzes for hand grenade is complete and has no missing data.

UZRGM fuzes are used in F-1, RG-42, RGO-88, RGZ-89 and CGR-42A hand grenades. In the prepared database of the test results of these fuzes, the results of the so-called scientific and test inquiries that are not reliable for other test results were eliminated. The test results for other ministries were not analyzed either, only studies for which the type of test specified in the test methodology was equal to one that was taken. All the constraints were aimed at creating a homogeneous set of data results.

The UZRGM fuze is a time fuze [1] characterized by a certain delay in working, which is the basic parameter that is checked during diagnostic tests. Other tested features are the parameters of the used spring, the action of a primer cap and detonating cap, the action of the fire chain, the corrosion state of individual parts and assemblies of the fuze, fuze safety condition, as well as the correct operation of the fuze pin.

All tested features of fuzes were divided according to test methodology [6] into five classes of importance: A, B, C, D and E. Depending on the inconsistencies detected during laboratory tests, the tested lot of fuzes receives a specific post-diagnostic decision.

In the designed model of boosting trees for UZRGM fuzes, in the first laboratory diagnostic tests, according to test methodology, we accepted eight predictors, which were: number inconsistencies in the importance class A (LA), number inconsistencies in the importance class B (LB), number of inconsistent fuzes in importance class B (NB), the total number of inconsistent fuzes (N), number of inconsistent fuzes in importance class C (NC), number inconsistencies in the importance class C (LC), number inconsistencies in the importance class D (LD), number inconsistencies in the importance class E (LE).

# 4. Results of building boosting classification trees

When designing our model of boosting classification trees, it was assumed during building that this is a classification task, and a dependent variable was determined, which in our case is the value of "DEC", i.e. undertaken the post-diagnostic decision. Quantitative

predictors were also determined, which were the obtained test results of the particular tested features of the UZRGM fuze.

The costs of incorrect classification were assumed to be "equal" and the a priori probabilities as "estimated". From list [3], we select the method of determining the likelihood of classifying an object into one of the categories, without knowing the value of the predictors in the model. In our case, the "estimated" position was selected, then the probability of hitting a specific class is proportional to the multiplicity of that class.

In order to find the best model to build, the value of the maximum number of nodes was changed from 7 to 3. This parameter specifies the maximum number of nodes allowed for the trees making up the model.

The minimum size of the node of component trees, which is divided (i.e. how many descendants it may have), was also determined. The maximum number of levels has been entered, specifying the maximum number of levels allowed for the trees making up our model. The software specified minimum descendants cardinality, which specifies the minimum number of objects in the node emerging after splitting, has been accepted. This criterion determines whether to look for splits for the node at all, and this setting allows only splits that result in nodes having at least the given number of objects.

A random testing sample proportion of 0.3 was introduced, which tells how much of the data set will be randomly selected for the testing sample. The proportion for the sub-samples was also specified as 0.5. This value determines what part of the data set will constitute the learning test at the subsequent stages of amplification of the model.

The learning coefficient, i.e. the value of a specific weight with which the next tree is added to the development, was taken as 0.1 as indicated by the software, then the model will have a good predictive ability.

The number of trees from which the model was built was set to 100 to minimize the error in the testing sample and our model was enlarged by another 100 trees, reaching the value of 800 trees. Further enlargement of this model did not introduce any changes to the model.

After analyzing the obtained models, a model was selected in which the error in the testing sample stabilized at a constant level. The model with the number of trees equal to 200 was selected, in which the mean polynomial deviation for the learning and testing data is presented in fig. 1. A slight change in the value of these deviations in the range 115 of the tree does not affect the overall course of these curves. In our model, we obtained the optimal number of trees amounting to 109, which means that with this number of trees, the building model begins to show an excessive fit to the analyzed data, and it is also the point where we get the smallest error for the test data.

