Applied Soft Computing 141 (2023) 110285

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

BrightBox — A rough set based technology for diagnosing mistakes of machine learning models

Andrzej Janusz[a,b,*], Andżelika Zalewska[a], Łukasz Wawrowski[a,d], Piotr Biczyk[a,c], Jan Ludziejewski[b], Marek Sikora[c], Dominik Ślęzak[a,b]

[a] QED Software sp. z o.o., Mazowiecka 11/49, 00-052 Warsaw, Poland
[b] Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland
[c] Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland
[d] Research Network Łukasiewicz – Institute of Innovative Technologies EMAG, ul. Leopolda 31, 40–189 Katowice, Poland

ARTICLE INFO

Article history:
Received 29 December 2022
Received in revised form 24 March 2023
Accepted 5 April 2023
Available online 11 April 2023

MSC:
68T99

Keywords:
Machine learning diagnostics
Surrogate models
Model approximation
Rough sets
Ensembles of reducts
Explainable artificial intelligence
BrightBox technology

ABSTRACT

The paper presents a novel approach to investigating mistakes in machine learning model operations. The considered approach is the basis for BrightBox – a diagnostic technology that can be used for analyzing prediction models and identifying model- and data-related issues. The idea is to generate surrogate rough set-based models from data that approximate decisions made by monitored black-box models. Such approximators are used to compute neighborhoods of instances that undergo the diagnostic process — the neighborhoods consist of historical instances that were processed in a similar way by rough set-based models. The diagnostic process is then based on the analysis of mistakes registered in such neighborhoods. The experiments performed on real-world data sets confirm that such analysis can provide us with efficient and valid insights about the reasons for the poor performance of machine learning models.

1. Introduction

Since the beginning of the industrial revolution, attempts are made at increasing the efficiency of industry by implementing automation. With recent advances in machine learning and data science, this expansion is now driven by artificial intelligence-enabled automation. In this context, the correct implementation and maintenance of machine learning models are of paramount importance. One of the aspects of machine learning model maintenance is diagnostics, allowing one to understand model operations, detect malfunctions, and explore the reasons behind the mistakes that models make.

The goal of this paper is to introduce the BrightBox technology that allows for diagnosing machine learning models — investigating types of mistakes and their causes for singular data points (instances) and then providing a framework for analyzing and generalizing such local results into global diagnostics of the model- and data-related issues. This way, we aim to provide machine learning engineers with insight into the actual reasons behind mistakes and enable better-informed decisions regarding the model and data updating process.

BrightBox is based on the exploration of neighborhoods of instances processed by the machine learning model. Neighborhoods are obtained using the diagnosed model approximator — an ensemble of approximate reducts known from the theory of rough sets. The neighborhood of a diagnosed instance is determined by the decision process of the model approximator. Neighbors are defined as instances covered by the same combinations of attribute values on reducts as the diagnosed instance. Given the fact that the model approximator is defined as an ensemble of reducts, we can consider a similarity measure between instances, i.e., check for how many reducts in the ensemble a given pair of instances is processed in the same way. The neighborhood is then analyzed for consistency of labels (ground truth labels, original model predictions, and their approximations), its size, and uncertainty of predictions. These characteristics are then included as diagnostic attributes that constitute the input for diagnostic rules, providing meaningful information on model operations.

It is important to note that this paper is the first one that explains the details of BrightBox technology. Novel contributions in the presented work – and the main novel features of BrightBox – include among others:

[*] Corresponding author at: Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland.
E-mail address: andrzej.janusz@qed.pl (A. Janusz).

https://doi.org/10.1016/j.asoc.2023.110285
1568-4946/© 2023 Elsevier B.V. All rights reserved.

- a general procedure for building a rough set-based approximator of an observed machine learning model, that retains certain quality defined by a convergence coefficient, and is independent of the model type;
- a definition of a neighborhood of an instance that is based on the notion of similarity in a decision-making process, i.e. similarity defined by a percentage of rough set-based reducts in the ensemble, for which pairs of instances have the same combinations of attribute values;
- a definition of diagnostic attributes (not to be confused with attributes in the investigated data sets), including a prediction uncertainty coefficient that is robust to skewed prior distributions of target variable; these attributes are utilized during the analysis of model predictions and committed mistakes, as well as a procedure of using those attributes to identify issues related to machine learning model generalization capabilities, e.g. over-fitting or under-parameterization;
- a thorough experimental evaluation of the proposed technology, including: (1) verification of correctness of the above procedure over a large set of pairs of the form "data set + machine learning model trained for that data set" (whereby the created models were e.g. intentionally over-fitted), and (2) a real-world case study showing the usefulness of BrightBox for companies deploying machine learning models and assessing the quality of rough set based approximators by means of comparison of their attribute importance rankings with analogous rankings for the diagnosed models.

BrightBox does not require knowledge about the diagnosed model nor direct access to it — the diagnosis is performed based solely on the diagnosed data set and model's outputs (predictions, classifications, etc.) for that data. We also assume access to the data set originally used to train the model, although predictions for that set are not needed. These assumptions are commonly fulfilled in industrial environments where machine learning models are deployed, which makes the presented approach even more useful. The above-mentioned experimental evaluation of BrightBox is therefore possible only in a research environment as, in particular, there is no way to compute the attribute importance rankings for the diagnosed black-box models. Oppositely, the rankings produced by our rough set-based surrogate models can be regarded as estimations of the actual rankings for the diagnosed models in practice.

In experiments, we verified the assumption that the proposed characteristics of instances' neighborhoods provide sufficient information to describe potential issues related to the model training process or data quality. In particular, we checked whether our diagnostic attributes allow for distinguishing between correct and erroneous model predictions. It is worth noting that diagnostic attributes are computed post-hoc, and we do not intend to use them to correct model predictions. However, we experimentally checked if they are descriptive enough to identify instances that were problematic for the model. We also verified that diagnostic attribute value distributions allow us to distinguish under-fitted and over-fitted models from the ones whose performance is near-optimal for the given data. This ability is important in many practical scenarios, not only in the ongoing model diagnostics but also e.g. at an earlier stage of choosing machine learning solutions to be deployed in particular applications.

The remaining of this paper is organized as follows: A summary of the related works is outlined in Section 2. Section 3 details the basic notation and methods used in BrightBox. In Section 4, the whole procedure aimed at the model diagnostics is presented. Section 5 presents the results of experiments aimed at the evaluation of the viability and usefulness of the proposed technology. Section 6 presents a case study showing how BrightBox technology works in practice and can be applied to diagnose scoring models from the cybersecurity domain. Finally, some conclusions and future work plans are presented in Section 7.

2. Related work

The work presented in the paper falls into a broad context of research related to testing, interpretation, and diagnosis of machine learning models — bound together into a common field of machine learning quality assurance. In a traditional approach, there are two ways of tackling the issue of model diagnosis. The first of them is the implementation of an analytical loop (e.g. in the CRISP-DM methodology [1]) that identifies the need for the model update if it does not fulfill predefined criteria, without a deeper analysis of reasons behind the model failure. The second way refers to global diagnostics accomplished by performing a comparison of model outputs and their efficiency on different subsets of a data set, followed by visualization of the detected differences [2].

The process of testing and diagnosing machine learning models differs significantly from traditional software testing [3]. A recent survey [4] of the machine learning testing and diagnosing methods shows that current techniques focus on bugs in machine learning frameworks, while paying less attention to, e.g., data quality. At the same time, 65.5% methods included in the survey concentrated on such qualities of machine learning models as their correctness and robustness, while putting less stress on, e.g., model relevance or interpretability of results. Based on the survey it is clear that most of the currently developed methods focus on actual finding of mistakes, and not explaining their causes.

Interpretable diagnostics of machine learning corresponds to the aforementioned domain of XAI which concentrates on techniques aiming at "opening" the black-box models and providing their comprehensive explanations to humans [5]. XAI is used to gain insights about investigated phenomena, diagnose predictive models, and ensure their fairness. Many methods leading to global (data set level) and local (instance level) explanations of machine learning models have been developed [6,7]. Such methods can be classified as attribute-oriented (SHAP, Grad-CAM), global (GAMs), concept (CAVs), and surrogate (LIME) models, as well as local (LRP) and human-centric. The importance of XAI grows with increasing real-world usage of deep neural networks [8], especially in the areas like image and sound processing [9]. A lack of sound explanation constitutes both ethical and practical problems with the widespread adoption of machine learning-based applications [10]. We rely on such studies when using the elements of XAI in our own work although it is worth noting that those elements have not been applied in this context before. In particular, our way of comparing attribute-oriented rankings in order to evaluate empirically the quality of rough set-based surrogate models is a novel idea.

One possible utilization of XAI is to improve the aforementioned analytical feedback loops, where various quality measures of a model being maintained are subject to monitoring. Maintenance actions are being performed based on detected changes in these qualities. Several companies deliver solutions that aim at facilitating such a maintenance process [11,12]. The approaches employed by these solutions provide warnings on several levels such as statistical properties of data (including detection of anomalies and assessment of data quality), the correctness of model operations based on comparison to ground truth, bias drift based on changes in metrics commonly attributed to known biases, as well as attribute drift based on changes in the morphology of attributes values. However, such approaches do not cover explanations of the causes of mistakes.

Table 1
Basic notations and definitions.

