Expert Systems With Applications 130 (2019) 97–112

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Empirical evaluation of feature projection algorithms for multi-view text classification

Marcin Michał Mirończuk[a,*], Jarosław Protasiewicz[a], Witold Pedrycz[b]

[a] National Information Processing Institute, Warsaw, Poland
[b] Department of Electrical & Computer Engineering, University of Alberta, Edmonton, Canada

ARTICLE INFO

Article history:
Received 18 May 2018
Revised 9 April 2019
Accepted 9 April 2019
Available online 12 April 2019

Keywords:
Multi-view text classification
Multi-view document classification
Information fusion
Information fusion in text classification
Evaluation method
Ranking method

ABSTRACT

This study aims to propose (i) a multi-view text classification method and (ii) a ranking method that allows for selecting the best information fusion layer among many variations. Multi-view document classification is worth a detailed study as it makes it possible to combine different feature sets into yet another view that further improves text classification. For this purpose, we propose a multi-view framework for text classification that is composed of two levels of information fusion. At the first level, classifiers are constructed using different data views, i.e. different vector space models by various machine learning algorithms. At the second level, the information fusion layer uses input information using a features projection method and a meta-classifier modelled by a selected machine learning algorithm. A final decision based on classification results produced by the models positioned at the first layer is reached. Moreover, we propose a ranking method to assess various configurations of the fusion layer. We use heuristics that utilise statistical properties of F-score values calculated for classification results produced at the fusion layer. The information fusion layer of the classification framework and ranking method has been empirically evaluated. For this purpose, we introduce a use case checking whether companies' domains identify their innovativeness. The results empirically demonstrate that the information fusion layer enhances classification quality. The Friedman's aligned rank and Wilcoxon signed-rank statistical tests and the effect size support this hypothesis. In addition, the Spearman statistical test carried out for the obtained results demonstrated that the assessment made by the proposed ranking method converges to a well-established method named Hellinger - The Technique for Order Preference by Similarity to Ideal Solution (H-TOPSIS). Thus, the proposed approach may be used for the assessment of classifier performance.

1. Introduction

Text classification or categorisation involves construction of models that are able to classify new documents as belonging to one of the previously defined classes (Liu, 2006; Manning, Raghavan, & Schütze, 2008). Currently, it is a sophisticated process involving not only training the models by using a labelled data set but also covering many additional procedures such as data preprocessing, transformation, dimensionality reduction, and various machine learning approaches. Despite the richness and complexity of document categorisation approaches, there are a number of challenging issues that deserve further research. There is an ongoing need to develop new or improve existing classification systems or their functional modules to achieve better results or enhance computational efficiency.

Among the several approaches on how to build a model, we focus on multi-view learning, which manifests as a significant trend of current modelling based on data and has yielded promising results (for details, see Section 2). Multi-view learning is also known as data or information fusion, or data integration coming from multiple feature sets, i.e. multiple feature spaces, diversified feature spaces, or information sources. It is assumed that feature sets have different distributions. Thus, views are perceived as independent, and each has a specific statistical property (Woźniak, Graña, & Corchado, 2014; Xu, Tao, & Xu, 2013; Zhao, Xie, Xu, & Sun, 2017). Multi-view learning aims to construct a single model for each view and then optimally utilise the models to improve the overall generalisation performance (Zhao et al., 2017). An extensive literature review, which is presented in Section 2, reveals that multi-view learning and its evaluation have not been widely discussed in the available works. According to Lim, Lee, and

* Corresponding author.
E-mail addresses: marcin.mironczuk@opi.org.pl (M.M. Mirończuk), jaroslaw.protasiewicz@opi.org.pl (J. Protasiewicz), wpedrycz@ualberta.ca (W. Pedrycz).

https://doi.org/10.1016/j.eswa.2019.04.020
0957-4174/© 2019 National Information Processing Institute. Published by Elsevier Ltd. This is an open access article under the CC BY license.
(http://creativecommons.org/licenses/by/4.0/)

Kim (2005), Ghani, Slattery, and Yang (2001) and Dasigi, Mann, and Protopopescu (2001), it is closely related to a fusion method for document categorisation.

Based on the above observations, we are of the opinion that multi-view document classification is worth a detailed study. It is beneficial to highlight this issue and fill a gap present because of insufficient research reported in this area. Therefore, in this study, we investigate the multi-view approach to document classification and concurrently propose its evaluation method. More specifically, we develop, evaluate, and discuss a multi-view framework for text classification that is based on two levels of information fusion. At the first layer, there are classifiers exploiting data views by engaging various machine learning algorithms. Each model utilises a different data view established on the same data set. At the second level, we encounter an information fusion layer, which is composed of a feature projection method and a meta-classifier. The feature projection method transforms the decisions produced by the models located at the first layer into another vector space model, whereas the meta-classifier (an information fusion processor) forms a final decision based on the new features. The proposed framework has been empirically evaluated to assess its performance using the Friedman's aligned rank and Wilcoxon signed-rank statistical tests, and a quantitative measure of the magnitude of a phenomenon called the effect size (Demsar, 2006; Field, Miles, & Field, 2012; Friedman, 1937; Myles, Douglas, & Eric, 2014). The evaluation is processed on real data from our information system called Inventorum (Protasiewicz, 2017a, 2017b) to resolve the real-world problem of innovative websites recognition (Mironczuk & Protasiewicz, 2016).

Few studies in the literature consider multi-view document classification in terms of supervised learning (Dasigi et al., 2001; Ghani et al., 2001; Lim et al., 2005). These works form a good starting point for exploring the concept of multi-view text classification. On the other hand, they are quite old and do not comprehensively discuss the impact of feature projection techniques on classification results. In light of this, it is worth comparing combinations of machine learning methods regarded as sound fusion mechanisms. Furthermore, we noticed that the multi-view text classification realised in the context of more than two views has not been widely discussed or empirically studied. However, we have to note that there exist all necessary components to create and diversify feature sets of multi-view text classification, so that they could be easily applied.

In addition, we propose a ranking method to assess various configurations of the information fusion layer. It is a heuristic method that utilises statistical properties of F-score values calculated on classification results produced by the fusion layer. We empirically verify the ranking correctness by using the Spearman statistical test (Myles et al., 2014).

Overall, the objectives and contributions of our work are outlined as follows:

1. To propose a classification system composed of multi-view classification models and a fusion layer.
2. To present an extensive experimental study which elaborates on the influence of training algorithms and feature projection methods on the performance of the information fusion layer working on outputs produced by the classifiers.
3. To propose a ranking method that allows for selecting the best information fusion layer among its possible variations.

The study concerns an original information fusion problem. The novelty of this is three-fold. Firstly, we report a comprehensive and critically enhanced literature review, which focuses on text classification issues related to the system proposed in this study. In addition to the better understanding of a categorisation system and its parts, we point out which document classification issues still need improvement. Secondly, we introduce the information fusion layer in the classification framework. This layer is implemented as a meta-classifier working at the second level of information classification. Through a suite of diversified experiments, we show that it can improve final classification results. In addition, we verify experimentally the influence of various information transformation methods and machine learning algorithms of the information fusion layer on its performance. Finally, we empirically demonstrate that the proposed ranking method is appropriate to rank and select optimal implementation of the information fusion layer. The ranking method produces the same results regarding statistical tests as another well-founded ranking process named Hellinger - The Technique for Order Preference by Similarity to Ideal Solution (H-TOPSIS) (Krohling, Lourenzutti, & Campos, 2015; Krohling & Pacheco, 2015) and it works without the need to select values of additional parameters.

In our opinion, the provided study significantly augments knowledge regarding modelling of classification systems. Particularly valuable are the findings concerning information fusion and its assessment methods. We have to underline that they are elaborated on through experimentation. Moreover, we show that it is possible to combine different sets of features in a meta-view to achieve better results of the text classification.

This paper is structured as follows. Section 2 covers a comprehensive overview of related works. Next, Section 3 describes the proposed classification framework with special emphasis on the information fusion layer. In Section 4, we discuss the ranking method that is the proposed heuristic approach aimed at the assessment of the information fusion layer. Next, Section 5 introduces a use case for empirical methods verification. Section 6 describes the evaluation process of the framework and the results obtained during experiments. Finally, Section 7 presents some concluding comments.

2. Related work

Text document classification (text classification) is a problem related to assigning predefined classes (labels) to an unlabelled text document. Numerous studies have focussed on describing various approaches and applications to text classification. We enumerate several studies that describe a classification task in a given domain. For example, there is research completed in industry (Ittoo, Nguyen, & van den Bosch, 2016; Lin, 2009), finance (de Fortuny, Smedt, Martens, & Daelemans, 2014; Kumar & Ravi, 2016), medicine (Mostafa & Lam, 2000; Parlak & Uysal, 2015; Shen et al., 2016), the Internet, such as analysis of email, server logs, web pages, websites, tweets, etc. (Basto-Fernandes et al., 2016; Chang & Poon, 2009; Cuzzola, Jovanović, Bagheri, & Gašević, 2015; Kan & Thi, 2005; Qi & Davison, 2009), patent databases (Giachanou, Salampasis, & Paltoglou, 2015; Li & Shawe-Taylor, 2007; Trappey, Hsu, Trappey, & Lin, 2006), and alike.

Multi-view learning is also referred to as data fusion, or data integration from multiple feature sets, multiple feature spaces, or diversified feature spaces that may have different distributions of features, i.e. data views are conditionally independent sets of features, and each view exhibits a specific statistical property (Xu et al., 2013; Zhao et al., 2017). Such learning aims to learn a function modelling a particular view and jointly optimising all functions to improve generalisation performance (Zhao et al., 2017). It seems that the multi-view approach is related to an ensemble learning method.

Ensemble learning may utilise techniques that create data views. They are methods that horizontally or vertically partition a dataset and use it to build a classifier committee of multi-view ensemble learning or stacked generalisation (Kumar & Minz, 2016; Sammut & Webb, 2010b; Sun, 2013; Zhao et al., 2017). Usually, these partitioning methods are used when no multiple natural views are available, so we need to construct multiple views from a single view. Furthermore, regarding ensemble stacked generalisation (Sammut & Webb, 2010b), a set of models is constructed from bootstrap samples of a data set and then their outputs from a hold-out data set are used as input to a "meta"-model. The set of base models is called a level-0, and the meta-model is a level-1. The task of the level-1 model is to combine the set of outputs to correctly classify a target, thereby correcting any mistakes made by the level-0 models. In this case, the bootstrap samples of the data set can be considered as views and the meta-model can be considered as a joint optimiser. In the multi-view learning on text classification field, for instance, web page classification is a popular theme. This case uses a co-training schema based on two views, such as the text content of the web page and anchor text of any web page linking to this web page.

There are a limited number of works that explain the multi-view approach to text classification in comparison to the rich literature describing various aspects of document classification. In multi-view learning in the text classification field, for instance, web page classification is widely discussed. This case uses a co-training schema based on two views, such as the text of a web page and anchor text of any web page linking to this web page (Blum & Mitchell, 1998). Also, the most similar works in this field focus on semi-supervised learning that utilises a co-training method to resolve a given multi-view classification task. For example, Matsubara, Monard, and Batista (2005) proposed a simple approach to textual document pre-processing to easily construct two different views required by any multi-view learning algorithm. Gu, Zhu, and Zhang (2009) developed a new co-features active semi-supervised learning algorithm. They utilised three types of data view, such as a lexical view, a semantic view, and a syntactic view in their document classification solution. Hajmohammadi, Ibrahim, and Selamat (2014) developed a multi-view semi-supervised learning system, which resolves a cross-lingual sentiment classification problem. They utilised different document translations as views and based on these created a classification framework to conduct semantic classification.