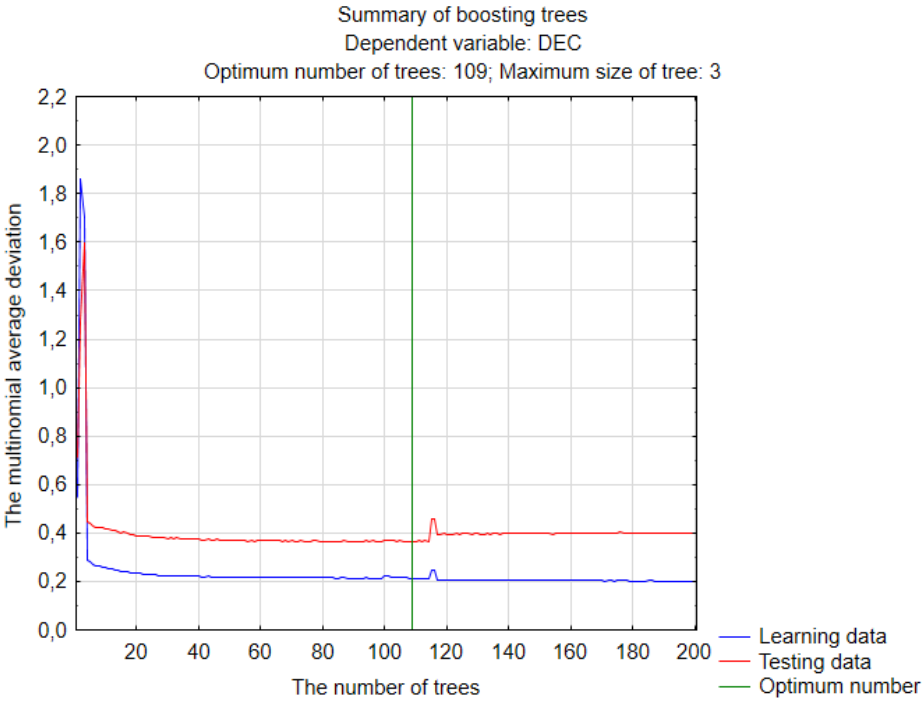


**Fig. 1.** Summary of boosting trees

Boosting trees are built using the stochastic gradient method, creating a series of simple trees from which you can determine the complexity of each of them. Individual trees in the built model can be viewed as a graph or a structure sheet after entering the appropriate numbers of these trees. Graphs of this type and structures are generated for each class of the dependent variable. An exemplary graph and structure sheet for tree number 1, for class "B3" are shown in figs. 2 and 3. The nodes are described by the correction of the mean *(Mu)* and the variance *(Var)*. The node number *(ID)*, the multiplicity of the analyzed cases *(N)* and the division criterion are also given, in our case it is the variable NB, which causes the division with a constant value of 0.5.

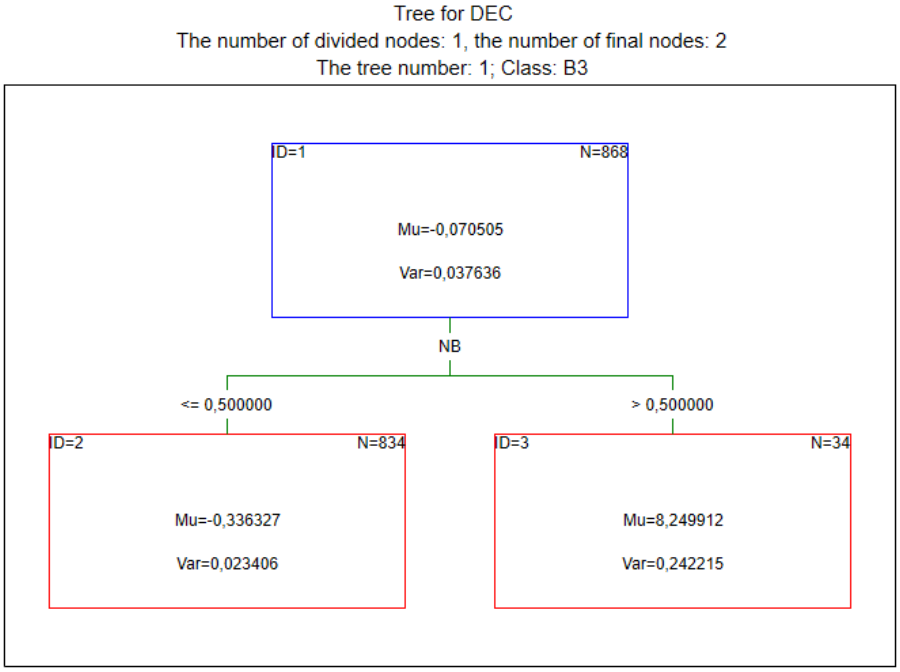


**Fig. 2.** The graph of tree No. 1 for "B3" class

The structure of the tree (UZRGM)
Dependent variable: DEC
The tree number: 1; Class: B3

| | Descendant node 1 | Descendant node 2 | The node size | The node mean | The node Var | Divide variable | Divide constant |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 868 | -0,070505 | 0,037636 | NB | 0,500000 |
| 2 | | | 834 | -0,336327 | 0,023406 | | |
| 3 | | | 34 | 8,249912 | 0,242215 | | |