Notation	Description
M, $M(x)$	are the diagnosed model and its prediction for an instance x, i.e. $M(x) = \arg\max_i q_i^M(x)$;
$\overline{M}$, $\overline{M}(x)$	are the approximator of model M, and the approximated prediction for x, respectively;
$D = (X_D, d_D)$	refers to the data set used for the diagnostic of the model M; $X_D \subset U$ is the set of instances and d_D are their decision classes (i.e. the ground truth target values);
$R = (X_R, d_R)$	is a reference data set — we assume that this set was used to train the diagnosed model M; it is further used to compute neighborhoods of instances from X_D;
$q^M(x) = (q_1^M(x), \dots q_l^M(x))$	is the estimation of decision class probabilities made by the diagnosed model M; we abbreviate this notation to $q(x)$ and $q_i(x)$ when M is obvious from the context;
$\overline{q^M}(x) = (\overline{q_1^M}(x), \dots \overline{q_l^M}(x))$	is the approximation of decision class probability predictions; we abbreviate this notation to $\overline{q}(x)$ and $\overline{q_i}(x)$ when the model is obvious from the context;
$N(x) \subseteq X_R$	is the neighborhood obtained for $x \in X_D$, computed using the approximator of the diagnosed model;
$p = (p_1, \dots p_l)$	is the prior probability distribution of decision classes; we consider only classes which exist in U, thus each $p_i > 0$;
$Unc^M(x)$	is the prediction uncertainty of M calculated for x; we abbreviate this notation to $Unc(x)$ when M is obvious from the context.

A typical feedback loop-based deployment of our BrightBox technology refers to the continuous analysis of new instances for which a diagnosed model made mistakes, searching for historical data instances which are similar to each such new instance (whereby similarity between instances is derived from an ensemble of rough set based reducts [13] that approximates the diagnosed model), and investigating the collections of such most similar instances (called neighborhoods of the analyzed new instances) from a perspective of analogous mistakes (or their lack). Herein, we do not assume the ability to access those diagnosed black-box models. There are also other – not loop-based – scenarios of machine learning model diagnostics, whereas such assumptions are important to be discussed. One of such scenarios refers to the integration of BrightBox with KnowledgePit — our online platform for organizing open data science competitions [14].[1]

Given no guarantee of access to the diagnosed models, we are basing just on the ability to compare the model outputs with the ground truth. In such a scenario, it is indeed crucial to build surrogate models or in other words — approximators. We use for this purpose methods based on the theory of rough sets [15]. However, the idea of using interpretable approximators in the area of matching learning is not new. In [16], a special decision tree induction algorithm is proposed to approximate the operation of complex (black-box) models. In [17], it is presented how to transform decision forests into interpretable decision trees. In [18], there is reported a COVID-19 detection system that uses a decision tree to explain decisions made by XGBoost and logistic regression models. Rule induction algorithms have also been applied to the problem of surrogate model construction. In [19], there is a technique to induce sets of if-then-else rules that capture the most important relations between attributes to globally explain the predictions of a neural network. Finally, in [20] authors propose a rule-based method for explaining classifier predictions on a specific instance by analyzing the joint effect of attribute subsets on classifier outcomes. This method is appropriate for instance-level explanations.

Our choice of using the theory of rough sets to build surrogate models is due to the fact that this theory has been designed from its very beginning as a tool for dealing with imperfect knowledge, in particular with vague and non-clearly-defined concepts. Its characteristics are that only the facts hidden in data are analyzed, no additional information about data is required, and after the analysis, a kind of compacted knowledge representation is obtained. That representation refers to the notion of approximate reduct — an irreducible subset of the original set of attributes that retains almost the same information about a target variable (called a decision attribute) as the full set of attributes [21]. In applications, machine learning models based on ensembles of such subsets are particularly efficient [22–24]. There are a number of heuristics and randomized algorithms developed to derive ensembles of possibly diverse (i.e. based on diverse attributes) reducts from data [25–27]. In BrightBox, we calculate such ensembles for decision attributes specified as the outputs of the diagnosed machine learning models. Then we define the similarity between pairs of data instances by means of checking to what extent they have the same combinations of attribute values for reducts in the ensemble. For reduct calculation, we adopted the ensemble-related algorithms reported in [28]. Although one may say that the applied algorithms have already existed, the ideas of using them to approximate machine learning models and to conduct neighborhood diagnostics based on the reduct ensemble-driven similarities are substantially new.

3. Basic notions and definitions

Let us assume that our data comes from a universe of possible instances U. Each instance $x \in U$ is characterized by a finite set of attributes $A = \{a_1, \dots, a_k\}$. Each attribute can be interpreted as a function $a : U \to V_a$, where V_a is a set of possible values. Depending on a type of V_a, attributes can be symbolic (discrete-valued), ordinal, or numeric (real-valued). We also discern a special attribute d, often called decision attribute or target. By $d(x)$ we will denote its value for an instance x and $[\cdot]_{d_i} = \{x' \in U : d(x') = d_i\}$. In Table 1, we introduce other notation that is used in the remaining parts of the paper.

[1] In [14], it was described how to integrate KnowledgePit and BrightBox but the details of the BrightBox technology were not exposed.

3.1. Prediction uncertainty measure

The prediction uncertainty is one of the most profound indicators of risks related to the operations of any machine learning model. Its appropriate estimation allows a better understanding of the model and may provide insights into the causes of prediction mistakes. For this reason, the uncertainty measure has a pivotal role in BrightBox.

Let us assume that for a given instance x, a prediction distribution $q(x)$ is a point of a simplex S with l vertices located at the standard basis of R^l. Pure states $P(S)$ correspond to the vertices of the simplex. They represent points of perfect information. Suppose we have a diagnosed model M: $U \rightarrow S$, returning a distribution over classes for a given instance from a data set. The probability mass function of a prior distribution $p \in S$ is the expected value of $q(x)$ when x is uniformly sampled from U.

We define a normalized entropy of a classification distribution $q(x)$ as

$$H_{norm}(q(x)) = -\frac{\sum_{i=1}^{l} q_i(x)\log(q_i(x))}{\log(l)}. \tag{1}$$

To make our uncertainty estimations in BrightBox comparable between various prediction models and data sets, we require that the uncertainty measure $Unc : S \rightarrow [0, 1]$ satisfies the following properties:

1. Pure states have zero uncertainty, i.e., $\forall_{q \in P(S)} Unc(q) = 0$.
2. Uncertainty is differentiable with one local and global maxima in prior distribution, i.e., $Unc(p) = 1$.
3. For a uniform prior, the uncertainty function should be equivalent to the normalized entropy, i.e., $(\forall_i p_i = [1/l]_i) \implies \forall_{q \in S} Unc(q) = H_{norm}(q)$.

In a case when the prior is not uniform, we need to transform it by scaling the simplex space. Let us take distribution norm of $q(x)$ with respect to p as $|q(x)|_p = \sum \frac{q_i(x)}{p_i}$. It measures the distribution $q(x)$ with the units of the prior distribution. Let us define a p-simplex off-centering, as

$$[OC_p(q_i(x))]_i = \frac{q_i(x)}{p_i\,|q(x)|_p}. \tag{2}$$

This function is a self-homeomorphism of the simplex and on each of its subsimplices. Therefore, it returns a proper distribution and keeps pure states. Moreover, it transforms the prior distribution into uniform, i.e., $OC_p(p) = u$, and is an identity for $p = u$.

If we transform classification distribution using Eq. (2) and calculate the normalized entropy using Eq. (1), we get prior-off-centered normalized entropy $OCE_p(q(x)) = H(OC_p(q(x)))$. We use it as the measure of uncertainty as it is satisfying all desired properties. For simplicity, from now on we will denote $OCE_p(q(x))$ as $Unc(x)$, which is the prediction uncertainty of model M calculated for a diagnosed instance x.

3.2. Model approximation

The core functionality provided by BrightBox is the ability to construct a surrogate model that can closely mimic any black-box classification or regression algorithm. We call such models *approximators* due to their aforementioned close connection to the theory of rough sets [15]. In particular, our approximations of predictions made by the investigated model M are computed using an ensemble of approximate reducts [21].

Definition 1. Let a universe of instances U be given in which all instances are characterized by attributes from A. An attribute subset $B \subseteq A$ determines a binary relation IND_B on U:

$$(x_i, x_j) \in IND_B \iff x_i, x_j \in U \wedge a(x_i) = a(x_j) \text{ for all } a \in B \tag{3}$$

In such a case, we say that x_i and x_j are indiscernible by B. An equivalence class of IND_B that contains a given $x \in U$ will be denoted by $[x]_B$.

Definition 2. Let a data set D be given in which all instances are characterized by attributes from A, and let $\phi_d : 2^A \rightarrow \mathbb{R}$ be a measure of functional dependency between attribute subsets and decisions, which is non-decreasing with regard to inclusion. A subset $AR \subseteq A$ is called an (ϕ_d, ε)-approximate reduct for an approximation threshold $\varepsilon \in [0, 1)$ iff the following conditions are met:

1. $\phi_d(AR) \geq (1 - \varepsilon)\phi_d(A)$.
2. There is no proper subset $AR' \subsetneq AR$ for which the first condition holds.

In other words, an approximate reduct is an irreducible subset of attributes that is sufficient to express almost the same information about the decisions as the whole attribute set. Examples of commonly used ϕ functions are mutual information or Gini impurity gain. It is worth highlighting that in our case, the decision values for which we construct the approximate reducts correspond to predictions of the diagnosed model, i.e., $M(X_D)$, not the actual ground truth target values.

Approximate reducts can be used as a classification model. For an instance $x \in U$ and a reduct AR, decision class probabilities can be calculated as:

$$\overline{q^{AR}}(x) = (\overline{q_1^{AR}}(x), \ldots, \overline{q_l^{AR}}(x)), \tag{4}$$

such that

$$\overline{q_i^{AR}}(x) = \begin{cases} \frac{|[x]_{AR} \cap X_D \cap [\cdot]_{d_i}|}{|[x]_{AR} \cap X_D|} & \text{if } [x]_{AR} \cap X_D \neq \emptyset, \\ p_i & \text{if } [x]_{AR} \cap X_D = \emptyset. \end{cases}$$

In the equation above, $[x]_{AR}$ denotes a set of all instances in the same indiscernibility class as x with regard to the attributes from AR, and $[\cdot]_{d_i}$ is an equivalence class of d_i in U.