To the best of our knowledge, there are only three studies that relate closely to multi-view document classification using supervised learning (Dasigi et al., 2001; Ghani et al., 2001; Lim et al., 2005). Lim et al. (2005) presented a system to classify genres of a web document using multiple, different feature sets. They proposed feature sets extracted from URLs and HTML tags. In addition, they utilised token information, lexical information and structural information to build appropriate sets. Unfortunately, they only studied how the different combinations of connected feature sets may impact the performance of classification results. They combined one view from the different views, i.e. the set of features, and they constructed and evaluated the classifier function that was based on this single view. Ghani et al. (2001) conducted similar research as Lim et al. (2005) but they set as a classification goal the recognition of company websites and university web pages. They established different feature sets and tested each feature set to evaluate which produced the best results. At the end, they mentioned briefly a conducted experiment that elaborated a voting ensemble method based on two feature sets which improved classification performance. Finally, Dasigi et al. (2001) proposed strict fusion learning for supervised text classification. They used the Reuters corpus and split them into three separate document subsets. In each subset, they used a feature set extractor to receive four different feature sets that were based on words and noun-phrases. As a fusion learning method, they considered a neural network whose inputs were pre-processed by Latent Semantic Indexing (LSI) methods.

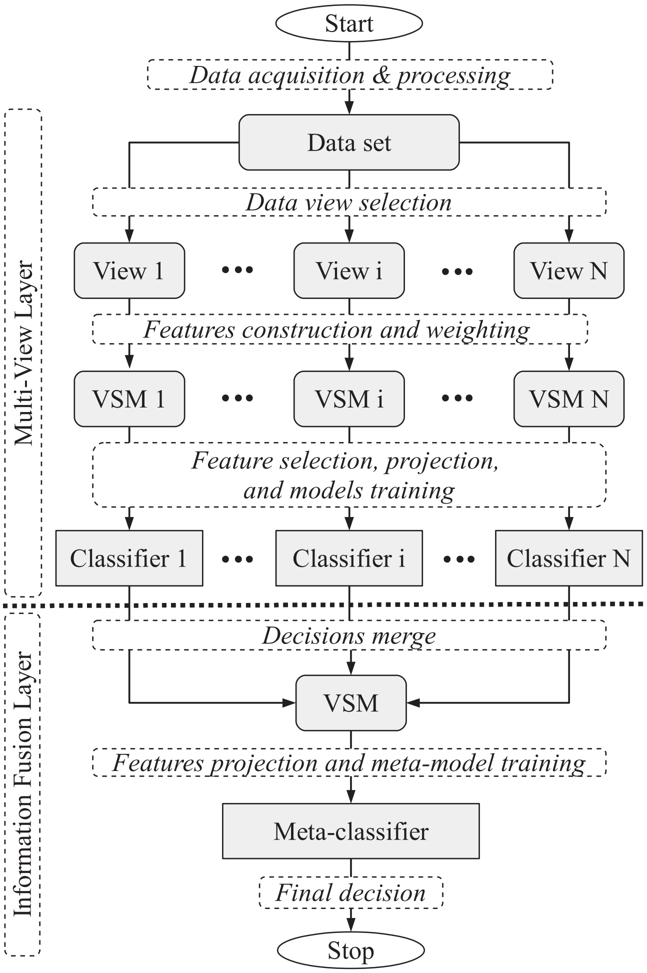

Fig. 1. Multi-view text classification framework.

3. Multi-view text classification framework

The multi-view text classification framework is composed of (i) a multi-view layer, and (ii) an information fusion layer. They process and categorise data more precisely in each succeeding tier. Fig. 1 displays the overall architecture of the framework.

The process begins with data acquisition, where the data could be collected from various sources by web crawlers or transferred from open databases. Having collected all necessary data, it is possible to verify a hypothesis that tests what type of data views (data parts) might produce the best classification model. Each data view represents a different look at the data, highlighting their peculiar properties. The process of data view selection covers two aspects, namely: (i) choosing view types, i.e. deciding which features should be included in each view and (ii) determining the number of views. The selection of view types may be based on meta-information received from data sources: heuristic assumptions or hypotheses; prior knowledge; intuition; experience gained from experiments, etc. The number of views may be determined algorithmically or through manual data analysis.

The data views are pre-processed separately to create an independent vector space model (VSM) for each view. It is a non-trivial process consisting of the following phases. Firstly, features are constructed from data located in the views. Then, they are weighted according to a selected algorithm for features representation. As a result, we develop a number of VSMs. The number of features in the models could be reduced by features selection procedures, and/ or can be projected into lower dimensional spaces. Finally, we achieve optimal VSMs representing data views.

The classification models are trained by using the VSMs and some other selected algorithms. Note that sometimes a training algorithm requires a particular VSM type. When models are properly prepared, they classify incoming data, which have to be represented in the same manner as in the training phase. As a result, each classifier produces decisions regarding whether input vectors belong to each possible class. Technically, the decisions are expressed as probabilities or weights. The above phases constitute the multi-view layer of the framework.

The outcomes produced by the classifiers are considered as new features, and are aggregated together into a new VSM (meta-VSM). The new VSM may be projected into another feature space characterised by a lower or the same dimensionality as the one used before. The meta-VSM is a data set for training a meta-classifier. In the working mode, the meta-classifier produces the final decision, i.e. assigns a class to some given input data. The meta-classifier fuses information coming for classifiers; thus, together with the features emerging and projection processes, we call their combination the information fusion layer.

4. Ranking methods

In this section, we propose a ranking method aimed to assess various combinations of the information fusion layer. Section 4.1 briefly summarises common indicators of the classification models. Section 4.2 shows the first ranking method utilising the statistical properties of indicators; whereas Section 4.3 shortly describes the second method called H-TOPSIS.

4.1. Common indicators and their statistical properties

We propose an evaluation procedure of the framework performance that is based on the investigation of statistical properties of $F-score$ values obtained for a given range of the numbers of the feature. More precisely, we assess experimental models based on their ranking, which utilises the statistical properties of $F-score$ values, such as mean µ and standard deviation s.

Each $F-score$ is obtained by using a 10-fold cross-validation procedure, and it is calculated as the indicator on a *micro* level (Sammut & Webb, 2010a; Wong, 2015). We note that the $F-score$ is a harmonic mean of *precision* and *recall* (Forman & Scholz, 2010; Manning et al., 2008; Sokolova & Lapalme, 2009).

4.2. Proposed ranking method

4.2.1. Experimental models

Assume that there is an experimental model m_i, which belongs to a set of experimental models M, i.e. $M = \{m_1, \cdots, m_k\}$, and k is the total number of models. Each model m_i represents a unique configuration of a machine learning algorithm regarded as a meta-classifier.

4.2.2. Ranking generation procedure

Ranking allows assessing various combinations of information fusion layers. Prior to its calculation, we establish, for the multi-view layer classifiers, a vector $\mathbb{v}_{FC} = (f_{C_1}, \cdots, f_{C_j}, \cdots, f_{C_N})$ containing feature counts $f_{C_j} \in \mathbb{N}^+$, where N is the number of features, and $f_{C_1} < \cdots < f_{C_N}$. In these models, the number of features is set by a user. In our case, we generated a sequence of numbers of features varying from 600 to 12,000 with a step of 100 features to observe how the F-score is affected. The ranking generation consists of the following steps:

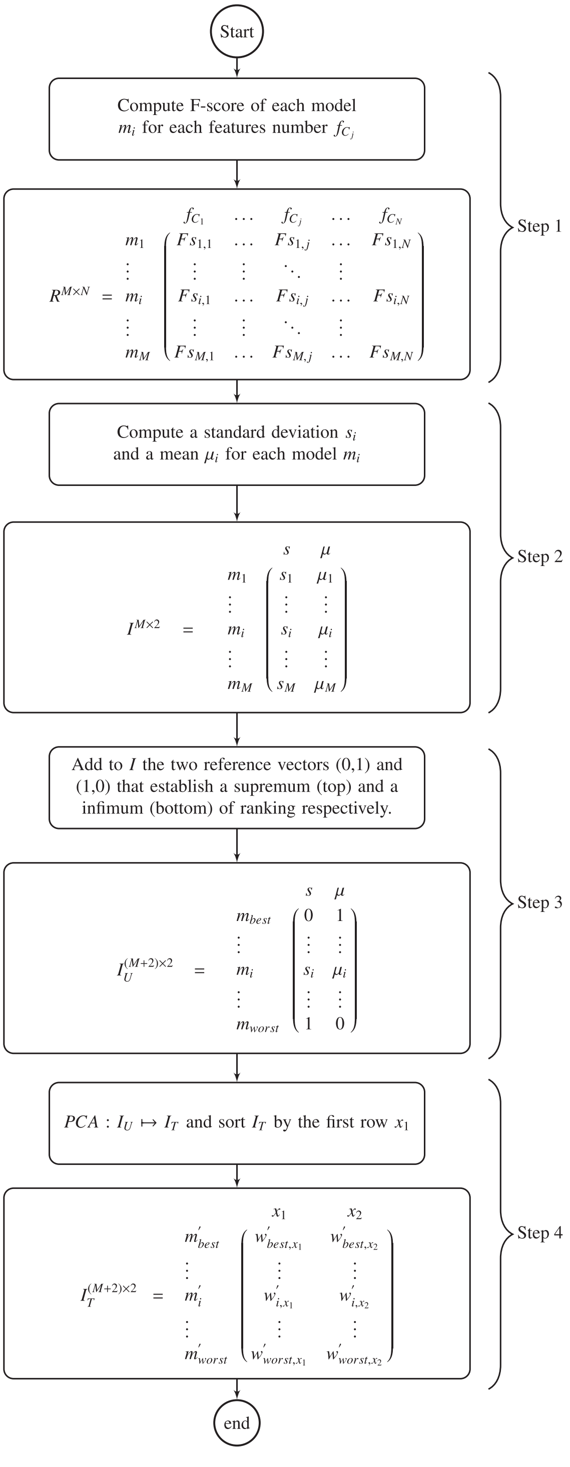
Start
Compute F-score of each model m_i for each features number f_{C_j}
$R^{M\times N} = \begin{array}{c|ccccc} & f_{C_1} & \dots & f_{C_j} & \dots & f_{C_N} \\ m_1 & Fs_{1,1} & \dots & Fs_{1,j} & \dots & Fs_{1,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_i & Fs_{i,1} & \dots & Fs_{i,j} & \dots & Fs_{i,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_M & Fs_{M,1} & \dots & Fs_{M,j} & \dots & Fs_{M,N} \end{array}$
Step 1
Compute a standard deviation s_i and a mean μ_i for each model m_i
$I^{M\times 2} = \begin{array}{c|cc} & s & \mu \\ m_1 & s_1 & \mu_1 \\ \vdots & \vdots & \vdots \\ m_i & s_i & \mu_i \\ \vdots & \vdots & \vdots \\ m_M & s_M & \mu_M \end{array}$
Step 2
Add to I the two reference vectors (0,1) and (1,0) that establish a supremum (top) and a infimum (bottom) of ranking respectively.
$I_U^{(M+2)\times 2} = \begin{array}{c|cc} & s & \mu \\ m_{best} & 0 & 1 \\ \vdots & \vdots & \vdots \\ m_i & s_i & \mu_i \\ \vdots & \vdots & \vdots \\ m_{worst} & 1 & 0 \end{array}$
Step 3
$PCA : I_U \mapsto I_T$ and sort I_T by the first row x_1
$I_T^{(M+2)\times 2} = \begin{array}{c|cc} & x_1 & x_2 \\ m'_{best} & w'_{best,x_1} & w'_{best,x_2} \\ \vdots & \vdots & \vdots \\ m'_i & w'_{i,x_1} & w'_{i,x_2} \\ \vdots & \vdots & \vdots \\ m'_{worst} & w'_{worst,x_1} & w'_{worst,x_2} \end{array}$
Step 4
end

Fig. 2. Overall view of the ranking procedure.