**Fig. 3.** The structure of tree No. 1 for "B3" class

Figure 4 shows the beginning of the resulting table of predictions for all observations, which determines the predicted classification for all observations for the analyzed samples. The fragment of the table shows that the observation marked in red, according to our model, should have a different value (Z). Therefore, the card with the results of this lot and the post-diagnostic decision should be carefully analyzed once again.

The matrix of incorrect classifications of our adopted model is shown in fig. 5. This matrix shows that for example the total error fraction for the "B5" class was 3.26% and it is the lowest of all other models in this class. The remaining error fractions for other classes also show the lowest values for the model built in this way. The use of a cross-validation in the analysis protects against over-fitting and allows for finding the estimates of the model parameters.

After building our model, we can define (fig. 6) a table of relative importance of each of the predictors in the final model (series of trees). As you can see, the "N" predictor is the most important, while the least important are the "LD" and "LA" predictors whose importance is zero in the model.

| | Prediction (UZRGM)<br>Dependent variable: DEC<br>The cross test; The number of trees: 109 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Observed value** | Predicted value | Probability B5 | Probability PS | Probability BP | Probability Z | Probability B3 | Probability W |
| **1** | B5 | B5 | 0,625666 | 0,022676 | 0,007405 | 0,042706 | 0,300933 | 0,000614 |
| 4 | BP | BP | 0,000840 | 0,006076 | 0,952723 | 0,011443 | 0,028753 | 0,000164 |
| 5 | B5 | B5 | 0,625666 | 0,022676 | 0,007405 | 0,042706 | 0,300933 | 0,000614 |
| 8 | BP | Z | 0,000037 | 0,000435 | 0,070265 | 0,921623 | 0,007628 | 0,000012 |
| 14 | B5 | B5 | 0,625666 | 0,022676 | 0,007405 | 0,042706 | 0,300933 | 0,000614 |
| 23 | B5 | B5 | 0,625666 | 0,022676 | 0,007405 | 0,042706 | 0,300933 | 0,000614 |
| 26 | B5 | B5 | 0,625666 | 0,022676 | 0,007405 | 0,042706 | 0,300933 | 0,000614 |
| 27 | B5 | B5 | 0,625666 | 0,022676 | 0,007405 | 0,042706 | 0,300933 | 0,000614 |
| 29 | BP | BP | 0,000840 | 0,006076 | 0,952723 | 0,011443 | 0,028753 | 0,000164 |
| 32 | B5 | B5 | 0,625666 | 0,022676 | 0,007405 | 0,042706 | 0,300933 | 0,000614 |
| 36 | B5 | B5 | 0,625666 | 0,022676 | 0,007405 | 0,042706 | 0,300933 | 0,000614 |
| 45 | B5 | B5 | 0,625666 | 0,022676 | 0,007405 | 0,042706 | 0,300933 | 0,000614 |
| 46 | BP | BP | 0,000840 | 0,006076 | 0,952723 | 0,011443 | 0,028753 | 0,000164 |
| 48 | BP | BP | 0,000840 | 0,006076 | 0,952723 | 0,011443 | 0,028753 | 0,000164 |
| 49 | B5 | B5 | 0,625666 | 0,022676 | 0,007405 | 0,042706 | 0,300933 | 0,000614 |
| 54 | B5 | B5 | 0,625666 | 0,022676 | 0,007405 | 0,042706 | 0,300933 | 0,000614 |
| 56 | B5 | B5 | 0,625666 | 0,022676 | 0,007405 | 0,042706 | 0,300933 | 0,000614 |
| 64 | B5 | B5 | 0,625666 | 0,022676 | 0,007405 | 0,042706 | 0,300933 | 0,000614 |
| 71 | BP | BP | 0,000840 | 0,006076 | 0,952723 | 0,011443 | 0,028753 | 0,000164 |
| 77 | B5 | B5 | 0,625666 | 0,022676 | 0,007405 | 0,042706 | 0,300933 | 0,000614 |