Since ensemble classifiers usually achieve better classification accuracy than individual models, BrightBox uses an ensemble of approximate reducts to improve approximations of diagnosed models. The prediction is calculated independently for each AR and the results are averaged. More specifically, decision class probabilities calculated for an ensemble of reducts $\mathcal{R}$ are computed as:

$$\overline{q^{\mathcal{R}}}(x) = |\{AR \in \mathcal{R} : [x]_{AR} \cap X_D \neq \emptyset\}|^{-1} \sum_{\{AR \in \mathcal{R} : [x]_{AR} \cap X_D \neq \emptyset\}} \overline{q^{AR}}(x) \tag{5}$$

and in the case when $\{AR \in \mathcal{R} : [x]_{AR} \cap X_D \neq \emptyset\} = \emptyset$, we assume that $\overline{q^{\mathcal{R}}}(x) = p$, i.e. out approximation corresponds to the prior distribution.

For simplicity, from now on we will denote $\overline{q^{\mathcal{R}}}(x)$ as $\overline{q^M}(x)$, which is the approximation of decision class probabilities predicted by M for an instance x. Then, $\overline{M}(x) = \arg\max_i \overline{q_i^M}(x)$ is the approximated prediction for x. If all elements of the vector $\overline{q^M}(x)$ have the same value, we assume that the approximate prediction for x is the result of the arg max function of the prior distribution.

3.3. Computation of neighborhoods

Let us assume that we have an approximator of a diagnosed model M, composed of a set of approximate reducts $\mathcal{R} = \{AR_1, \ldots, AR_i, \ldots, AR_k\}$. Since we want to identify instances that are similar with regard to predictions made by M, we define a notion of neighborhood based on the prediction process of the model approximator $\overline{M}$. In particular, the neighborhood $N^{AR}(x)$ for a diagnosed instance x with regard to a single reduct $AR \in \mathcal{R}$ is defined as a subset of instances from X_R that belong to the same indiscernibility class, i.e. $[x]_{AR}$. The final neighborhood is the sum of neighborhoods computed for all reducts in the ensemble:

$$N^{\mathcal{R}}(x) = X_R \cap \left(\bigcup_{AR \in \mathcal{R}} [x]_{AR} \right). \tag{6}$$

We abbreviate this notation to $N(x)$ when $\mathcal{R}$ is obvious from the context.

We can also assign a weight to each instance $x_R \in N(x)$ by counting the number of reducts for which it appears in the same indiscernibility class as x:

$$w(x_R, x) = \frac{|\{i : x_R \in [x]_{AR_i}\}|}{|\mathcal{R}|} \tag{7}$$

where $x_R \in N(x)$. We use these weights in further calculations of diagnostic attributes described in Section 4.2. It is worth noting that the neighborhoods are composed only of instances from the reference data X_R, i.e. $\forall_{x \in X_D} N(x) \subset X_R$.

4. Model diagnostics

4.1. Surrogate model construction procedure

The construction of our surrogate model in BrightBox, described in Section 3.2, requires that all attributes have discrete values. Thus, we start the approximation procedure by discretizing all numeric attributes using the quantile method [29] — all resulting intervals contain approximately the same number of instances. We construct the ensemble of approximate reducts for instances from the data set $(X_D, M(X_D))$.

The construction of the approximator depends on the selection of two hyper-parameters: an ϵ representing the approximation threshold for reducts, and a number of reducts in the ensemble. Since the aim of the procedure is to find the appropriate approximation of the model's predictions, we use the grid search to tune the hyper-parameter settings. The final selection of the surrogate model is made when the approximation quality measured with Cohen's Kappa reaches at least 0.9. If the desired quality cannot be achieved, we train the reduct ensemble with settings that ensure the highest possible approximation quality. The resulting approximator is later used to determine neighborhoods of instances from the diagnosed data and to compute values of our diagnostic attributes.

4.2. Diagnostic attributes

We compute a number of diagnostic attributes to characterize each instance $x \in X_D$. Values of these attributes are derived by analyzing observations from instances' neighborhoods $N(x) \subseteq X_R$. In this process, we only assume access to predictions of the diagnosed model M for the diagnosed data table, i.e., $M(X_D)$, and the availability of data sets $D = (X_D, d_D)$ and $R = (X_R, d_R)$. We describe the most important diagnostic attributes below.

Neighborhood size: The neighborhood size for instance x is computed as a fraction of the size of the set X_R, i.e., $\frac{|N(x)|}{|X_R|}$.

Uncertainty: The prediction uncertainty of the model M calculated for x $(Unc^M(x))$ is estimated according to the method described in Section 3.1. If we do not have access to information about the estimated decision class probabilities made by the diagnosed model $q(x)$ for all diagnosed instances, we use the approximation of decision class probability predictions denoted as $\overline{q}(x)$.

Before we describe the remaining attributes, it is necessary to define a measure that expresses the proximity of a discrete probability distribution to some prior distribution:

$$h(p, q(x)) = \frac{H(p)}{H(p, q(x))}, \tag{8}$$

where the numerator is the entropy of the prior distribution of decision classes p:

$$H(p) = -\sum_{i=1}^{l} p_i \log p_i. \tag{9}$$

In the above formula, l is the number of classes and p_i corresponds to the probability of ith decision class. The denominator in Eq. (8) is the cross-entropy between the distributions p and q:

$$H(p, q(x)) = -\sum_{i=1}^{l} p_i \log q_i(x), \tag{10}$$

where $q(x)$ is the distribution of decision class probabilities predicted by the diagnosed model M for the instance x, and $q_i(x)$ corresponds to the ith class.

Targets diversity in neighborhood: It is calculated as $h(p, p(N(x)))$, where $p(N(x))$ is the distribution of decision classes in the neighborhood of x.

Approximations diversity in neighborhood: It refers to $h(p, \overline{q}(N(x)))$, where $\overline{q}(N(x))$ is the distribution of approximated predictions in $N(x)$.

Target consistency with targets in neighborhood: It expresses how often values from $d_{N(x)}$ agree with the class of x, i.e. d_x.

Target consistency with approximations in neighborhood: This attribute expresses how often values from $\overline{M}(N(x))$ are the same as d_x.

Prediction consistency with targets in neighborhood: It expresses how often values from $d_{N(x)}$ are the same as $M(x)$.

Targets and approximations inconsistency in neighborhood: This attribute is calculated as $\frac{1}{|N(x)|} \sum_{x' \in N(x)} \left[1 - \overline{q_{d_{x'}}}(x') \right]$, where $d_{x'}$ is the ground truth target class of x', and $\overline{q_{d_{x'}}}$ is the approximation of its probability.

4.3. Diagnostic method workflow

Following the procedure from Section 4.1, we use the set of instances X_D and diagnosed model predictions for these instances $M(X_D)$ to prepare an approximator of model M. We calculate the approximations and neighborhoods for all instances. We also compute diagnostic attributes defined in Section 4.2.

In the second step, we run a global diagnostics of the model M. For this purpose, we compute a summary of the diagnostic attribute values obtained for X_D, and we use a pretrained classifier to assign M into one of three categories: *Near optimal fit*, *Under-fitted model*, and *Over-fitted model*. The classifier has been pretrained on a collection of diagnostic attribute summaries computed for a large number of data sets and commonly used prediction models. Currently, we are using a simple random forest classifier for this task, however, other types of models, including classifier ensembles can be applied. More details regarding the computed summaries and the training procedure for the global diagnostic model are given in Section 5.3.

In the next step, BrightBox diagnostics focuses on the investigation of individual data instances and predictions made by M. We use the previously computed diagnostic attribute values and the output of the global diagnostic model to provide a comprehensive analysis of model predictions and potentially related issues. Additionally, we apply a set of local diagnostic rules defined by experts to provide end-users with accessible insights. A few examples of such rules and fix recommendations are listed below:

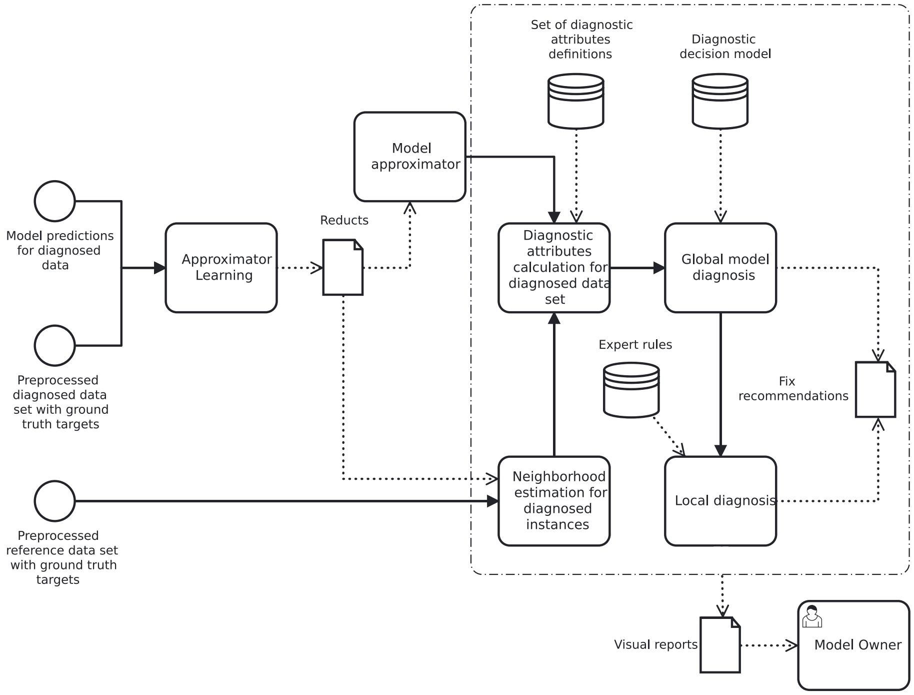

Set of diagnostic attributes definitions
Diagnostic decision model
Model approximator
Model predictions for diagnosed data
Approximator Learning
Reducts
Diagnostic attributes calculation for diagnosed data set
Global model diagnosis
Preprocessed diagnosed data set with ground truth targets
Expert rules
Fix recommendations
Neighborhood estimation for diagnosed instances
Local diagnosis
Preprocessed reference data set with ground truth targets
Visual reports
Model Owner

Fig. 1. Diagram of diagnostic method workflow.