1. Calculate F-score $F_{S_{i,j}}$ for each feature number $f_{C_j} \in N$ and for each experimental model $m_i \in M$. As a result, a matrix $R^{M\times N}$ of F-score values is produced (see Step 1 in Fig. 2).
2. Calculate a standard deviation s_i and mean μ_i of each row of the matrix R. Note that each row contains all F-scores of a single experimental model m_i. As a result, a matrix $I^{M\times 2}$ is formed (see Step 2 in Fig. 2)
3. Extend matrix I by a vector (0, 1) representing the best possible model, for which the standard deviation of F-scores equals 0 and the mean equals 1, and by a vector (1, 0) representing the

worst possible model, where the standard deviation of F-scores is equal to 1 and the mean is equal to 0. As a result, we form a matrix $I_U^{(M+2)\times 2}$ (see Step 3 in Fig. 2). We remark that vectors (0, 1) and (1, 0) represent the best and the worst results possible, respectively. In other words, they are the baselines marking the top and bottom points of the ranking. The results produced by the ranking methods may only be positioned between them.

4. Apply the Principal Components Analysis (PCA) algorithm to the matrix I_U achieving a new matrix $I_T^{(M+2)\times 2}$, which has the same dimensionality as the original matrix but includes transformed values. Then, the rows of the new matrix are sorted in descending order according to the first column (x_1) containing transformed values (w'_{i,x_1}) of the vector m_i. As a result, each row of the matrix I_U represents one experimental model. The models are ordered in descending order proceeding from the best to the worst model (see Step 4 shown in Fig. 2).

In this way, we achieve the ranking of all experimental models (Fig. 2). The best results of the ranking are visualised for the preliminary assessment of the models' performance. A more detailed analysis is performed by using (1) the Friedman's aligned rank and Wilcoxon signed-rank statistical tests for our repeated-measures design (dependent design) and (2) the effect size (Demsar, 2006; Field et al., 2012; Friedman, 1937; Myles et al., 2014) to check whether the differences are statistically meaningful.

4.2.3. Complexity of the proposed method

The proposed ranking method has the complexity of the PCA algorithm if we exclude pre-processing steps 1–3 from its estimation. Ge and Song (2012) and Du and Fowler (2008) estimated the complexity of PCA as $O(MN2)$, where M is the number of examples and N is the dimension of each example. The first step is the most time-consuming. At that stage, the matrix of the measurement values is constructed, and the complexity depends on the number of features, types of models used and their number. Steps 2–3 include simple computations of the mean with standard deviation and modification of the matrix, and have limited influence on the complexity.

4.3. H-TOPSIS method

The H-TOPSIS multi-criteria evaluation method (Krohling et al., 2015; Krohling & Pacheco, 2015) is an extension of a technique for determining order of preference by similarity to an ideal solution. It was developed by Hwang and Yoon (1981). In general, H-TOPSIS is one of the methods of multi-criteria decision making (MCDM) that is widely used to select a finite number of alternatives characterised by multiple conflicting criteria (attributes) (Kou, Lu, Peng, & Shi, 2012; Krohling et al., 2015; Zavadskas, Zakarevicius, & Antucheviciene, 2006). Krohling et al. (2015) used the H-TOPSIS method to rank evolutionary algorithms based on the mean and the standard deviation. Therefore, in this sense, our proposition is similar to their solution, where the Spearman statistical test showed that the proposed ranking method converges to H-TOPSIS. Furthermore, their method is based on the following facets: (1) Central Limit Theorem (CLT), (2) the Hellinger distance between two Gaussian distributions, and (3) the positive ideal solutions (benefits) and the negative ideal solutions (costs). The benefit criterion implies that a higher value of an indicator, e.g., a mean, is better. Conversely, the cost criterion assumes that a lower value of an indicator, e.g., a standard deviation, is better for a model (Krohling et al., 2015). In the context of algorithms (methods, models) comparison, alternatives consist of multiple algorithms, and the criteria are benchmarks, i.e. pairs of the mean and standard deviation for each benchmark. The proposed procedure to create a ranking does not make any statistical assumptions and does not utilise any information about the weights of benefits and costs.

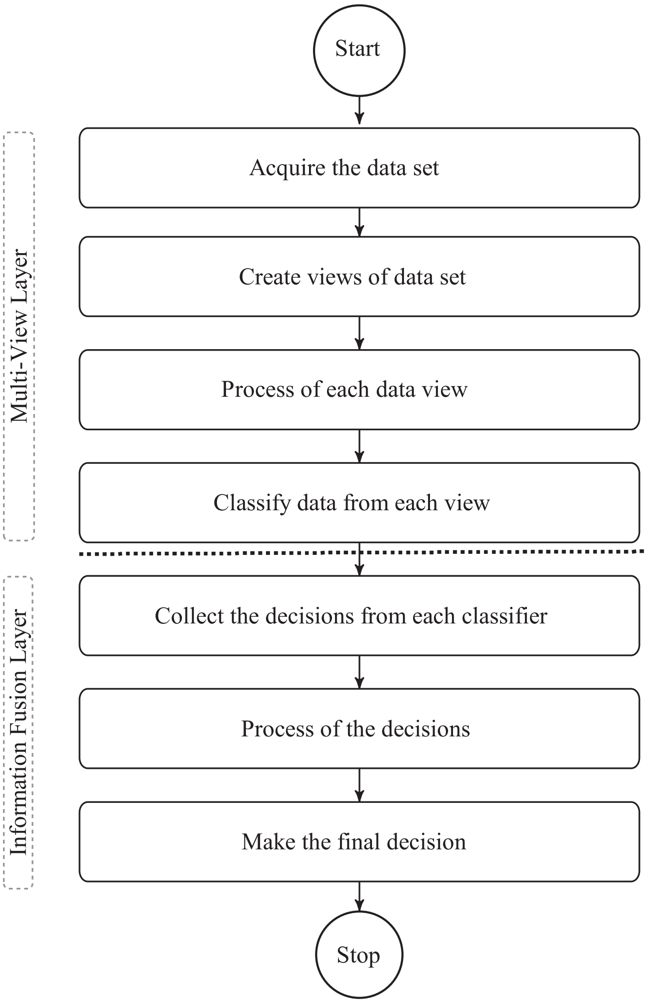

Fig. 3. Use case implementing the framework for multi-view text classification.

5. Use case

In this section we introduce a use case depicting how the framework works.

5.1. Brief overview

We use a detection problem of innovative websites on the Internet as a real use case implementing the proposed framework (Fig. 1). In this task, we classify companies, based on their websites, as belonging to one of two classes, namely $c_1 = innovative\ company\ website$ or $c_2 = non-innovative\ company\ website$. The use case realises the framework following a series of steps as presented in Fig. 3.

Fig. 3 presents the realisation of the multi-view framework for text classification based on the two levels of information fusion. At the first layer, i.e. the multi-view level (1) data are acquired, (2) different views (the VSMs) are created, (3) feature selection methods are used to carry out dimensionality reduction, and finally (4) classifiers are modelled based on the data views using various machine learning algorithms. Each model utilises a different data view established for the same data set. The second level, i.e. the information fusion layer, is composed of (1) a decisions collector, (2) a feature projection method, and (3) a meta-classifier. The feature projection method transforms the decisions produced by models located at the first layer and these are collected into another vector space model. The meta-classifier (the information fusion processor) forms a final decision based on the new features.

5.2. Practical implementation of the framework

The detailed overview of framework implementation includes the seven following steps (Fig. 3):

Step 1. The process begins with data acquisition that is realised by a *Acquire the data set* element. A focussed crawler collects a set S of Internet websites that describe companies. Each website $s \in S$ is a collection P_s of HTML documents $p \in P_s$.

Step 2. Then, the data views are created in an element *Create views of data set*. Although data views may be created algorithmically, for example by utilising ensemble learning techniques (Kumar & Minz, 2016; Sammut & Webb, 2010b; Sun, 2013; Zhao et al., 2017), we decided to rely on our previous research (Mironczuk & Protasiewicz, 2016), in which the number of data views and their features are determined through detailed data analysis and pre-experiments. As we have observed that each website could be described by three separate views, a data set A is composed of three data views, namely: (i) label links $L \in A$, (ii) company description $D \in A$, and (iii) big document $B \in A$, which is a combination (concatenation) of selected web pages. Technically, each website $s \in S$ is processed to achieve:

- the data view L containing link labels, which we denote as $p_{L,1}$: $P_s \mapsto d_L$;
- the data view D containing a company description, which we denote as $p_{D,1}$: $P_s \mapsto d_D$;
- the data view B containing big document, which we denote as $p_{B,1}$: $P_s \mapsto d_B$;

where d_L, d_D, and d_B are the extracted text data, respectively.

Step 3. Next, a *Process of each data view* component prepares data by representing them as vector space models, it selects features, and then it trains a set of classification functions (classification models). More specifically, this is realised as follows:

- The data from views d_L, d_D, and d_B are transformed into three vector space models representing:
  - a set $\mathbb{V}_L$ of vectors $\mathbb{v}_L \in \mathbb{V}_L$ representing the view of link labels $p_{L,2} : d_L \mapsto \mathbb{v}_L$;
  - a set $\mathbb{V}_D$ of vectors $\mathbb{v}_D \in \mathbb{V}_D$ representing the view of a company description $p_{D,2} : d_D \mapsto \mathbb{v}_D$;
  - a set $\mathbb{V}_B$ of vectors $\mathbb{v}_B \in \mathbb{V}_B$ representing the view of big document $p_{B,2} : d_B \mapsto \mathbb{v}_B$.
- The vector space models $\mathbb{V}_L$, $\mathbb{V}_D$, and $\mathbb{V}_B$ are analysed to remove non-useful features. As a result, we obtain modified vector space models, i.e.:
  - $\mathbb{V}_L^{'}$ by using the transformation $\forall_{\mathbb{v}_L \in \mathbb{V}_L} f_P : \mathbb{v}_L \mapsto \mathbb{v}_L^{'}$, where $\mathbb{v}_L^{'} \in \mathbb{V}_L^{'}$ and $|\mathbb{v}_L^{'}| \ll |\mathbb{v}_L|$;
  - $\mathbb{V}_D^{'}$ by using the transformation $\forall_{\mathbb{v}_D \in \mathbb{V}_D} f_P : \mathbb{v}_D \mapsto \mathbb{v}_D^{'}$, where $\mathbb{v}_D^{'} \in \mathbb{V}_D^{'}$ and $|\mathbb{v}_D^{'}| \ll |\mathbb{v}_D|$;
  - $\mathbb{V}_B^{'}$ by using the transformation $\forall_{\mathbb{v}_B \in \mathbb{V}_B} f_P : \mathbb{v}_B \mapsto \mathbb{v}_B^{'}$, where $\mathbb{v}_B^{'} \in \mathbb{V}_B^{'}$ and $|\mathbb{v}_B^{'}| \ll |\mathbb{v}_B|$.
- The modified vector space models are utilised to train classification functions γ giving:
  - a classification model based on the view of link labels $\Gamma_L : \mathbb{V}_L^{'} \mapsto \gamma_L$;
  - a classification model based on the view of a company description $\Gamma_D : \mathbb{V}_D^{'} \mapsto \gamma_D$;
  - a classification model based on the view of big document $\Gamma_B : \mathbb{V}_B^{'} \mapsto \gamma_B$.