**Fig. 4.** Predicted values in individual classes

| | Matrix of classification (UZRGM)<br>Dependent variable: DEC<br>The cross test; The number of trees: 109 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Observed** | Predicted B5 | Predicted PS | Predicted BP | Predicted Z | Predicted B3 | Predicted W | Together in the line |
| **Number** | B5 | 579 | 1 | | 2 | | 1 | 583 |
| % from column | | 96.34% | 3.13% | 0.00% | 4.35% | 0.00% | 3.70% | |
| % from line | | 99.31% | 0.17% | 0.00% | 0.34% | 0.00% | 0.17% | |
| % from total | | 72.47% | 0.13% | 0.00% | 0.25% | 0.00% | 0.13% | 72.97% |
| Number | PS | | 11 | | | | | 11 |
| % from column | | 0.00% | 34.38% | 0.00% | 0.00% | 0.00% | 0.00% | |
| % from line | | 0.00% | 100.00% | 0.00% | 0.00% | 0.00% | 0.00% | |
| % from total | | 0.00% | 1.38% | 0.00% | 0.00% | 0.00% | 0.00% | 1.38% |
| Number | BP | | | 66 | 3 | 2 | | 71 |
| % from column | | 0.00% | 0.00% | 92.96% | 6.52% | 9.09% | 0.00% | |
| % from line | | 0.00% | 0.00% | 92.96% | 4.23% | 2.82% | 0.00% | |
| % from total | | 0.00% | 0.00% | 8.26% | 0.38% | 0.25% | 0.00% | 8.89% |
| Number | Z | 2 | 1 | 2 | 38 | 3 | | 46 |
| % from column | | 0.33% | 3.13% | 2.82% | 82.61% | 13.64% | 0.00% | |
| % from line | | 4.35% | 2.17% | 4.35% | 82.61% | 6.52% | 0.00% | |
| % from total | | 0.25% | 0.13% | 0.25% | 4.76% | 0.38% | 0.00% | 5.76% |
| Number | B3 | 20 | | 3 | 2 | 17 | | 42 |
| % from column | | 3.33% | 0.00% | 4.23% | 4.35% | 77.27% | 0.00% | |
| % from line | | 47.62% | 0.00% | 7.14% | 4.76% | 40.48% | 0.00% | |
| % from total | | 2.50% | 0.00% | 0.38% | 0.25% | 2.13% | 0.00% | 5.26% |
| Number | W | | 19 | | 1 | | 26 | 46 |
| % from column | | 0.00% | 59.38% | 0.00% | 2.17% | 0.00% | 96.30% | |
| % from line | | 0.00% | 41.30% | 0.00% | 2.17% | 0.00% | 56.52% | |
| % from total | | 0.00% | 2.38% | 0.00% | 0.13% | 0.00% | 3.25% | 5.76% |
| Number | Total groups | 601 | 32 | 71 | 46 | 22 | 27 | 799 |
| % together | | 75.22% | 4.01% | 8.89% | 5.76% | 2.75% | 3.38% | |

**Fig. 5.** The matrix of incorrect classifications

| | Importance of predictors (UZRGM)<br>Dependent variable: DEC | |
|---|---|---|
| | **Variable rank** | Importance |
| **N** | 100 | 1,000000 |
| NC | 95 | 0,952837 |
| LC | 95 | 0,952322 |
| LE | 78 | 0,784040 |
| NB | 37 | 0,373954 |
| LB | 37 | 0,372772 |
| LD | 0 | 0,000000 |
| LA | 0 | 0,000000 |

**Fig. 6.** The table of importance predictors

As each tree is built, for each division, prediction statistics are computed for each predictor, and the best predictor (i.e. the one giving the best division in the node under consideration) is used for the division. The software calculates the mean statistic for all variables and all trees in the strengthening sequence. Final predictor importance is computed by normalizing these means (highest mean is one and all others are expressed as the mean statistic for the predictor relative to the highest value).