- If the model was diagnosed as over-fitted, and for a given erroneously classified instance model's uncertainty was low, and its neighborhood was not small, then it is likely that the mistake was caused by over-fitting. Improve the model fitting procedure.
- If the model was diagnosed as under-fitted, and for a given erroneously classified instance model's uncertainty was high, and its neighborhood was not small, then it is likely that the mistake was caused by under-fitting. Improve the model fitting procedure.
- If the diagnosed instance has a very small neighborhood, i.e., is dissimilar to training instances, then it is likely that the instance was an outlier. Provide similar training instances for the model.

In the last step, BrightBox prepares visual reports that highlight the most significant findings. Visualizations showcase all relevant statistics related to the diagnosed model and its approximator quality, as well as distributions of diagnostic attributes. Interactive plots allow end-users to investigate diagnoses for individual instances and analyze their statistics for selected groups. Finally, the report provides useful information on the importance of original attributes that is approximated by the importance of attributes in the surrogate model. An overview of the whole BrightBox workflow is presented in Fig. 1.

5. Experiments and results

5.1. Data sets used in experiments

We started the experimental evaluation of the BrightBox technology with the preparation of benchmark data sets. In total, we used 23 benchmark data sets describing various classification tasks, ranging from binary classification to multi-class problems. Data sets were obtained from an open repository OpenML.[2] Their basic characteristics are presented in Table 2. The benchmark data sets were chosen such that each predefined test partition (which we are using to diagnose the models) contained at least 100 data instances.

For each of the selected data sets, we fitted seven prediction model types, i.e., lasso regression, SVM with Gaussian kernel, naive Bayes, decision tree, multi-layer perceptron, random forest, and XGBoost. Each model was fitted with three different hyper-parameter settings — corresponding to different model generalization capability levels. These hyper-parameter settings were tuned separately for each data set, and the resulting models were labeled by four independent experts with one of four possible labels:

- **Near optimal fit** — the performance of the model is close to the best possible prediction performance reported in the literature for a given data set.
- **Under-fitted model** — the model was over-generalized or not sufficiently fitted to the available training data. Typically, it manifests in relatively low prediction quality on both training and validation data.
- **Over-fitted model** — the model was closely fitted to the training data, however, its generalization quality (measured on a validation set) was poor.
- A border case model — this label was used if an expert could not decide which of the three previous labels should be assigned.

Predictions of the fitted models were analyzed using the methodology described in Section 4. In particular, we computed the model approximations and diagnostic attribute values for each test instance from diagnosed (validation) sets, as described in Section 4.2. We run the diagnostics only for the models for which at least three out of four experts had assigned the same label. Thus, we obtained 440 sets of diagnosed (data set, model) pairs with labels from the set {*Near optimal fit*, *Under-fitted model*, *Over-fitted model*}. The total number of instances in the resulting sets was 419,699. We use this data in two experiments that aim to evaluate the ability of BrightBox to distinctively describe erroneous predictions, and the ability to distinguish between models with different generalization capabilities.

[2] https://www.openml.org/

Table 2
Basic characteristics of data sets used in experiments. The columns N, $|A|$, and $|L|$ show the total number of instances, attributes, and classes, respectively.

Name	N	$\|A\|$	$\|L\|$	Distribution of classes
Bioresponse	3751	1776	2	.458; .542
churn	5000	20	2	.859; .141
cmc	2000	47	10	.1; .1; .1; .1; .1; .1; .1; .1; .1; .1
cnae-9	1080	856	9	.11; .11; .11; .11; .11; .11; .11; .11 .11;
dna	3186	180	3	.24; .24; .52
har	10299	561	6	.167; .149; .137; .173; .185; .189
madelon	2600	500	2	.5; .5
mfeat-factors	2000	47	10	.1; .1; .1; .1; .1; .1; .1; .1; .1; .1
mfeat-fourier	2000	76	10	.1; .1; .1; .1; .1; .1; .1; .1; .1; .1
mfeat-karhunen	2000	47	10	.1; .1; .1; .1; .1; .1; .1; .1; .1; .1
mfeat-morphological	2000	6	10	.1; .1; .1; .1; .1; .1; .1; .1; .1; .1
mfeat-zernike	2000	47	10	.1; .1; .1; .1; .1; .1; .1; .1; .1; .1
nomao	34465	118	2	.286; .714
optdigits	2000	47	10	.1; .1; .1; .1; .1; .1; .1; .1; .1; .1
pendigits	10992	16	10	.104; .104; .104; .096; .104; .096; .096; .104; .096; .096
phoneme	5404	5	2	.707; .293
qsar-biodeg	1055	41	2	.66; .34
satimage	6430	36	6	.238; .109; .211; .097; .110; .235
semeion	1593	256	10	.1; .1; .1; .1; .1; .1; .1; .1; .1; .1
spambase	2000	47	10	.1; .1; .1; .1; .1; .1; .1; .1; .1; .1
wall-robot-navigation	5456	24	4	.404; .385; .060; .151
wdbc	569	30	2	.63; .37
wilt	4839	5	2	.946; .054

Moreover, for each of these 440 data sets, we examined the execution time of the entire workflow presented in Fig. 1. Obtained results show that run time depends on the number of instances and attributes. The overall average is 29 s with a high standard deviation of 56 s. It shows a very strong diversity of results. For a better understanding, we divided the analyzed data sets into three clusters:

1. small data sets (number of instances $\leq$ 10000, number of attributes $\leq$ 500 — 12 s on average with a standard deviation of 11 s,
2. data sets with a number of attributes $>$ 500 — 95 s on average with a standard deviation of 42 s,
3. data sets with a number of instances $>$ 10000 — 240 s on average with a standard deviation of 77 s.

Calculations were conducted on a machine with 8 CPUs and 32 GB RAM.

5.2. Verification of mistake identification capabilities

Our first experiment was aimed at the verification of the ability of BrightBox to distinctively describe erroneous predictions. To this end, we prepared a test that checks if the representation of instances by vectors of our diagnostic attribute values preserves the similarity in the context of mistakes made by the diagnosed model. In particular, we would like to make sure that if the model made a mistake for a particular instance, the description of such an instance in terms of diagnostic attributes will likely be more similar to some other erroneous data instances than to instances for which the model worked correctly.

To test it, for data sets described in Table 2, we checked the accuracy of the 1-nearest neighbor classifier in recognizing erroneous predictions based on diagnostic attribute values. We conducted this experiment in three different evaluation setups corresponding to realistic application scenarios:

1. We estimated the classification performance using the *leave-one-data-set-out* test. This experiment allows us to verify the usefulness of our diagnostic attributes in a scenario when we want to analyze a known prediction model type on a new, previously unseen data set.
2. We estimated the classification performance using the *leave-one-model-out* test. This scenario corresponds to a situation when we analyze a completely new type of classification model. However, we still have a chance to run some standard diagnostic benchmarks on the available data beforehand.
3. We estimated the classification performance using the standard cross-validation test with a stratified division of data between folds. This standard benchmark scenario was included as a reference. It corresponds to a situation when we want to diagnose a commonly used type of predictive model on a previously known data set.

Table 3
Evaluation results of the 1-nearest neighbor classifier for the model mistake identification task. The average results of the experiment are presented with standard deviations in the brackets.

Measure	Scenario 1	Scenario 2	Scenario 3
Accuracy	0.965 (0.041)	0.987 (0.006)	0.988 (0.010)
Balanced accuracy	0.941 (0.075)	0.976 (0.014)	0.979 (0.023)
Cohen's Kappa	0.881 (0.127)	0.947 (0.017)	0.951 (0.032)

Before the experiment, the diagnostic attributes were linearly scaled to the [0, 1] interval. The nearest neighbor algorithm was using the Euclidean distance, and due to the imbalanced distribution of mistakes, its performance was measured using three different metrics, i.e., standard accuracy, balanced accuracy, and Cohen's Kappa coefficient. Results of this experiment in all three evaluation scenarios are presented in Table 3.

The results of these experiments clearly show that our diagnostic attributes are able to sufficiently characterize incorrectly classified instances to distinguish them from instances for which the diagnosed model worked well. It provides a strong argument supporting the soundness of our approach to the diagnostics of predictive models. Noticeable is the fact that the results obtained in the first evaluation scenario are significantly worse than the results for the other two. It suggests that running model diagnostics for completely new data is a much harder problem than analyzing a new model type. This is a valuable insight for our future works on the development of BrightBox technology.

Table 4
Evaluation results of the random forest classifier for the model generalization capability prediction task. Average results and estimations of standard deviations (in brackets) from ten repetitions of the experiment are presented.

Measure	Scenario 1	Scenario 2	Scenario 3
Accuracy	0.717 (0.055)	0.729 (0.039)	0.791 (0.025)
Balanced accuracy	0.721 (0.059)	0.728 (0.044)	0.794 (0.028)
Cohen's Kappa	0.559 (0.082)	0.578 (0.067)	0.673 (0.038)

5.3. Verification of the global diagnostic capabilities

In the second series of experiments, we aimed to verify the global diagnostic capabilities of our system, i.e., the ability to distinguish between models with different generalization capabilities. We used the previously prepared 440 (data set, model) pairs with manually assigned labels from the set {*Near optimal fit*, *Under-fitted model*, *Over-fitted model*}. The labels were assigned to each pair based on voting between four experts, as described in Section 5.1.