Step 4. Next, the classification models categorise data originating from each data view, using a component *Classify data from each view*. Technically, the functions γ produce decisions, i.e. they assign weights $w \in R$ to a class c_1 or c_2. As a result, we obtain weights vectors, namely:

- $r_L = (w_{c_1,r_L}, w_{c_2,r_L})$ by classifying $\forall_{\mathbb{v}_L^{'} \in \mathbb{V}_L^{'}} \gamma_L : \mathbb{v}_L^{'} \mapsto r_L$ of the label links view;
- $r_D = (w_{c_1,r_D}, w_{c_2,r_D})$ by classifying $\forall_{\mathbb{v}_D^{'} \in \mathbb{V}_D^{'}} \gamma_D : \mathbb{v}_D^{'} \mapsto r_D$ of the company description view;
- $r_B = (w_{c_1,r_B}, w_{c_2,r_B})$ by classifying $\forall_{\mathbb{v}_B^{'} \in \mathbb{V}_B^{'}} \gamma_B : \mathbb{v}_B^{'} \mapsto r_B$ of the big document view.

Step 5. Next, the decisions are collected together, where $p_M : (r_L, r_D, r_B) \mapsto \mathbb{v}_M$ in a *Collect the decisions from each classifier* component, resulting in a set $\mathbb{V}_M$ of meta-vectors $\mathbb{v}_M = (w_{c_1,r_L}, w_{c_1,r_D}, w_{c_1,r_B})$, $\mathbb{v}_M \in \mathbb{V}_M$

Step 6. Then, the meta-vectors set is forwarded to the *Process of the decisions* component, which transforms features $\forall_{\mathbb{v}_M \in \mathbb{V}_M} p_P : \mathbb{v}_M \mapsto \mathbb{v}_M^{'}$ into a new feature space $\mathbb{v}_M^{'}$ characterised by lower dimensionality, where $\mathbb{v}_M^{'} \in \mathbb{V}_M^{'}$ and $\mathbb{V}_M^{'}$ is a set of transformed meta-vectors.

Step 7. Finally, a classification function γ_M is modelled in a learning process Γ_M that is based on the set of meta-vectors $\mathbb{V}_M$. When the meta-classifier is ready, the *Make the final decision* component assigns a class label to the input data, i.e. $\forall_{\mathbb{v}_M^{'} \in \mathbb{V}_M^{'}} \gamma_M : \mathbb{v}_M^{'} \mapsto r_F$, where $r_F = c_1$ *or* c_2.

5.3. Discussion of the use case

It is noteworthy that in our previous research (Mironczuk & Protasiewicz, 2016), we tested the hypothesis that a meta-classifier γ_M could produce better results than single classifiers. The meta-classifier was trained on the outputs for classifiers modelling various data views. The hypothesis was correct, which was empirically validated with the following set-up:

1. The label links view
   (a) binary representation of $\mathbb{v}_L$ vectors,
   (b) a Naive Bayes classifier based on a Bernoulli distribution model γ_L,
   (c) a *Fisher* feature selection method f_s;
2. The company description view
   (a) binary representation of $\mathbb{v}_D$ vectors,
   (b) a Naive Bayes classifier based on a Bernoulli distribution model γ_D,
   (c) a χ^2 feature selection method f_s;
3. The big document view
   (a) term-frequency representation of the $\mathbb{v}_B$ vectors,
   (b) a Naive Bayes classifier based on a multinomial distribution model γ_B,
   (c) a *Fisher* feature selection method f_s.

In this study, we extend the assumptions and experiments by introducing the information fusion layer with feature projection methods (see Figs. 1 and 3) and analyse its influence on the results produced by the meta-classifier γ_M in the *Take the final decision* component shown in Fig. 3.

6. Empirical evaluation

In this section, we present and discuss the evaluation results of the framework, namely Section 6.1 explains briefly the experimental procedure and Section 6.2 covers the detailed results and

Table 1
Ranking of the information fusion layers including a meta-classifier γ_M without meta-vectors transformation, i.e. $\mathbb{v}'_M = \mathbb{v}_M$ and $|\mathbb{v}'_M| = 3$.

Metaclassifier	Configuration	s	μ	Our ranking	H-TOPSIS ranking
SVM	kernel = radial	0.0472	0.8337	1	1
DT	cp = 0.01 minsplit = 2	0.0474	0.833	2	2
k-NN	k = 9	0.0516	0.8312	3	3
k-NN	k = 3	0.0538	0.8267	4	4
NB	laplace = 1	0.0397	0.7897	5	6
k-NN	k = 1	0.05758	0.8	6	5

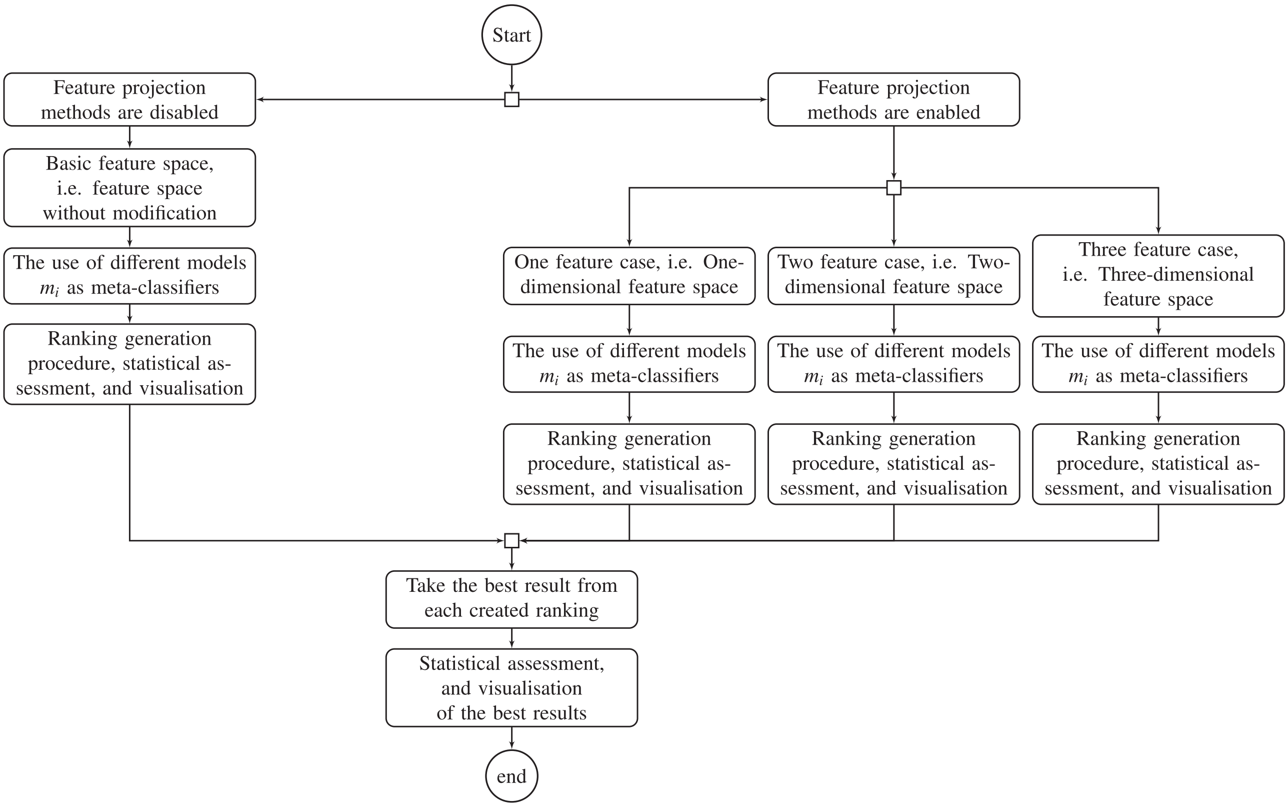
Start
Feature projection methods are disabled
Feature projection methods are enabled
Basic feature space, i.e. feature space without modification
The use of different models m_i as meta-classifiers
Ranking generation procedure, statistical assessment, and visualisation
One feature case, i.e. One-dimensional feature space
Two feature case, i.e. Two-dimensional feature space
Three feature case, i.e. Three-dimensional feature space
The use of different models m_i as meta-classifiers
The use of different models m_i as meta-classifiers
The use of different models m_i as meta-classifiers
Ranking generation procedure, statistical assessment, and visualisation
Ranking generation procedure, statistical assessment, and visualisation
Ranking generation procedure, statistical assessment, and visualisation
Take the best result from each created ranking
Statistical assessment, and visualisation of the best results
end

Fig. 4. Flow of experiments assessing the performance of information fusion layers.

discussion. In addition, Appendix A includes another empirical test which compares the H-TOPSIS and our ranking method.

6.1. Experimental study

6.1.1. Experiment plan and setups

We have planned and conducted the empirical experiments to ensure that there are further opportunities:

- To improve classification results thanks by using a certain configuration of feature projection with the experimental model/meta-classifier, implemented at the fusion layer.
- To rank the experimental models using our ranking method.

The empirical experiments and acquired data helped us to provide answers to the following detailed research questions:

- Can we create a new feature space with the same, higher or lower dimensionality that improves or gives the same results as the unmodified feature space?
- Which combination of feature projection technique and meta-classifier leads to the best results?
- How can we choose the best combination of techniques from their many possibilities?
- Can we use our ranking method to rank the experimental models?

We created a set of experiments that are composed of several independent experiments to deliver empirical data to answer the above questions. The acquired data were evaluated using statistical tests, and a comparative analysis is reported. Fig. 4 outlines the proposed plan.

We test several experimental models m_i as meta-classifiers γ_M by applying or not the feature projection methods in the fusion layer as depicted in Fig. 4. We test one, two, or three dimensional feature spaces when enabling the features projection. The tests are carried out according to the procedure described in Section 4.2. As a result, we obtain a ranking of the applied models m_i for each feature space.

We constructed two rankings. The first was created by using the proposed method. The second was built using the H-TOPSIS multi-criteria evaluation method. Then, we applied the Spearman's rank method r_s to compare both rankings (Myles et al., 2014). The method helps to check whether there is no monotonic association

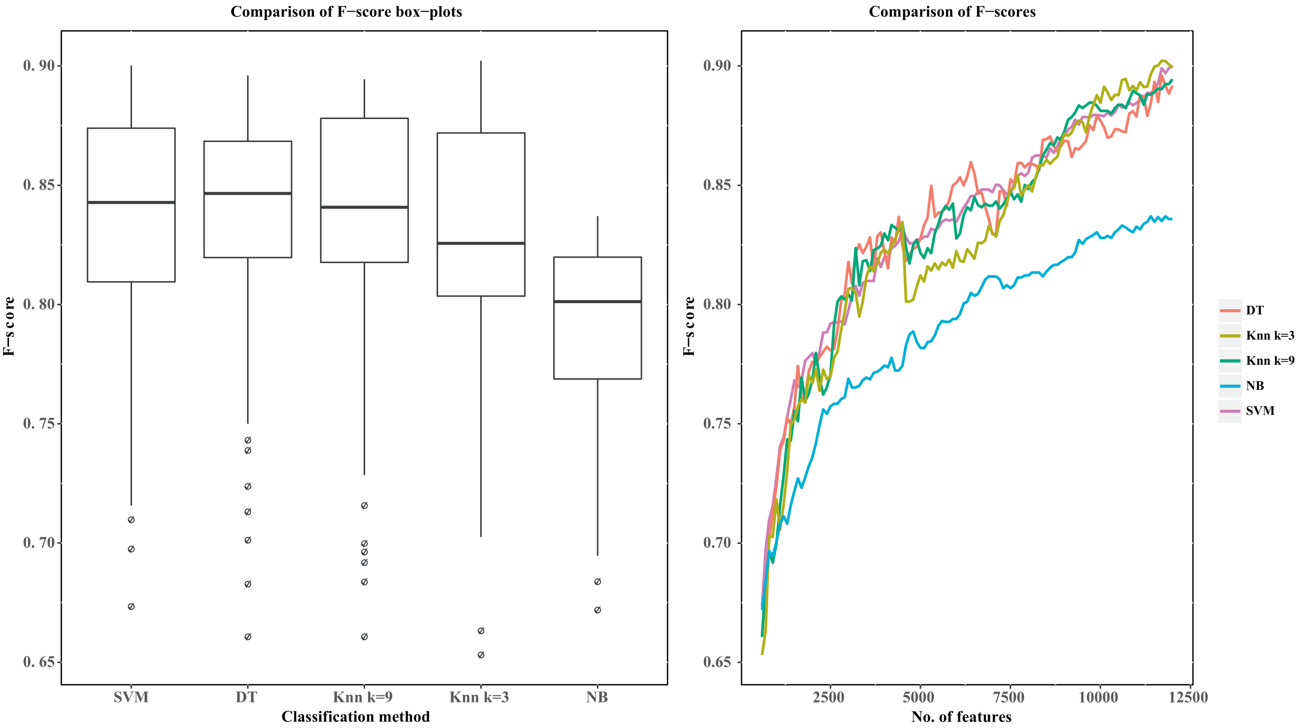

Fig. 5. Box-plot of the *F*-score median values (on the left) and the *F*-score plot (on the right) of the top five information fusion layers originating from the rankings presented in Table 1.