You can also define a sheet with risk evaluation for the learning sample and the testing sample. Such a sheet for our model is presented in fig. 7. The risk, i.e. the fraction of cases incorrectly classified by the tree, is expressed in terms of the total cost. As can be seen from the sheet, low values of the risk evaluation and the standard error that was committed qualify the built model to be used in practice.

| | Risk evaluation (UZRGM) Dependent variable: DEC | |
|---|---|---|
| | **Risk Evaluation** | Standard error |
| **Learning** | 0,058758 | 0,005537 |
| Testing | 0,077597 | 0,009465 |

**Fig. 7.** The sheet of risk evaluation

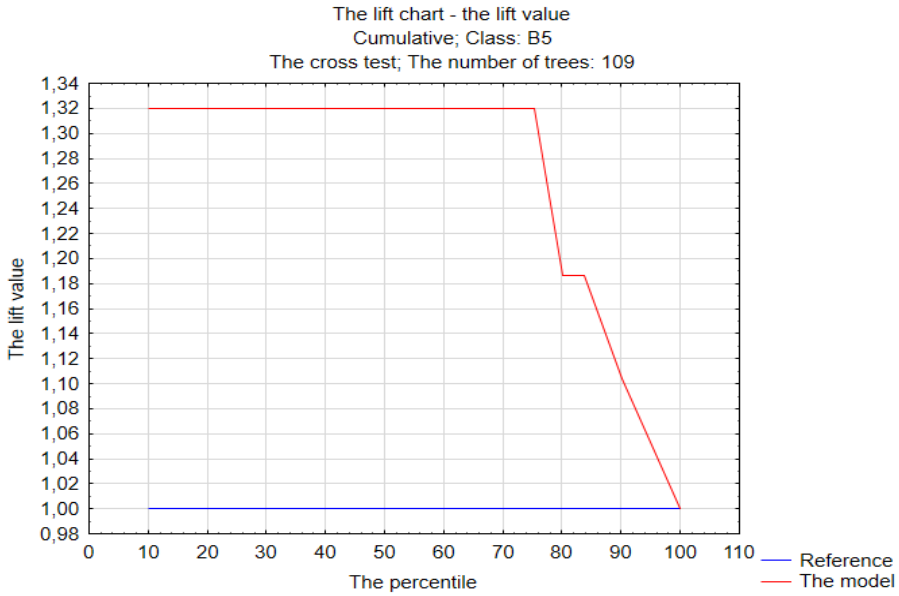


**Fig. 8.** The lift chart for "B5" class

To evaluate the quality of the built predictive model, a lift chart should be drawn up, which is a graphical summary of the usefulness of the model for predicting the value of the dependent variable. An example of a graph value of the increment for class "B5" is shown in fig. 8.

The graphs of this type were created for all classes of the dependent variable. On the (y) axis you can see the increase values, i.e. multiples with respect to the reference line, and on the (x) axis the percentile values were calculated, i.e. values below which the values of the given percentage of samples fall. Thus, taking, for example, 10% of the cases most

likely classified as class "B5" (with the highest probabilities of classification), we will get a sample that contains about 1.32 times more cases than if the selection was random. The resulting graph is created for changing sets that contain an increasing number of cases with the highest probability of hitting a given class, resulting from the created model, so that the next one contains the preceding one (in this sense the graph is cumulative).