For each (data set, model) pair, we aggregated the corresponding values of diagnostic attributes using simple statistics, such as the mean, standard deviation, min, max, and quantiles. We also added some basic characteristics of each data set, e.g., the number of instances, and the number of decision classes. Finally, we included information about the estimated prediction efficiency of the diagnosed model, i.e., the model's balanced accuracy estimated on the diagnosed data set. In this way, for each pair, we obtained a single labeled instance that we could use for the global diagnostics of the corresponding prediction model.

To assess the global diagnostic capabilities of BrightBox, we conducted a similar test to that described in Section 5.2. We considered the same evaluation scenarios. This time, however, we trained a simple prediction model, i.e. the random forest classifier, using default parameter settings apart from the number of fitted trees, which we increased to 1000. We experimentally checked that using a higher number of trees brings negligible improvements in classification accuracy.

Since fitting a random forest classifier is a non-deterministic process, we repeated the experiments for each evaluation scenario ten times to obtain accurate estimates of the performance. Table 4 shows the obtained evaluation results — as in the previous experiment, we measured the accuracy, balanced accuracy, and Cohen's Kappa metrics.

The results clearly show that the global diagnostic of prediction models using our diagnostic attributes is feasible. All considered measures indicate that random forest is able to distinguish between the three considered model classes significantly better than random or naive (majority) predictions. Since in the discussed experiments we did not try to optimize the hyperparameter settings or select the most efficient prediction model for this task, we believe that further improvements of the presented metrics are possible.

As in the previous experiment, the comparison of the performance between different scenarios shows that the global diagnostic of a prediction model is most difficult when it is done on a completely new data set (the first scenario). For all metrics, the results for scenario 1 were significantly lower than for scenario 2, i.e., the *p*-value of a paired, one-sided Wilcoxon rank test was $\leq$ 0.01 for the accuracy and Cohen's Kappa measures, and it was $\leq$ 0.02 for the balanced accuracy measure. The differences between scenarios 2 and 3 were even greater.

We also investigated the influence of individual attributes on the performance of global model diagnostics. We calculated Shapley values for the model's attributes, and then we aggregated them by individual diagnostic attributes. The influence of diagnostic attributes on the predictions for each label is presented in Fig. 2. One of the expected findings is the fact that models with a large balanced accuracy value are more likely to be classified as *Near optimal fit* and less likely as over- or under-fitted. There is also an intuitive dependency between the neighborhood size and predictions into the over-fitted and under-fitted class. In particular, a dominance of small neighborhoods makes a model more likely to be classified as over-fitted, whereas if the most of neighborhoods are very large, then the model is more likely to be classified as under-fitted.

6. A case study in cybersecurity domain

To further evaluate our approach, we checked how BrightBox technology works in a practical application related to cybersecurity. We chose a real-world problem that was a topic of our data science competition *IEEE BigData 2019 Cup: Suspicious Network Event Recognition*. This event was organized jointly by companies Security On-Demand (SOD) and QED Software using the aforementioned KnowledgePit platform.[3] The competition outcomes enabled SOD to design new machine learning models helping in their operations [30].

Companies such as SOD develop comprehensive systems for monitoring and analyzing the network traffic of their customers. Such systems generate vast amounts of data in the form of network event logs. To identify cyber threats, this data needs to be parsed, processed, and aggregated into higher-level events – so-called alerts – which need to be further analyzed by experts working at Security Operations Centers (SOCs). The task of the IEEE BigData 2019 Cup challenge was to predict which of the investigated alerts were considered truly suspicious by the SOC team and led to issuing a notification to SOD's clients. Data for this competition was provided by SOD in form of three separate sets which reflect the stages of network traffic processing. A detailed description of the competition data and preprocessing steps applied in the construction of the extended baseline for this competition can be found in [31]. We closely followed this preprocessing procedure in our experiment.

6.1. Diagnosis of an over-fitted model

We aimed to verify how much insight can we get about prediction models constructed for scoring the alerts issued by SOD's monitoring systems to prioritize them for SOC experts. Firstly, we trained an XGBoost model on the preprocessed data, since this type of prediction algorithm was the most popular choice during the competition [31]. We deliberately set the hyper-parameters so that the resulting model fitted the training data very closely, but generalized poorly to other data used for evaluation. In particular, we trained a tree-based XGBoost model with 1000 trees, a maximal tree depth set to 6, and a very high learning rate of 0.3. We assessed the model's predictions on the training and test sets from the competition. Due to the imbalanced distribution of the positive class (i.e, only $\approx$ 5.7% of investigated alerts were marked as suspicious and reported to SOD's clients), we used for the evaluation AUC and balanced accuracy (BAC) metrics. Table 5 shows the obtained evaluation results.

Before we diagnosed the model, we measured the quality of its approximation on the diagnosed set using Cohen's Kappa coefficient. The resulting approximation quality was 0.98, which means that it was good enough and does not exceed the assumptions of the BrightBox technology. We also calculated some characteristics of the neighborhoods generated by the approximate to check their stability and verify the impact of the chosen

[3] https://knowledgepit.ai/suspicious-network-event-recognition/

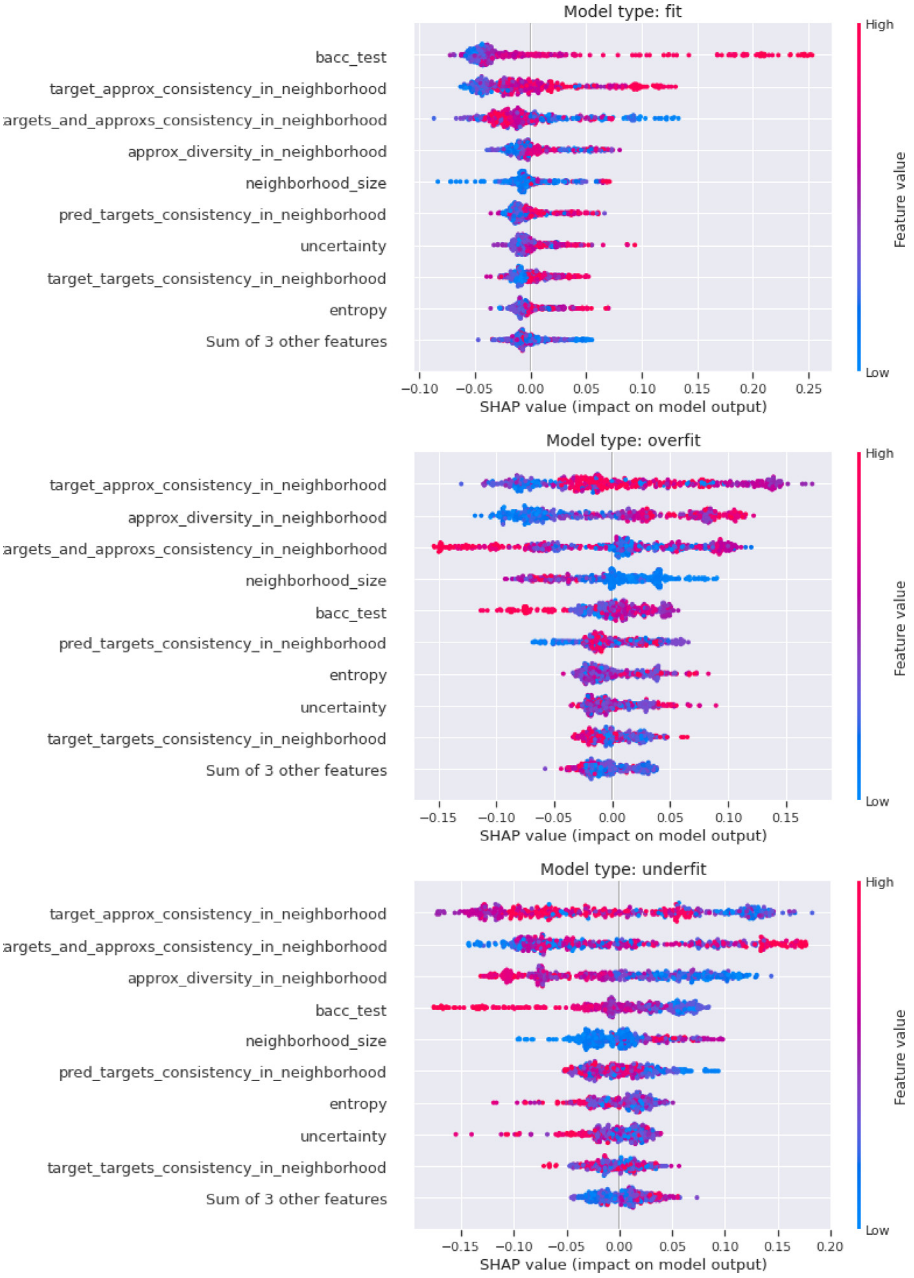


Fig. 2. The influence of diagnostic attributes on the prediction of generalization capabilities of diagnosed prediction models using the random forest classifier. To create these figures, a random forest classifier was trained on all available data.

approximation hyper-parameters. Fig. 3 shows how the number of empty neighborhoods and neighborhood similarity were changing with the growing number of reducts included in the constructed approximation of the predictions. In our approach, we treat the reduct ensemble as a bagging estimator, so a greater number of reducts implies more consistent results. The described case is based on a large data set in terms of the number of instances (59427) and conditional attributes (455). We observe that for 1000 reducts there are instances in the diagnosed data set that are not associated with any observation from the reference data set. With an increasing number of reducts, it is possible to find different indiscernibility classes that cover these instances. The same mechanism is observed for the calculation of Jaccard neighborhood similarity between successive batches of reducts. It shows that the variance of this measure is decreasing and that alignment is very close to the maximum possible value.

Table 5
Evaluation results of the diagnosed scoring model on the training and test data. The model is nearly perfectly fitted to the training set, but its performance on test data is considerably lower.