between our rank and the H-TOPSIS rank, which forms the null hypothesis H_{S0}. For this test, we assume a statistical significance of $\alpha = 0.05$. We note that Ali, Lee, and Chung (2017) applied this method to the similarity comparison of rankings given by different multi-criteria methods.

The second type of statistical test was applied to verify whether the $F - score$ distributions of different information fusion layers are identical without assuming that they follow the normal distribution. For this, we take the best results generated by the top-ranked experimental models (the first position of each created ranking) for each feature space. Firstly, we used the Friedman's aligned rank to indicate that we can safely reject the null hypothesis H_{F0} that the experimental models perform the same (Demsar, 2006; Friedman, 1937). It is assumed that the statistical significance α is equal to 0.05. Secondly, the results were visualised and the experimental models were compared using the Wilcoxon signed-rank statistic test using the Bonferroni correction of *p*-values for multiple testing (corrected pairwise tests) (Myles et al., 2014; R Core Team, 2016; Salkind, 2010). The null hypothesis H_{W0} of the Wilcoxon signed-rank test is that the median difference between pairs of experimental models is zero and α is equal to 0.05. Also, we computed the effect size (r) which measures the size of an effect during experimental manipulation. It is a standardised, simple and objective measure of the magnitude of the observed effect (Bordens & Abbott, 2017; Field et al., 2012). We can distinguish three levels of effect, such as low effect (r =0.10), medium effect (r =0.30), and high effect (r =0.50) (Cohen, 1988, 1992; Field et al., 2012).

6.1.2. Data set

The experimental data set originates from our previous work (Mironczuk & Protasiewicz, 2016), where we created our original test data set owing to the lack of such data in other works. It is composed of 2747 real websites, where 509 are labelled as innovative websites and 2238 are labelled as non-innovative ones, i.e. they describe innovative or non-innovative companies. From these data, we created three data views, each containing 2747 examples, as follows: (1) the link labels view including 140,271 features, (2) the company description view covering 140,699 features, and (3) the big document view consisting of 663,015 features.

6.1.3. Implementation

We use the R-project statistics package (R Core Team, 2016) to construct the information fusion layer from the feature projection p_p and machine training γ_M processes. For this purpose, we utilise various training algorithms, such as: Decision Tree (DT) (Therneau, Atkinson, & Ripley, 2015), k-Nearest Neighbour (k-NN), Naive Bayes (NB), and Support Vector Machine (SVM) (Meyer, Dimitriadou, Hornik, Weingessel, & Leisch, 2015) to train the meta-classifiers. This selection is based on the benchmark of classification algorithms provided by Zhang, Liu, Zhang, and Almpanidis (2017) and our personal experience. In this study, the authors validated 11 state-of-the-art algorithms on 71 publicly available datasets, including deep learning approaches. According to our findings, the most efficient algorithms for classifications tasks are Gradient Boosted Decision Tree (GBDT), SVM, and Random Forest (RF). However, based on our experience, algorithm popularity, and technical limitations, we decided to include k-NN, NB, and DT without boosted gradient, and exclude RF.

Concurrently, we use the following feature transformation algorithms: Latent Discriminant Analysis, Sammon projection that is one form of non-metric multidimensional scaling (Sammon) (Venables & Ripley, 2002), Principal Component Analysis (PCA) (R Core Team, 2016), non-linear PCA (Karatzoglou, Smola, Hornik, & Zeileis, 2004), and t-Distributed Stochastic Neighbour Embedding (tSNE) (Krijthe, 2015). Moreover, the Hmisc R-project package (Harrell Jr, 2016) is used to compute the Spearman's ranks and $p - values$ of the statistical significance tests. The Scmamp R-project package (Calvo & Santafe, 2015) is used to conduct the Friedman's aligned rank tests and the standard Stat package (R Core Team, 2016) is used to perform the Wilcoxon signed-rank statistical tests with Bonferroni correction.

Table 2
The p-values of the Wilcoxon–Mann–Whitney test of the top information fusion layers originating from ranking presented in Table 1.

Layers pair	p-value	Are similar results? $(p - value \geq \alpha,\ \alpha = 0.05)$	Effect size (r)	Effect size label
SVM vs DT	1	Yes	−0.074	Small
SVM vs k-NN (k = 9)	0.32806	Yes	−0.151	Small
SVM vs k-NN (k = 3)	2.1e−06	No	−0.347	Medium
SVM vs NB	< 2e−16	No	−0.614	Large
SVM vs k-NN (k = 1)	< 2e−16	No	−0.614	Large
DT vs k-NN (k = 9)	1	Yes	−0.094	Small
DT vs k-NN (k = 3)	0.00077	No	−0.267	Small
DT vs NB	< 2e−16	No	−0.613	Large
DT vs k-NN (k = 1)	< 2e−16	No	−0.613	Large
k-NN (k = 9) vs k-NN (k = 3)	0.00059	No	−0.271	Small
k-NN (k = 9) vs NB	< 2e−16	No	−0.61	Large
k-NN (k = 9) vs k-NN (k = 1)	< 2e−16	No	−0.614	Large
k-NN (k = 3) vs NB	< 2e−16	No	−0.606	Large
k-NN (k = 3) vs k-NN (k = 1)	< 2e−16	No	−0.613	Large
NB vs k-NN (k = 1)	0.00043	No	−0.276	Small

Table 3
Ranking of the information fusion layers including a meta-classifier γ_M with meta-vectors transformation and $|\mathbb{v}_M'| = 1$. O stands for Our ranking; H stands for H-TOPSIS ranking.

Metaclassifier	Configuration	Feature reduction method	Configuration	s	μ	O	H
DT	cp = 0.01 minsplit = 2	Sammon	trace = false	0.0474	0.8179	1	2
k-NN	k = 9	tSNE	pca = false	0.0526	0.8191	2	1
NB	laplace = 1	Sammon	trace = false	0.0418	0.8074	3	7
SVM	kernel = radial	Sammon	trace = false	0.0512	0.8102	4	4
DT	cp = 0.01 minsplit = 2	tSNE	pca = false	0.0524	0.8101	5	3
k-NN	k = 9	Sammon	trace = false	0.05197	0.8071	6	6
k-NN	k = 3	tSNE	pca = false	0.0556	0.8088	7	5
NB	laplace = 1	LDA	–	0.0478	0.7857	8	10
LDA	–	LDA	–	0.0489	0.7866	9	9
k-NN	k = 3	Sammon	trace = false	0.0549	0.7882	10	8
DT	cp = 0.01 minsplit = 2	LDA	–	0.0489	0.7762	11	12
SVM	kernel = radial	LDA	–	0.0552	0.7691	12	13
k-NN	k = 1	tSNE	pca = false	0.06	0.7732	13	11
DT	cp = 0.01 minsplit = 2	linear PCA	–	0.0543	0.7672	14	15
k-NN	k = 9	LDA	–	0.0547	0.7662	15	16
SVM	kernel = radial	linear PCA	–	0.0577	0.7669	16	14
k-NN	k = 9	linear PCA	–	0.0606	0.7638	17	17
k-NN	k = 3	LDA	–	0.0603	0.7502	18	19
k-NN	k = 1	Sammon	trace = false	0.0631	0.745	19	22
NB	laplace = 1	non-linear PCA	kernel = rbfdot sigma = 0.2	0.071	0.7463	20	20
k-NN	k = 3	linear PCA	–	0.0668	0.7421	21	23
SVM	kernel = radial	tSNE	pca = false	0.0821	0.7571	22	18
SVM	kernel = radial	non-linear PCA	kernel = rbfdot sigma = 0.2	0.069	0.7352	23	24
DT	cp = 0.01 minsplit = 2	non-linear PCA	kernel = rbfdot sigma = 0.2	0.0766	0.7418	24	21
k-NN	k = 9	non-linear PCA	kernel = rbfdot sigma = 0.2	0.082	0.7268	25	25
k-NN	k = 1	LDA	–	0.0627	0.7029	26	27
k-NN	k = 1	linear PCA	–	0.0701	0.6941	27	28
k-NN	k = 3	non-linear PCA	kernel = rbfdot sigma = 0.2	0.0918	0.7051	28	26
k-NN	k = 1	non-linear PCA	kernel = rbfdot sigma = 0.2	0.0945	0.664	29	29
NB	laplace = 1	tSNE	pca = false	0.1366	0.5612	30	30
NB	laplace = 1	linear PCA	–	0	0.3126	31	31

6.2. Experiment plan realisation

6.2.1. First experiment use case - information fusion layer with a meta-classifier without meta-vectors transformation

In the first experiment, we test the information fusion layer composed of a meta-classifier, which works directly on decisions produced by classifiers, so feature projection methods are not applied to these data (see the left flow in Fig. 4). The meta-classifiers are modelled by six models m_i, i.e. the machine algorithms with different configurations.

Table 1 lists the information fusion layers ordered according to our ranking method. The mean μ and standard deviation s values of $F - scores$ are provided, as well as the rank positions produced by the H-TOPSIS ranking algorithm. Note that $F - score$ values are presented in Fig. 5. The Spearman test indicates a strong positive relationship between both ranking methods. That is, the higher we ranked a particular fusion layer in the H-TOPSIS rank, the higher we ranked it in our ranking method, and vice versa. This conclusion is based on the Spearman coefficient of $r_s = 0.94$. Because the calculated statistical significance coefficient $p - value = 0.0048$ of the Spearman test is lower than the statistical significance value $\alpha = 0.05$, we can reject the null hypothesis H_{S0} stating that there is no monotonic association between the H-TOPSIS rank and our rank.