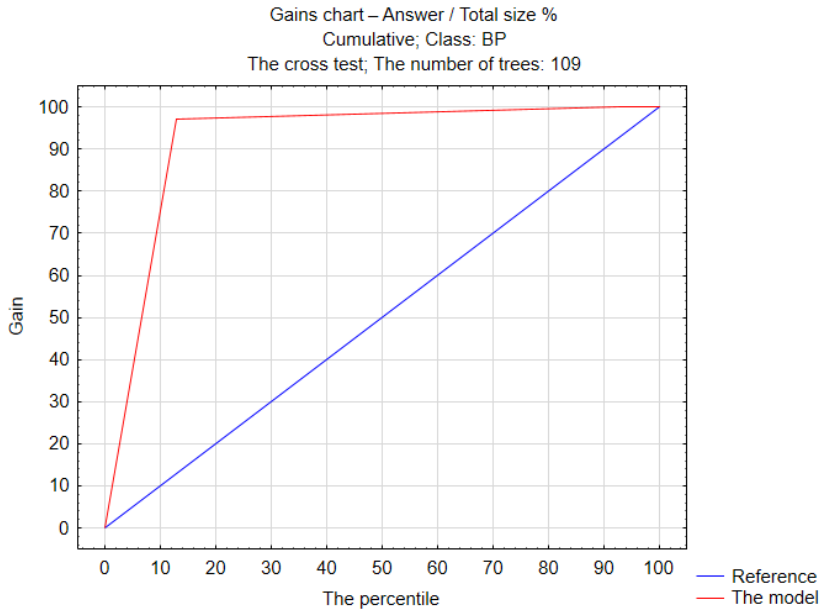


**Fig. 9.** Gains chart for “BP” class

You can also use gains chart in each class of the dependent variable to evaluate the quality of the prediction. It shows the percentage of observations correctly classified to the selected class. A straight line is always marked on this chart, which is the reference line for comparing the relevant models in the classification. By comparing this line with the curve obtained for our model, we are able to assess the quality and usefulness of the built model. An example of a gains chart for the "BP" class is shown in fig. 9. The chart shows that if we take 13% of cases with the highest classification probabilities for this class, they will account for about 97% of all correctly classified cases in this class. Gains charts were also made for all classes in our model.

In the designed model of boosting classification trees, for the further use of this built model, code was generated that will be used by the "quick implementation" module of predictive models, which effectively calculates the predicted values.

Thanks to this code, in our case, the PMML (Predictive Models Markup Language) code was used. This model can be used for new results of the tested lots of UZRGM fuzes. It is also possible to generate other types of codes, e.g. C/C++, SVB - Statistica Visual Basic, Java, SAS or the SQL code of a given user.

# 5. Summary

Boosting trees is a powerful machine learning tool for predictive data mining. This technique can approximate virtually all types of non-linear relationships between predictors and independent variables, and often provides much faster and simpler solutions than, for example, the various neural network architectures available in software [8].

There is a general tendency to increase the degree of automation of manufacturing processes and research processes. When performing these processes, a huge amount of data is obtained, at every production or research stage of a given process, and which data can be used for existing artificial intelligence tools. One of these tools is the boosting decision trees, the idea of which was developed in the late 20th century.

The article shows the possibility of designing and building boosting classification trees for the existing laboratory diagnostic test results of elements of UZRGM hand grenade fuzes. The aim set at the beginning of the article has been fully achieved.

Therefore, a model of boosting classification trees was designed according to the technical requirements of the software. Then, based on the adopted detailed parameters, a model of boosting classification trees was built, which can be successfully used for new tested lots of UZRGM fuzes. The predictive module used in the software allows us to accurately determine the post-diagnostic decision for the new empirical data obtained.

The elimination of the human factor from the evaluation process of the tested elements of hand grenade fuzes increases the credibility of the evaluation mechanism. On the other hand, the date of implementing such a designed and built model of boosting classification trees depends on the management of the research unit in which diagnostic laboratory tests are carried out.

The boosting classification trees module can also be applied to other tested elements of ammunition. However, there is one condition, namely the database containing the results of previous studies, must contain at least a significant number of observations of the examined features, otherwise the existing relationships between the analyzed predictors may be considered by the software as too weak.

Boosting classification trees is thus another data mining tool that can be successfully used to predict the dependent variable for new lots of UZRGM hand grenade fuzes tested. The model presented in the article, thanks to the Statistica software, confirms this possibility, while its implementation depends only on the management of the research facility in which the laboratory diagnostic tests are performed.

# 6. References

1. Amunicja wojsk lądowych. Ministry of National Defence Publishing House, Warszawa 1985.

2. Cards from laboratory tests of fuzes type UZRGM – archive Military Institute of Armament Technology (MIAT).
3. Electronic handbook "Statistica" – Statsoft Poland 2018.
4. Grabowska E.: Jak udoskonalić algorytm drzew decyzyjnych?, https://predictivesolutions.pl / jak – udoskonalić – algorytm – drzew – decyzyjnych.
5. Łapczyński M., Demski T.: Data mining – predictive methods. Statsoft Poland, materials from course, Cracow 2019.
6. Metodyka badań diagnostycznych amunicji – Indeks N-5001b – archive MIAT, 1985.
7. Reports from tests of ammunition – archive MIAT.
8. Statistica 13.3 PL – computer software, Statsoft Poland 2018.