Measure	Training set	Testing set
AUC	0.994	0.879
Balanced accuracy	0.968	0.774

After the approximation procedure was performed, the further diagnostic procedure was continued. As expected, the obtained global diagnosis was *Over-fitted model*. Moreover, as an additional verification layer, we measured the attribute importance of the

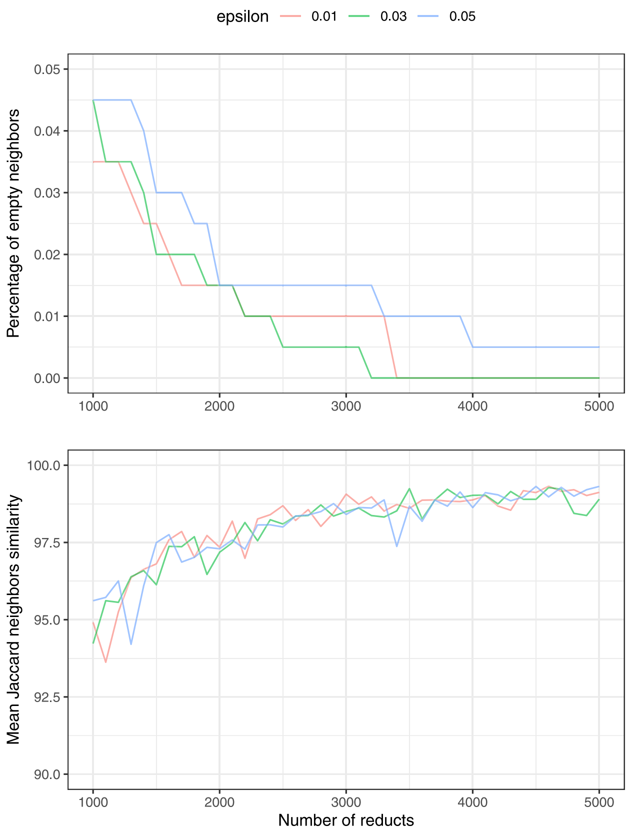


Fig. 3. Analysis of the number of empty neighborhoods and average neighborhood similarity for a growing number of reducts used for the construction of approximations of over-fitted predictions on the cyber-security data (SOD).

approximator, and we compared these results with the feature importance of the original model. The results are shown in Fig. 4. The plots show that the attribute importance of the approximator and the diagnosed model are similar. In particular, nearly all features in the top 20 of obtained rankings appear on both plots. We computed Pearson's correlation and Spearman's rank correlation between the obtained attribute importance values. The resulting coefficients were 0.7620 and 0.5822, respectively.

When investigating the distribution of final diagnoses issued by BrightBox, we also noticed that the vast majority of errors made by the model (i.e., over 95% errors within *Class 0* and 70% within *Class 1*) were labeled as cases of erroneous predictions far from the decision boundary. This is also something that we would expect to see in predictions of a highly over-fitted model.

6.2. Comparison with a diagnosis of the SOTA model

For comparison with the model diagnosed as over-fitted, we performed a similar diagnostic procedure for predictions made by the winners of the IEEE BigData 2019 Cup [32]. The AUC metric for this solution was 0.932 and BAC was 0.854, compared to 0.879 and 0.774, respectively, obtained by our over-fitted model. Since in this case, we did not have access to the diagnosed classifier – only to the model's predictions submitted to the KnowledgePit.ai platform – this experiment corresponds to a real-world application scenario when our system is deployed for the diagnosis of solutions in a data science competition.

The module responsible for global diagnostics labeled the predictions as *Near optimal fit*. The quality of the model approximation on the diagnosed set, measured with Cohen's Kappa coefficient, was 0.98. Of all neighborhoods, only 3.75% were empty and the average neighborhood size was 61. Fig. 5 shows the generated attribute importance computed using the constructed approximator.

Comparing the results of the attribute importance of the approximators presented in Figs. 4 and 5, we can notice that the predictions diagnosed as *Near optimal fit* were much more relying on the *overallseverity* and *untrustscore* features (their importance was much greater) than those obtained for the over-fitted model. These two features are indeed important indicators that are commonly used by cybersecurity experts. Moreover, the SOTA model does not rely much on the *suspicious_smb_activity* alert type indicator, which was one of the top 20 important features used by the over-fitted model. This is also reasonable because that particular alert type is quite common (its severity is low) and is often raised by typical network activities of regular users.

The observed distribution of diagnoses over the erroneous predictions was considerably different than in the case of the over-fitted model. Overall, the SOTA predictions had a much greater recall and slightly lower precision. We also noticed that there were much fewer instances labeled by BrightBox as located far from the decision boundary and having small neighborhoods, e.g., there were almost no such cases among instances from *Class 0*. A large number of such instances is typically a strong indication of over-fitting predictions. Fig. 6 shows the distribution of BrightBox labels across target values in the data.

Lastly, we analyzed diagnosed data instances with regard to the similarity of their neighborhoods. Fig. 7 shows a UMAP visualization [33] of data in which the Jaccard similarity of neighborhoods is used to express the proximity of instances. The data instances corresponding to erroneous predictions are grouped in the center part of this visualization. A closer investigation of this data by domain experts could reveal insightful information about characteristics of instances that have a significant impact on the risk of prediction errors.

7. Conclusions and future work

We presented technology for comprehensive diagnostics of machine learning models. In our approach, ensembles of simple rough set-based reducts [21] are used for black-box machine learning model approximations. Operating with such surrogate models lays in the heart of XAI [7], whereby black-box models are approximated with models that are easier to interpret. However, in our case such ensembles are utilized for something more — they are the basis for deriving, for each new instance, the collections of historical instances that are processed in a similar way. Such collections, called neighborhoods, let us categorize mistakes of the diagnosed models. This kind of analysis allows us to take better decisions in the maintenance process, and thus can be useful in applications where the quality of machine learning models is of high importance.

The conducted experiments allow for conclusions on the viability and usefulness of the presented approach in the diagnosis and monitoring of machine learning models — by both general analyses of particular model's errors, as well as by global diagnostic capabilities. A point of note is that the main strength of the presented technology lies in the fact that the diagnosis relies on the analysis of neighborhoods, not attribute values themselves. Due to this feature, we were able to obtain over 70% accuracy in the identification of models that over/under-fit to available training data. We have also demonstrated that our diagnostic attributes allow for capturing insightful characteristics of diagnosed data instances that could help domain experts and data scientists in constructing improved versions of their prediction models.

To summarize, we demonstrated a novel XAI technology, where the only assumptions are access to a sufficiently large

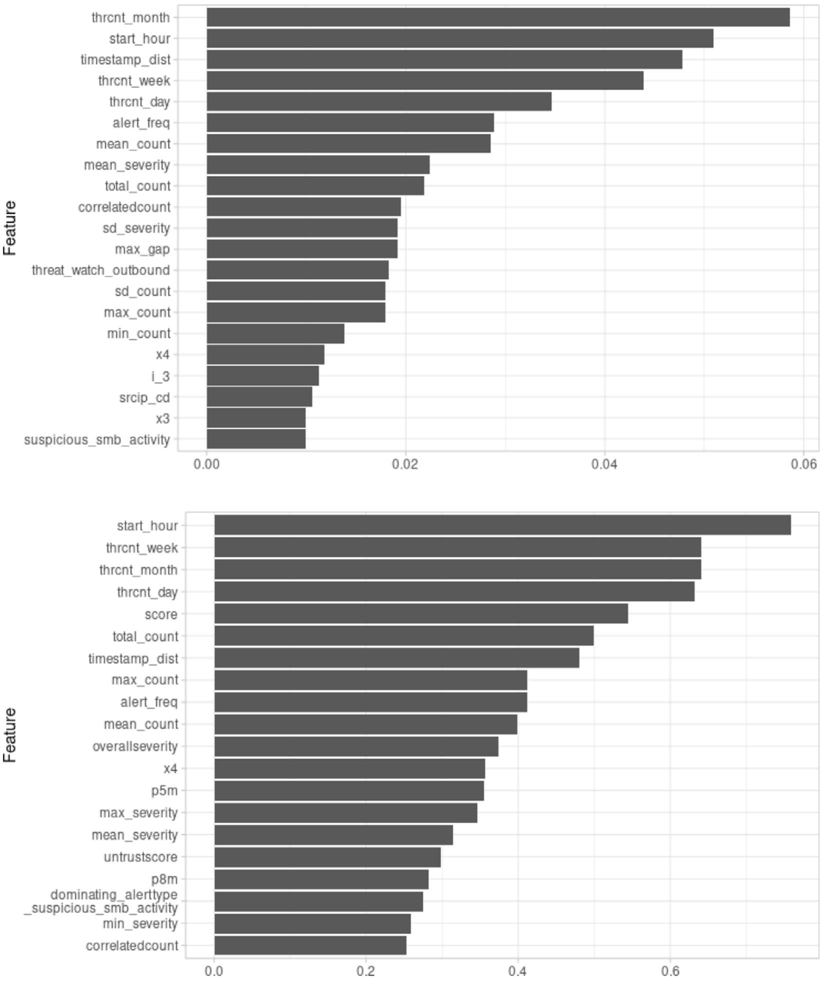


Fig. 4. Attribute importance of the model that we purposely trained to over-fit the training data set (top) and the attribute importance of the model's approximator (bottom).