The Friedman's aligned rank test indicated that the F-scores from the experiments (the F-scores of the six best classification models from Table 1) were significantly different for the classification models ($p - value < 2.2e - 16$ and $p - value < .05$; thus we can reject hypothesis H_{F0}). In addition, Table 2 lists the p-values of the Wilcoxon signed-rank test for the pairs of results originating

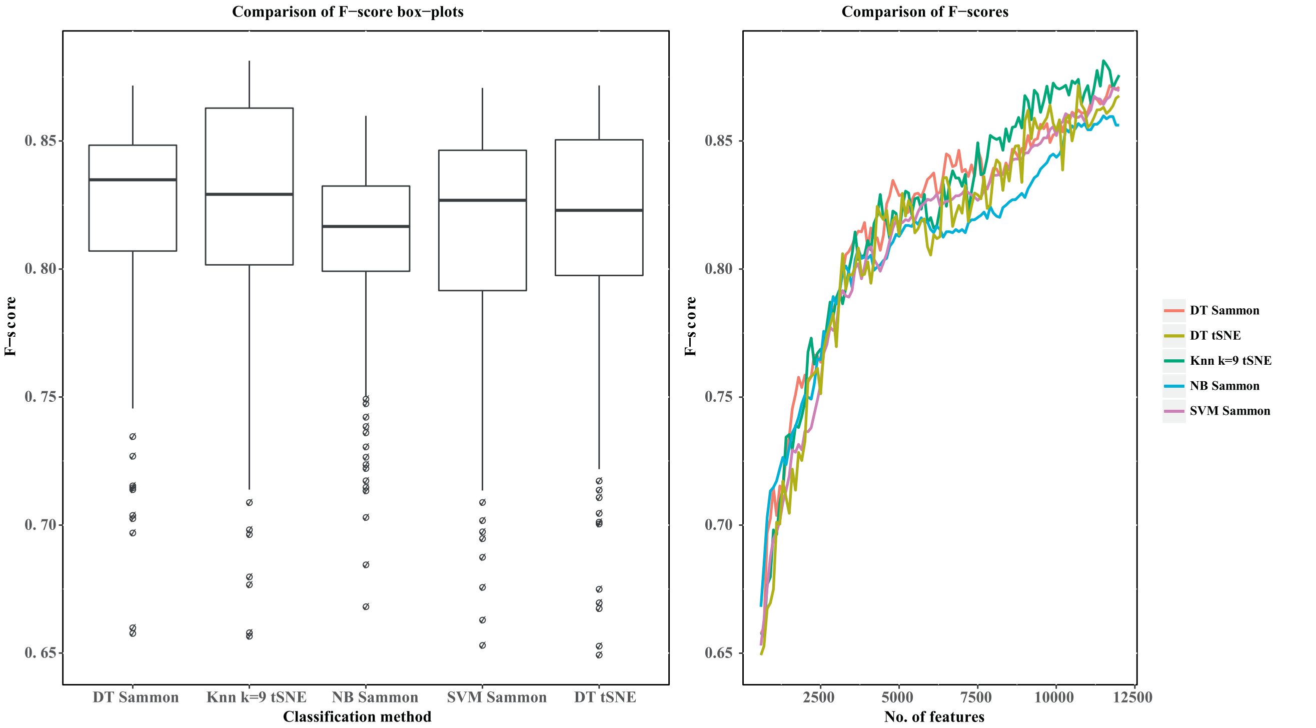

Fig. 6. Box-plot of the *F*-score median values (on the left) and the *F*-score plot (on the right) of the top five information fusion layers originating from the rankings presented in Table 3.

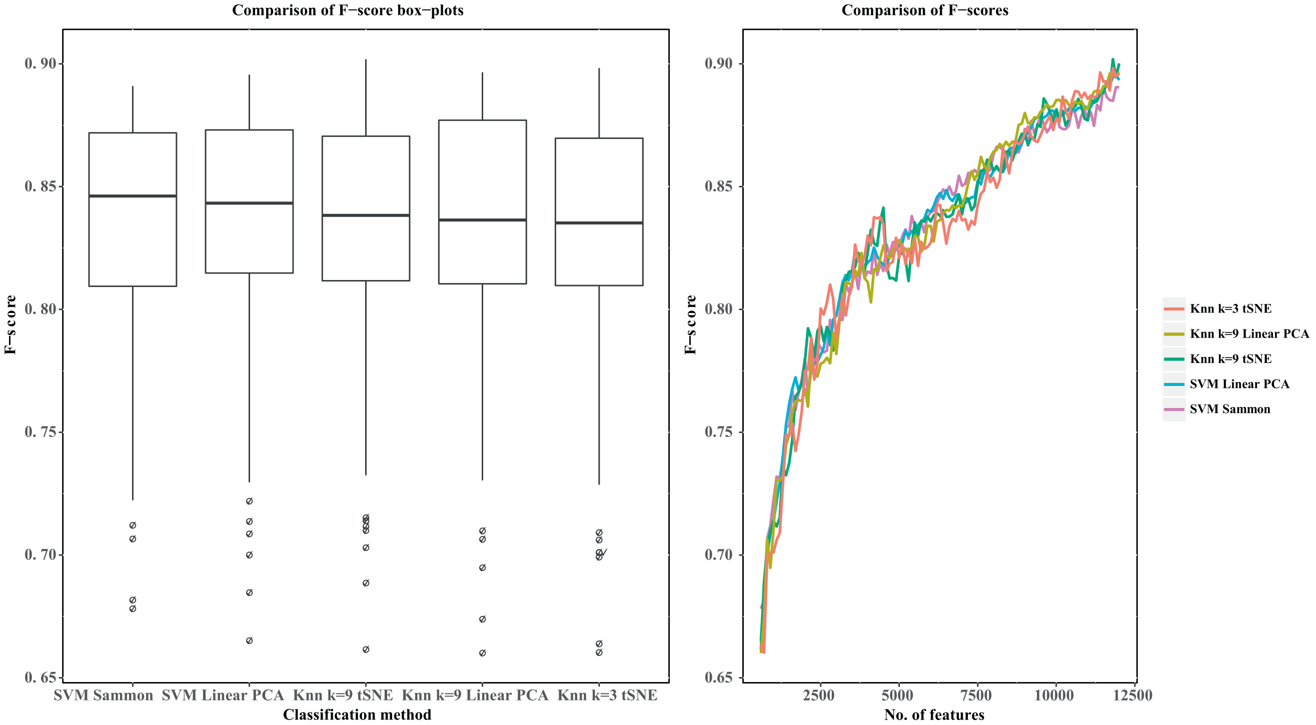

Fig. 7. Box-plot of the *F*-score median values (on the left) and the F-score plot (on the right) of the top five information fusion layers originating from the rankings presented in Table 5.

from Table 1; whereas, Fig. 5 presents the box-plots and F-score plots of the selected information fusion layers. Note that there is no significant difference between the results obtained by SVM method vs. DT and SVM vs. and k-NN, where k = 9. Moreover, the Wilcoxon signed-rank test indicates that these pairs of the algorithms are convergent to each other. According to the results, we cannot reject the null hypothesis H_{W0} for the above cases. However, the fusion layer based on the SVM produces different results than those based on the NB or k-NN (k = 1, and k = 3). Thus, we can reject the null hypothesis H_{W0} for these pairs. Also, for these pairs we noticed medium and large effect sizes, indicating that SVM outperforms the compared classification models considerably.

Table 4
The p-values of the Wilcoxon–Mann–Whitney test of the top information fusion layers originating from ranking presented in Table 3.

Layers pair	p-value	Are similar results? ($p - value \geq \alpha,\ \alpha = 0.05$)	Effect size (r)	Effect size label
DT Sammon vs k-NN k = 9 tSNE	1	Yes	−0.08	Small
DT Sammon vs NB Sammon	1.1e−14	No	−0.528	Large
DT Sammon vs SVM Sammon	6.3e−16	No	−0.551	Large
DT Sammon vs DT tSNE	4.7e−09	No	−0.411	Medium
k-NN k = 9 tSNE vs NB Sammon	2.7e−11	No	−0.461	Medium
k-NN k = 9 tSNE vs SVM Sammon	1.8e−14	No	−0.525	Large
k-NN k = 9 tSNE vs DT tSNE	2.8e−15	No	−0.539	Large
NB Sammon vs SVM Sammon	0.031	No	−0.195	Small
NB Sammonvs vs DT tSNE	0.057	Yes	−0.182	Small
SVM Sammon vs DT tSNE	1	Yes	−0.023	Small

Table 5
Ranking of the information fusion layers including a meta-classifier γ_M with meta-vectors transformation and $|\mathbb{v}'_M| = 2$. O stands for Our ranking; H stands for H-TOPSIS ranking.

Metaclassifier	Configuration	Feature reduction method	Configuration	s	μ	O	H
SVM	kernel = radial	Sammon	trace = false	0.0486	0.8311	1	2
SVM	kernel = radial	linear PCA	–	0.0494	0.8318	2	1
k-NN	k = 9	tSNE	pca = false	0.0506	0.8306	3	3
k-NN	k = 9	linear PCA	–	0.0528	0.8306	4	5
k-NN	k = 3	tSNE	pca = false	0.0522	0.829	5	4
k-NN	k = 9	Sammon	trace = false	0.0513	0.8236	6	6
k-NN	k = 3	Sammon	trace = false	0.0558	0.8147	7	7
k-NN	k = 3	linear PCA	–	0.0562	0.8141	8	8
DT	cp = 0.01 minsplit = 2	tSNE	pca = false	0.0543	0.8111	9	9
DT	cp = 0.01 minsplit = 2	linear PCA	–	0.0506	0.8059	10	10
DT	cp = 0.01 minsplit = 2	Sammon	trace = false	0.0508	0.8047	11	11
NB	laplace = 1	linear PCA	–	0.0449	0.7961	12	12
NB	laplace = 1	Sammon	trace = false	0.0454	0.7931	13	14
SVM	kernel = radial	tSNE	pca = false	0.0479	0.7932	14	15
k-NN	k = 1	tSNE	pca = false	0.058	0.7985	15	13
SVM	kernel = radial	non-linear PCA	kernel = rbfdot sigma = 0.2	0.0638	0.7899	16	16
k-NN	k = 1	linear PCA	–	0.0597	0.7808	17	18
k-NN	k = 1	Sammon	trace = false	0.0601	0.7809	18	19
k-NN	k = 9	non-linear PCA	kernel = rbfdot sigma = 0.2	0.0658	0.785	19	17
DT	cp = 0.01 minsplit = 2	non-linear PCA	kernel = rbfdot sigma = 0.2	0.0581	0.7713	20	20
NB	laplace = 1	non-linear PCA	kernel = rbfdot sigma = 0.2	0.0567	0.7684	21	21
k-NN	k = 3	non-linear PCA	kernel = rbfdot sigma = 0.2	0.0669	0.7676	22	22
NB	laplace = 1	tSNE	pca = false	0.0652	0.7503	23	23
k-NN	k = 1	non-linear PCA	kernel = rbfdot sigma = 0.2	0.0746	0.7343	24	24

Since we experimentally showed that (i) there is a monotonic association between the H-TOPSIS rank and our rank and (ii) some of the information fusion layers produce statistically different results when using different training algorithms, whereas some do not, we can conclude the following:

1. Our ranking method gives similar ranks as the H-TOPSIS method.
2. The results of information fusion can be influenced by some training algorithms, especially when we consider SVM vs. NB or SVM vs. k-NN (k = 1 and k = 3). Also note that other algorithms, such as SVM vs. DT or SVM vs. k-NN (k = 9), do not influence the fusion layer.

These conclusions are valid for the information fusion layer composed of the meta-classifier utilising decisions directly from prior classifiers, i.e. without any transformation.

6.2.2. Second experiment use case - information fusion layer with a meta-classifier and meta-vectors transformation

The next three experiments involve an examination of an information fusion layer composed of a meta-classifier when feature projection methods are applied. The decisions coming from classifiers are merged into a VSM, and are then projected to another features space, which is utilised by the meta-classifier. The three subsequent experiments assume that the resulting VSMs contain one, two, or three features, respectively.

First experiment sub-use case - One feature case. The projected VSM contains only one feature $|\mathbb{v}'_M| = 1$, which is the minimum degree of feature space reduction possible. We prepare 31 different information fusion layers using this data set and various training and projection algorithms.

Table 3 shows two rankings of the fusion layers and the values of mean μ and standard deviation s of the $F - scores$, and values of the best five cases are depicted in Fig. 6. As in the previous experiment, the Spearman test on the rankings indicates a strong positive relationship between our rank and the H-TOPSIS rank, where $r_s = 0.98$. Moreover, as the statistical significance coefficient $p - value = 0$, we can reject the null hypothesis H_{S0}. Thus, there is a monotonic association between the H-TOPSIS rank and our rank.

The Friedman's aligned rank test indicated that the F-score from the experiments (the F-score of the five best classification models from Table 3) significantly change for the classification models ($p - value < 2.2e - 16$ and $p - value < .05$. Thus, we can reject hypothesis H_{F0}). In addition, Table 4 contains the selected p-values of the Wilcoxon signed-rank test. The test shows that there is no difference between results obtained by the DT Sammon method vs. k-NN (k = 9) tSNE, so we conclude that these pairs are dependent. As a consequence, we cannot reject the null hypothesis H_{W0}. By contrast, the fusion layers based on the DT Sammon produced different results to those based NB Sammon, SVM Sammon, and DT tSNE, so we can reject the null hypothesis H_{W0} for these

Table 6
The p-values of the Wilcoxon–Mann–Whitney test of the top information fusion layers originating from ranking presented in Table 5.

Layers pair	p-value	Are similar results? ($p - value \geq \alpha,\ \alpha = 0.05$)	Effect size (r)	Effect size label
SVM Sammon vs. SVM Linear PCA	1	Yes	−0.09	Small
SVM Sammon vs. k-NN k = 9 tSNE	1	Yes	−0.04	Small
SVM Sammon vs. k-NN k = 9 Linear PCA	1	Yes	−0.038	Small
SVM Sammon vs. knn k = 3 tSNE	0.474	Yes	−0.131	Small
SVM Linear PCA vs. k-NN k = 9 tSNE	0.45	Yes	−0.132	Small
SVM Linear PCA vs. k-NN k = 9 Linear PCA	0.927	Yes	−0.111	Small
SVM Linear PCA vs. k = 3 tSNE	0.006	No	−0.226	Small
k-NN k = 9 tSNE vs. k-NN k = 9 Linear PCA	1	Yes	−0.042	Small
k-NN k = 9 tSNE vs. knn k = 3 tSNE	0.212	Yes	−0.152	Small
k-NN k = 9 Linear PCA vs. knn k = 3 tSNE	0.155	Yes	−0.16	Small

Table 7
Ranking of the information fusion layers including a meta-classifier γ_M with meta-vectors transformation and $|\mathbb{v}^{'}_M| = 3$. O stands for Our ranking; H stands for H-TOPSIS ranking.

Metaclassifier	Configuration	Feature reduction method	Configuration	s	μ	O	H
SVM	kernel = radial	Sammon	trace = false	0.0493	0.8478	1	1
SVM	kernel = radial	linear PCA	–	0.051	0.8469	2	2
k-NN	k = 9	linear PCA	–	0.0502	0.8372	3	3
k-NN	k = 9	Sammon	trace = false	0.0494	0.8315	4	4
k-NN	k = 9	tSNE	pca = false	0.0502	0.8314	5	5
k-NN	k = 3	linear PCA	–	0.0562	0.8141	6	10
k-NN	k = 3	tSNE	pca = false	0.05269	0.8261	7	7
k-NN	k = 3	Sammon	trace = false	0.0537	0.8268	8	6
NB	laplace = 1	linear PCA	–	0.045	0.8126	9	11
NB	laplace = 1	Sammon	trace = false	0.041	0.8072	10	14
DT	cp = 0.01 minsplit = 2	tSNE	pca = false	0.053	0.8171	11	8
SVM	kernel = radial	tSNE	pca = false	0.0536	0.8156	12	9
k-NN	k = 9	non-linear PCA	kernel = rbfdot sigma = 0.2	0.0502	0.8079	13	12
DT	cp = 0.01 minsplit = 2	Sammon	trace = false	0.0554	0.8076	14	13
DT	cp = 0.01 minsplit = 2	linear PCA	–	0.0554	0.8054	15	16
k-NN	k = 1	linear PCA	–	0.0586	0.8062	16	15
SVM	kernel = radial	non-linear PCA	kernel = rbfdot sigma = 0.2	0.0528	0.7984	17	17
k-NN	k = 3	non-linear PCA	kernel = rbfdot sigma = 0.2	0.0492	0.7937	18	20
k-NN	k = 1	Sammon	trace = false	0.0579	0.7973	19	18
k-NN	k = 1	tSNE	pca = false	0.0563	0.7955	20	19
DT	cp = 0.01 minsplit = 2	non-linear PCA	kernel = rbfdot sigma = 0.2	0.0524	0.7893	21	21
NB	laplace = 1	non-linear PCA	kernel = rbfdot sigma = 0.2	0.0557	0.768	22	22
k-NN	k = 1	non-linear PCA	kernel = rbfdot sigma = 0.2	0.0563	0.7663	23	23
NB	laplace = 1	tSNE	pca = false	0.0508	0.7565	24	24

cases. Also, for these pairs we noticed large and medium effect sizes, meaning that DT Sammon outperforms other compared classification models considerably.

We conclude that both ranking methods produce similar results when the meta-classifier uses the reduced VSM representing one feature. Mostly, the results of meta-classification are different regarding the combination of training algorithms and projection methods; however, we found some influence when comparing Decision Tress with Naive Bayes, SVM and the Sammon projection methods.

Second experiment sub-use case - Two features case. In the next experiment, the projected VSM contains two features $|\mathbb{v}^{'}_M| = 2$. Based on the previous results, we extended the number of fusion layer combinations to 24. They are trained by using the various combinations of training algorithms and projection methods. The results are shown in Tables 5, 6, and Fig. 7.

According to the Spearman test carried out on the rankings included in Table 5, we can reject the null hypothesis H_{S0} because the statistical significance coefficient is $p - value = 0$. Thus, we can conclude that there is monotonic association between the H-TOPSIS rank and our rank. The Spearman coefficient $r_s = 0.99$ indicates that these rankings are strongly positively correlated.

The Friedman's aligned rank test indicated that the F-score from the experiments (the F-score of the five best classification models from Table 5) are significantly changed for the classification models ($p - value = 0.004893$ and $p - value < .05$, thus we can reject hypothesis H_{F0}). In addition, the Wilcoxon signed-rank test carried out on data shown in Table 5 indicates that almost all pairs of experiments are dependent (see the p-values in Table 6). Thus, we cannot reject the null hypothesis H_{W0}. There is only one different pair, i.e. SVM Linear PCA obtains different results that k-NN (k = 3) tSNE. Also, for this pair we noticed a small effect size, meaning that SVM outperforms the other compared classification models considerably.

In conclusion, we observe that the meta-classifier trained on the VSM reduced to two features is independent of the training algorithm and data projection method except for one case. Moreover, both ranking methods assess the meta-classifiers in the same manner.

Third experiment sub-use case - Three features case. Finally, we test the meta-classifiers that use the projected VSM containing three features $|\mathbb{v}^{'}_M| = 3$. There are 24 combinations of the information fusion layer, i.e. training and projection methods as in the prior experiment. The results are shown in Tables 7, 8, and Fig. 8.

The Spearman coefficient $r_s = 0.97$ calculated on the rankings included in Table 7 proves again that the outcomes of both ranking methods are positively correlated. As in the two features case, the statistical significance coefficient $p - value$ is equal to 0, thus once more we can reject the null hypothesis H_{S0}. Therefore, there is a monotonic association between both rankings.

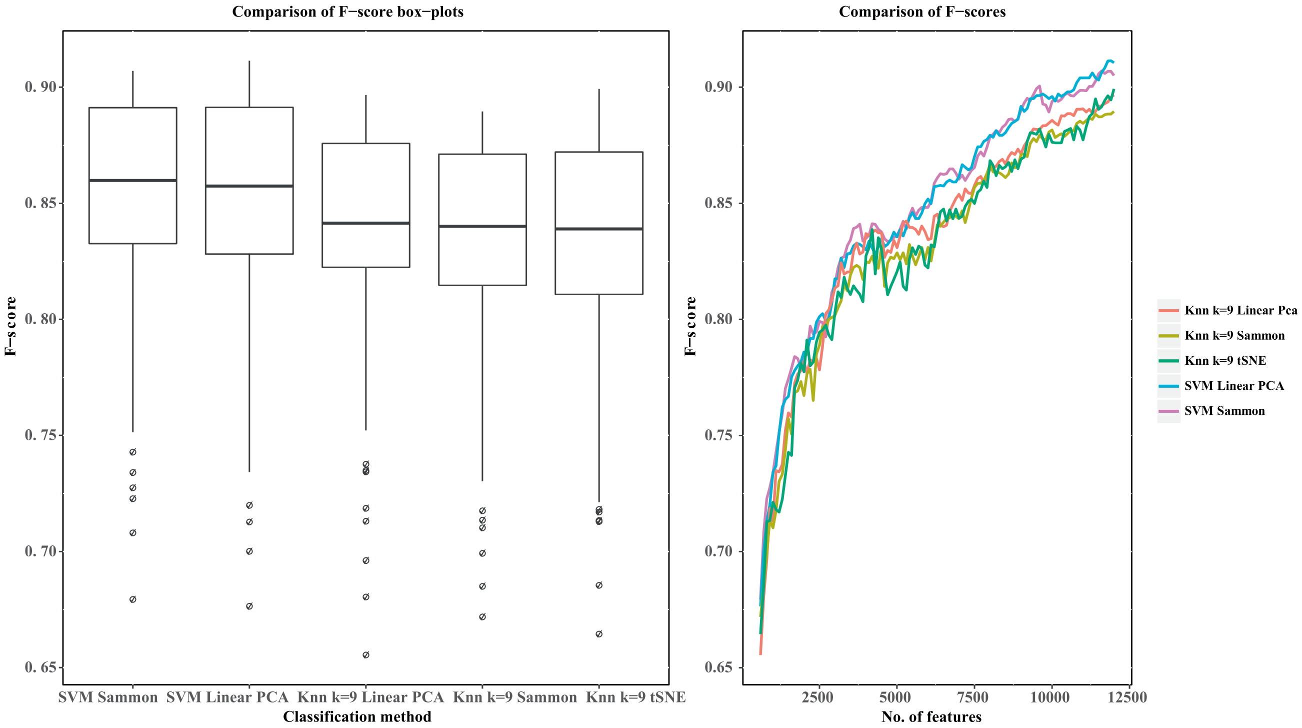

Fig. 8. Box-plot of the *F*-score median values (on the left) and the *F*-score plot (on the right) of the top five information fusion layers originating from the rankings presented in Table 7.

Table 8
The *p*-values of the Wilcoxon–Mann–Whitney test of the top information fusion layers originating from ranking presented in Table 7.