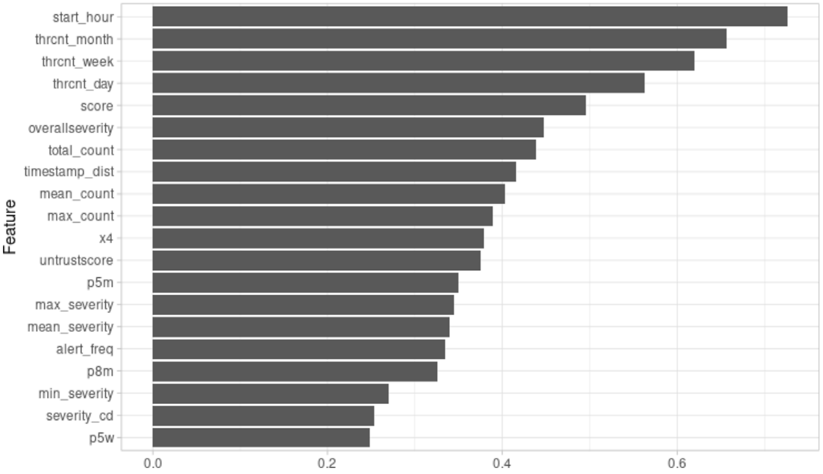
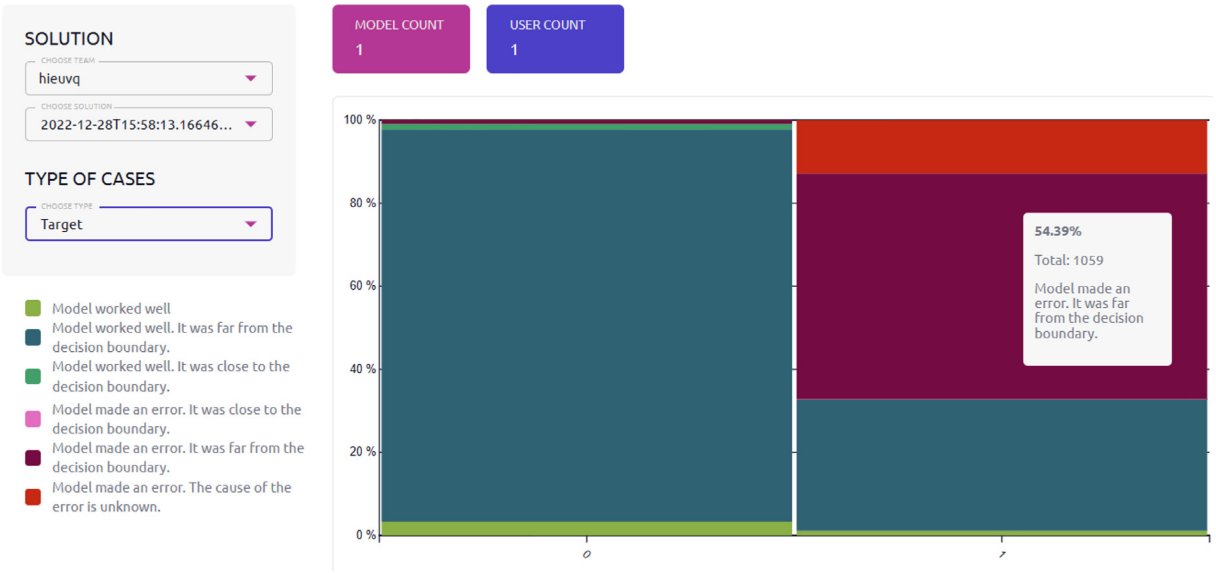


Fig. 5. Attribute importance estimated for the winning solution from the IEEE BigData 2019 Cup. Only the model's predictions were used to compute the approximator.

data set with diagnosed model's predictions. Because our solution is based on rough set-based reducts, its limitations stem from the necessity of working with discretized data. Furthermore, current approach focuses solely on diagnosing a prediction model, without producing any recommendations related to its possible improvements.


SOLUTION

CHOOSE TEAM: hieuvq

CHOOSE SOLUTION: 2022-12-28T15:58:13.16646...

TYPE OF CASES

CHOOSE TYPE: Target

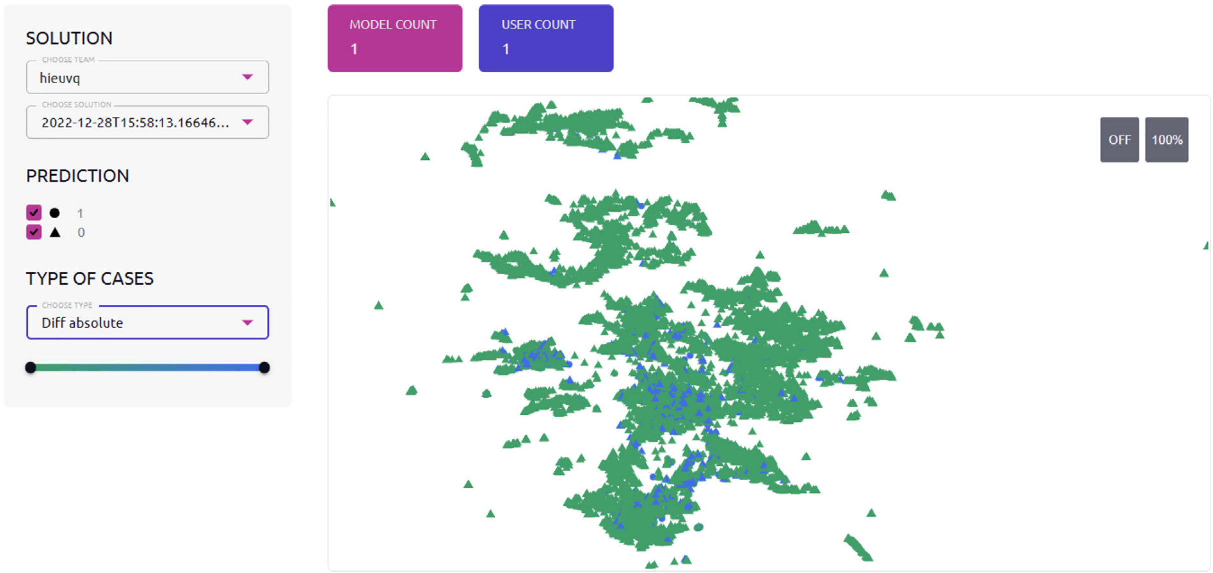
MODEL COUNT 1

USER COUNT 1

- 🟩 Model worked well
- 🟦 Model worked well. It was far from the decision boundary.
- 🟩 Model worked well. It was close to the decision boundary.
- 🟪 Model made an error. It was close to the decision boundary.
- 🟪 Model made an error. It was far from the decision boundary.
- 🟥 Model made an error. The cause of the error is unknown.

54.39%
Total: 1059
Model made an error. It was far from the decision boundary.

Fig. 6. A distribution of BrightBox labels across diagnosed instances from different target classes, computed for the winning solution of IEEE BigData 2019 Cup.


SOLUTION

CHOOSE TEAM: hieuvq

CHOOSE SOLUTION: 2022-12-28T15:58:13.16646...

PREDICTION

☑ ● 1
☑ ▲ 0

TYPE OF CASES

CHOOSE TYPE: Diff absolute

MODEL COUNT 1

USER COUNT 1

Fig. 7. A visualization of IEEE BigData 2019 Cup data computed using the UMAP algorithm with the Jaccard similarity of neighborhoods used as the instance proximity measure.

The future work will continue in many distinctive areas. Firstly, we will focus on enhancing our technology to cover regression problems. We will develop local (per instance) diagnostic capabilities, explanations, and recommendations for model fixtures. We will also verify the presented approach on varied data sets, including different data types reflecting the areas such as e.g. computer vision [9] or sensor measurements [24]. Herein, it will be important to integrate our methods with various feature extraction techniques so it is possible to compute ensembles of reducts for images, time series data, and so on.

Going further, we plan to verify our approach on diverse examples of erroneous data — this workstream will include synthetic generation of noises in data sets, attempting to reproduce various mistake types relating to real-world scenarios. We will focus on the development of tools for drift monitoring, e.g. shift in the distribution of data and exploration of the usefulness of our technology for detecting adversarial attacks on machine learning models. Our works will also include the aspects related to the exploration of performance boundaries for the presented technology, with a goal of achieving computational performance allowing for large-scale diagnostics tasks, e.g. for the purpose of storage and scanning of historical data for diagnostics analysis. In a parallel work stream, we want to explore the possibility of using the method presented in this work as a basis for a prescriptive analytical tool that will recommend changes in the data processing or in the model, resulting in meaningful improvements. This stream of work will require combining the black-box and white-box approaches. Currently, we are working on releasing a public demo with the possibility of testing the described solution. It will be available at bbdemo.qed.pl.

Finally, we intend to continue working on practical integrations of the BrightBox technology, such as the one described in [14]. Therein, BrightBox can be useful to diagnose mistakes of models submitted as solutions to online data science competitions. In this way, the organizers and sponsors of such competitions can be provided with richer insights into the way particular solutions perform. This is important from the perspective of deployment of such solutions (after improvements) in production-ready environments [31].

CRediT authorship contribution statement

Andrzej Janusz: Conceptualization, Methodology, Validation, Formal analysis, Data curation, Investigation, Visualization, Writing – original draft, Writing – review & editing, Supervision. **Andżelika Zalewska:** Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Visualization, Supervision, Writing – original draft, Writing – review & editing. **Łukasz Wawrowski:** Software, Validation, Investigation, Data curation, Visualization, Writing – original draft, Writing – review & editing. **Piotr Biczyk:** Writing – original draft, Writing – review & editing. **Jan Ludziejewski:** Methodology, Validation, Formal analysis, Software, Writing – original draft. **Marek Sikora:** Conceptualization, Writing – original draft, Writing – review & editing. **Dominik Ślęzak:** Conceptualization, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Andrzej Janusz reports financial support was provided by National Centre for Research and Development.

Data availability

Data sets were obtained from OpenML repository or from KnowledgePit.ai data science competition platform. All data sets are publically available and appropriate linked were included in the manuscript

Acknowledgments

This research was co-funded by the Polish National Centre for Research and Development in the frame of project MAZOWSZE/0198/19.

References

[1] W.Y. Ayele, Adapting CRISP-DM for idea mining: A data mining process for generating ideas using a textual dataset, Int. J. Adv. Comput. Sci. Appl. 11 (6) (2020) http://dx.doi.org/10.14569/IJACSA.2020.0110603.

[2] J. Zhang, Y. Wang, P. Molino, L. Li, D.S. Ebert, Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models, IEEE Trans. Vis. Comput. Graphics 25 (1) (2018) 364–373.

[3] A. Aggarwal, P. Lohia, S. Nagar, K. Dey, D. Saha, Black box fairness testing of machine learning models, in: M. Dumas, D. Pfahl, S. Apel, A. Russo (Eds.), Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE 2019, Tallinn, Estonia, August 26-30, 2019, ACM, 2019, pp. 625–635, http://dx.doi.org/10.1145/3338906.3338937.

[4] J.M. Zhang, M. Harman, L. Ma, Y. Liu, Machine learning testing: Survey, landscapes and horizons, IEEE Trans. Softw. Eng. (2022).

[5] P.J. Phillips, C.A. Hahn, P.C. Fontana, D.A. Broniatowski, M.A. Przybocki, Four principles of explainable artificial intelligence, 2020, Gaithersburg, Maryland.

[6] P.P. Angelov, E.A. Soares, R. Jiang, N.I. Arnold, P.M. Atkinson, Explainable artificial intelligence: An analytical review, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 11 (5) (2021) e1424.

[7] P. Biecek, T. Burzykowski, Explanatory Model Analysis: Explore, Explain and Examine Predictive Models, Chapman and Hall/CRC, 2021.

[8] G. Montavon, W. Samek, K.-R. Müller, Methods for interpreting and understanding deep neural networks, Digit. Signal Process. 73 (2018) 1–15.

[9] H.J. Escalante, S. Escalera, I. Guyon, X. Baró, Y. Güçlütürk, U. Güçlü, M. van Gerven, R. van Lier, Explainable and Interpretable Models in Computer Vision and Machine Learning, Springer, 2018.

[10] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (5) (2018) 1–42.

[11] D. Nigenda, Z. Karnin, M.B. Zafar, R. Ramesha, A. Tan, M. Donini, K. Kenthapadi, Amazon SageMaker model monitor: A system for real-time insights into deployed machine learning models, 2021, arXiv preprint arXiv:2111.13657.

[12] V. Arya, R.K.E. Bellamy, P. Chen, A. Dhurandhar, M. Hind, S.C. Hoffman, S. Houde, Q.V. Liao, R. Luss, A. Mojsilovic, S. Mourad, P. Pedemonte, R. Raghavendra, J.T. Richards, P. Sattigeri, K. Shanmugam, M. Singh, K.R. Varshney, D. Wei, Y. Zhang, One explanation does not fit all: A toolkit and taxonomy of AI explainability techniques, 2019, CoRR, arXiv:1909.03012.

[13] K. Thangavel, A. Pethalakshmi, Dimensionality reduction based on rough set theory: A review, Appl. Soft Comput. 9 (1) (2009) 1–12, http://dx.doi.org/10.1016/j.asoc.2008.05.006.

[14] A. Janusz, D. Ślęzak, KnowledgePit meets BrightBox: A step toward insightful investigation of the results of data science competitions, in: M. Ganzha, L.A. Maciaszek, M. Paprzycki, D. Ślęzak (Eds.), Proceedings of the 17th Conference on Computer Science and Intelligence Systems, FedCSIS 2022, Sofia, Bulgaria, September 4–7, 2022, in: Annals of Computer Science and Information Systems, vol. 30, 2022, pp. 393–398, http://dx.doi.org/10.15439/2022F309.

[15] A. Skowron, D. Ślęzak, Rough sets turn 40: From information systems to intelligent systems, in: M. Ganzha, L.A. Maciaszek, M. Paprzycki, D. Ślęzak (Eds.), Proceedings of the 17th Conference on Computer Science and Intelligence Systems, FedCSIS 2022, Sofia, Bulgaria, September 4–7, 2022, in: Annals of Computer Science and Information Systems, vol. 30, 2022, pp. 23–34, http://dx.doi.org/10.15439/2022F310.

[16] O. Bastani, C. Kim, H. Bastani, Interpreting Blackbox Models via Model Extraction, 2017, CoRR, arXiv:1705.08504.

[17] O. Sagi, L. Rokach, Approximating XGBoost with an interpretable decision tree, Inform. Sci. 572 (2021) http://dx.doi.org/10.1016/j.ins.2021.05.055.

[18] J. Henzel, J. Tobiasz, M. Kozielski, M. Bach, P. Foszner, A. Gruca, M. Kania, J. Mika, A. Papież, A. Werner, J. Żyła, J. Jaroszewicz, J. Polańska, M. Sikora, Screening support system based on patient survey data — Case study on classification of initial, locally collected COVID-19 data, Appl. Sci. 11 (22) (2021) http://dx.doi.org/10.3390/app112210790, URL https://www.mdpi.com/2076-3417/11/22/10790.

[19] M. Sushil, S. Šuster, W. Daelemans, Rule induction for global explanation of trained models, 2018.

[20] E. Pastor, E. Baralis, Explaining black box models by means of local rules, in: C. Hung, G.A. Papadopoulos (Eds.), Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019, ACM, 2019, pp. 510–517, http://dx.doi.org/10.1145/3297280.3297328.

[21] S. Stawicki, D. Ślęzak, A. Janusz, S. Widz, Decision bireducts and decision reducts — A comparison, Internat. J. Approx. Reason. 84 (2017) 75–109, http://dx.doi.org/10.1016/j.ijar.2017.02.007.

[22] E. Debie, K. Shafi, C. Lokan, K. Kasmarik, Reduct based ensemble of learning classifier system for real-valued classification problems, 2013, http://dx.doi.org/10.1109/CIEL.2013.6613142.

[23] Y. Guo, L. Jiao, S. Wang, S. Wang, F. Liu, K. Rong, T. Xiong, A novel dynamic rough subspace based selective ensemble, Pattern Recognit. 48 (5) (2015) 1638–1652, http://dx.doi.org/10.1016/j.patcog.2014.11.001.

[24] D. Ślęzak, M. Grzegorowski, A. Janusz, M. Kozielski, S.H. Nguyen, M. Sikora, S. Stawicki, Ł. Wróbel, A framework for learning and embedding multi-sensor forecasting models into a decision support system: A case study of methane concentration in coal mines, Inform. Sci. 451–452 (2018) 112–133, http://dx.doi.org/10.1016/j.ins.2018.04.026.

[25] A. Janusz, D. Ślęzak, Random probes in computation and assessment of approximate reducts, in: M. Kryszkiewicz, C. Cornelis, D. Ciucci, J. Medina-Moreno, H. Motoda, Z.W. Raś (Eds.), Rough Sets and Intelligent Systems Paradigms — Second International Conference, RSEISP 2014, Held as Part of JRS 2014, Granada and Madrid, Spain, July 9–13, 2014. Proceedings, in: Lecture Notes in Computer Science, vol. 8537, Springer, 2014, pp. 53–64, http://dx.doi.org/10.1007/978-3-319-08729-0_5.

[26] F. Jiang, X. Yu, J. Du, D. Gong, Y. Zhang, Y. Peng, Ensemble learning based on approximate reducts and bootstrap sampling, Inform. Sci. 547 (2021) 797–813, http://dx.doi.org/10.1016/j.ins.2020.08.069.

[27] M.M. Mafarja, S. Mirjalili, Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection, Soft Comput. 23 (15) (2019) 6249–6265, http://dx.doi.org/10.1007/s00500-018-3282-y.

[28] L.S. Riza, A. Janusz, C. Bergmeir, C. Cornelis, F. Herrera, D. Ślęzak, J.M. Benítez, Implementing algorithms of rough set theory and fuzzy rough set theory in the R package ''RoughSets'', Inform. Sci. 287 (2014) 68–89, http://dx.doi.org/10.1016/j.ins.2014.07.029.

[29] J. Dougherty, R. Kohavi, M. Sahami, Supervised and unsupervised discretization of continuous features, in: Machine Learning Proceedings 1995, Elsevier, 1995, pp. 194–202.

[30] D. Ślęzak, A. Chądzyńska-Krasowska, Approximate decision tree induction over approximately engineered data features, in: R. Bello, D. Miao, R. Falcon, M. Nakata, A. Rosete, D. Ciucci (Eds.), Rough Sets – International Joint Conference, IJCRS 2020, Havana, Cuba, June 29–July 3, 2020, Proceedings, in: Lecture Notes in Computer Science, vol. 12179, Springer, 2020, pp. 376–384, http://dx.doi.org/10.1007/978-3-030-52705-1_28.

[31] A. Janusz, D. Kałuża, A. Chądzyńska-Krasowska, B. Konarski, J. Holland, D. Ślęzak, IEEE BigData 2019 cup: Suspicious network event recognition, in: C.K. Baru, J. Huan, L. Khan, X. Hu, R. Ak, Y. Tian, R.S. Barga, C. Zaniolo, K. Lee, Y.F. Ye (Eds.), 2019 IEEE International Conference on Big Data (IEEE BigData), Los Angeles, CA, USA, December 9–12, 2019, IEEE, 2019, pp. 5881–5887, http://dx.doi.org/10.1109/BigData47090.2019.9005668.

[32] Q.H. Vu, D. Ruta, L. Cen, Gradient boosting decision trees for cyber security threats detection based on network events logs, in: C.K. Baru, J. Huan, L. Khan, X. Hu, R. Ak, Y. Tian, R.S. Barga, C. Zaniolo, K. Lee, Y.F. Ye (Eds.), 2019 IEEE International Conference on Big Data (IEEE BigData), Los Angeles, CA, USA, December 9–12, 2019, IEEE, 2019, pp. 5921–5928, http://dx.doi.org/10.1109/BigData47090.2019.9006061.

[33] T. Sainburg, L. McInnes, T.Q. Gentner, Parametric UMAP embeddings for representation and semisupervised learning, Neural Comput. 33 (11) (2021) 2881–2907, http://dx.doi.org/10.1162/neco_a_01434.