Layers pair	*p*-value	Are similar results? ($p - value \geq \alpha,\ \alpha = 0.05$)	Effect size (r)	Effect size label
SVM Sammon vs. SVM Linear PCA	0.53	Yes	−0.128	Small
SVM Sammon vs. k-NN k = 9 Linear PCA	< 2e−16	No	−0.598	Large
SVM Sammon vs. k-NN k = 9 Sammon	< 2e−16	No	−0.613	Large
SVM Sammon vs. knn k = 9 tSNE	< 2e−16	No	−0.61	Large
SVM Linear PCA vs. k-NN k = 9 Linear PCA	< 2e−16	No	−0.568	Large
SVM Linear PCA vs. k-NN k = 9 Sammon	< 2e−16	No	−0.611	Large
SVM Linear PCA vs. knn k = 9 tSNE	< 2e−16	No	−0.60	Large
k-NN k = 9 Linear PCA vs. k-NN k = 9 Sammon	6.7e−15	No	−0.532	Large
k-NN k = 9 Linear PCA vs. knn k = 9 tSNE	5.7e−10	No	−0.432	Medium
k-NN k = 9 Sammon vs. knn k = 9 tSNE	1	Yes	−0.013	Small

The Friedman's aligned rank test indicated that the *F*-score from the experiments (the F-score of the five best classification models from Table 7) are significantly changed over the classification models ($p - value < 2.2e - 16$ and $p - value < .05$, thus we can reject hypothesis H_{F0}). In addition, the Wilcoxon signed-rank test on data included in Table 7 shows that there is no difference between results obtained by the meta-classifiers trained by SVM Sammon or SVM Linear PCA algorithms. The results of these experiments are convergent, so we cannot reject the null hypothesis H_{W0}. By contrast, we can reject the null hypothesis H_{W0} when we compare the meta-classifiers based on the SVM Sammon vs. k-NN (k = 9) Linear PCA, k-NN (k = 9) Sammon, and k-NN (k = 9) tSNE. Also, for these pairs we noticed large effect sizes, meaning that SVM outperforms the compared classification models considerably.

In conclusion, we observe again that both training methods produce the same results. On the other hand, the meta-classifiers utilising the VSM with three features may be influenced by some training algorithms.

6.3. Summarisation and discussion of the results

In this subsection, we summarise the experiments above and highlight the most promising results. For this purpose, we selected the best configuration of the fusion layer from each experiment (Table 9). Table 10 contains the comparison of *p*-values of the Wilcoxon signed-rank test carried out on data presented in Table 9. Fig. 9 includes the box-plots and *F*-score plots of the best configurations of the fusion layers.

The Friedman's aligned rank test indicated that the F-scores from the experiments (the *F*-scores of the four best classification models from Table 1) are significantly changed for the classification models ($p - value < 2.2e - 16$ and $p - value < .05$, thus we can reject hypothesis H_{F0}). In addition, the results included

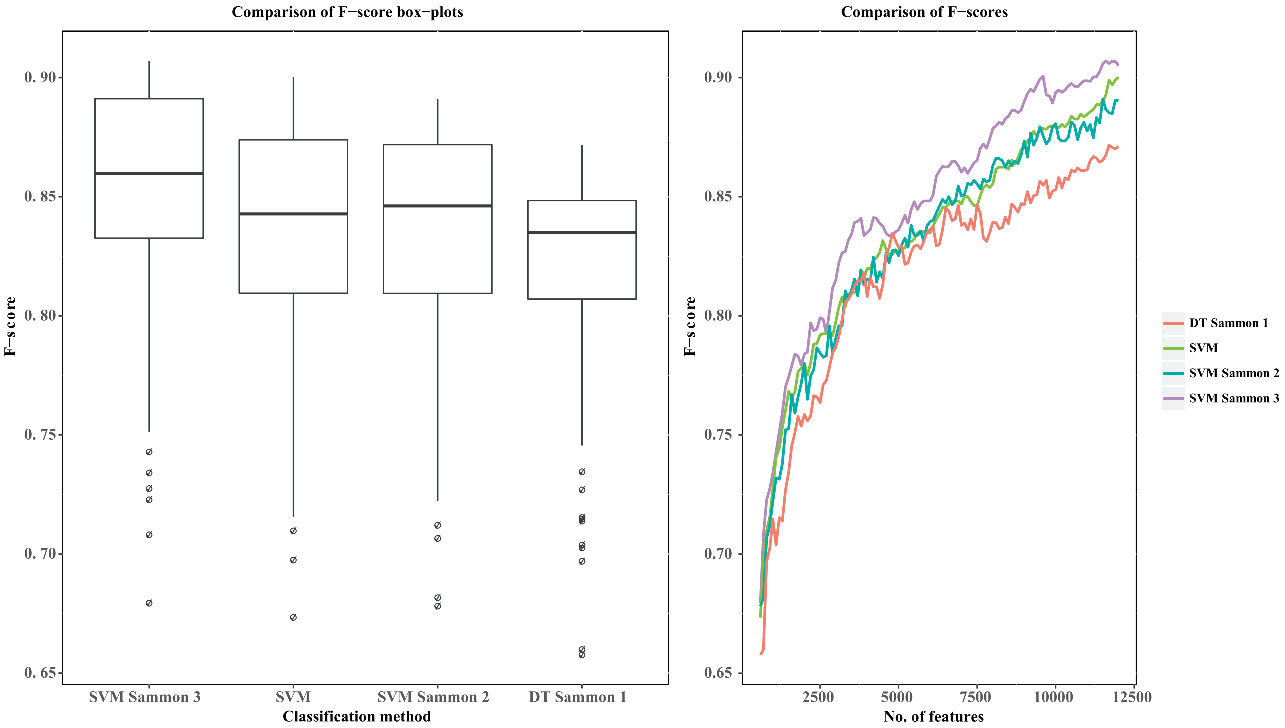

Fig. 9. Box-plot in terms of median (left chart) and *F*-score plot (right chart) of the best five meta-classification methods γ_M from the rankings in Table 9.

Table 9
Ranking information fusion layers including a meta-classifier γ_M obtained for $\mathbb{v}'_M = \mathbb{v}_M(|\mathbb{v}'_M| = 3)$, $|\mathbb{v}'_M| = 1$, $|\mathbb{v}'_M| = 2$, and $|\mathbb{v}'_M| = 3$.

Metaclassifier	Configuration	Feature reduction method	Configuration	Features number	s	μ
SVM	kernel = radial	Sammon	trace = false	3	0.0493	0.8478
SVM	kernel = radial	–	–	3	0.0472	0.8337
SVM	kernel = radial	Sammon	trace = false	2	0.0486	0.8311
DT	cp = 0.01 minsplit = 2	Sammon	trace = false	1	0.0474	0.8179

Table 10
The *p*-values of the Wilcoxon–Mann–Whitney test of the top information fusion layers originating from ranking presented in Table 9.

Layers pair	*p*-value	Are similar results? ($p - value \geq \alpha$, $\alpha = 0.05$)	Effect size (r)	Effect size label
SVM Sammon 3 vs. SVM	< 2e−16	No	−0.613	Large
SVM Sammon 3 vs. SVM Sammon 2	< 2e−16	No	−0.614	Large
SVM Sammon 3 vs. DT Sammon 1	< 2e−16	No	−0.613	Large
SVM vs. SVM Sammon 2	0.00057	No	−0.257	Small
SVM vs. DT Sammon 1	< 2e−16	No	−0.585	Large
SVM Sammon 2 vs. DT Sammon 1	< 2e−16	No	−0.584	Large

in Table 10 indicate that all classification models are different. The Wilcoxon signed-rank test indicates that all pairs of experiments are independent, i.e. we can reject the null hypothesis H_{W0}. Moreover, for the first pair, we noticed large effect size, meaning that SVM with the Sammon projection feature in three dimensional space considerably outperforms the SVM classification models without this modification in the same space.

The results empirically prove that the information fusion layer enhances classification quality. The SVM classifier utilising Sammon projection to three dimensional feature space is better by 1.41% point in terms of the F-score mean than SVM that does not utilise the transformed feature space. From a statistical point of view, these results are different. Furthermore, we can observe that SVM with Sammon projection with the three features tends to achieve better results over the whole spectrum of feature numbers. Also, it interesting that it yields almost the same results as basic SVM using a lower space, i.e. a two dimensional feature space.

7. Conclusions

We have proposed a multi-view framework for text classification. The framework is composed of (i) a multi-view layer and (ii) an information fusion layer. These two layers help to process and categorise data more precisely in succeeding layers. The multi-view layer covers classification models trained on data views for the same data set but cover different collections of features. The information fusion layer includes a feature projection method and a meta-classifier. The projection method transforms outputs produced by classification models in the first layer into a new vector space model, whereas the meta-classifier forms the final decision.

In addition, we have proposed a heuristic ranking method aimed to assess various combinations of the information fusion layer. More specifically, it ranks various compositions of feature projection and model training algorithms, including their parameters. The ranking method utilises the statistical properties of *F*-score values such as mean μ and standard deviation *s*. The *F*-scores are calculated on classification results that are produced by the fusion layer. For this purpose, we introduced a use case checking whether companies' domains indicate their innovativeness. The significance of results was validated by the Spearman, Friedman's aligned rank and Wilcoxon signed-rank statistical tests.

The experimental results indicate that the information fusion layer may improve the classification quality when a meta-classifier works with a features projection method. In addition, tests carried out on the results verified that our ranking method is convergent to the well-established and popular H-TOPSIS method. Thus, it may be used for classifier assessments.

The idea of the fusion of several views of data may be applied to methods for extracting information from textual data to provide multi-view patterns. Moreover, they can be assessed by the proposed ranking method to select the best pattern. As text mining methods differ from classification tasks, it is impossible to simply transfer the elaborated solution. Such a new approach may be the topic of further studies on multi-view information fusion in text mining.

Our research has some limitations that might be alleviated in future studies. We only investigated the problem of Internet website categorisation considering two classes: *an innovative company website* or *a non-innovative company website*. Furthermore, we only focussed on three naturally created views from text data. However, aside from these limitations, we explore and discuss an interesting real use case representing a real problem firstly evaluated in our information system called Inventorum (Protasiewicz, 2017a,b).

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Another empirical evaluation of the rankings

In this appendix, we show additional empirical evidence for convergence of both rankings, i.e. the proposed one and H-TOPSIS. This section compares the proposed ranking method with H-TOPSIS. We use the data sets and resultsobtained by Krohling et al. (2015) and Krohling and Pacheco (2015) to check if our ranking method yields the same or nearly equal results to H-TOPSIS. For this purpose, we test the rankings obtained by H-TOPSIS and our ranking method using the Spearman test r_s (Myles et al., 2014) and assuming that the statistical significance coefficient is $\alpha = 0.05$. There are two experimental series, namely (i) tests of 21 algorithms yielded $r_s = 0.94$ and $p - value = 0$, whereas (ii) tests of 22 algorithms produced $r_s = 0.96$ and $p - value = 0$. In both cases, we can reject the null hypothesis H_0 stating that there is no monotonic association between the H-TOPSIS ranking and our ranking. Table A.11 includes the obtained results.

Table A.11
Comparison of the H-TOPSIS ranking and our ranking for 21 algorithms (on the left) and for 22 algorithms (on the right). H stands for H-TOPSIS ranking; O stands for Our ranking.

Algorithm	H	O	Algorithm	H	O
.dGenocop	1	1	GSA	1	1
GenocopwUPCwNRR	2	2	.dGenocop	2	2
Genocop	3	4	GenocopwUPCwNRR	3	3
dRepairHyperMOOR	4	3	Genocop	4	5
dRepairHyperM	5	5	dRepairHyperMOOR	5	4
dRepairRIGAOOR	6	6	dRepairHyperM	6	6
dRepairRIGA	7	7	dRepairRIGAOOR	7	7
dRepairGAOOR	8	8	dRepairRIGA	8	8
dRepairGA	9	10	dRepairGAOOR	9	9
GA+RepairwUPCwNRR	10	9	dRepairGA	10	11
RIGAelit	11	12	GA+RepairwUPCwNRR	11	10
GA+Repair	12	11	RIGAelit	12	13
HyperMelit	13	15	GA+Repair	13	12
HyperMnoElit	14	17	HyperMelit	14	14
RIGAnoElit	15	20	HyperMnoElit	15	17
Gaelit	16	19	Gaelit	16	20
GA+RepairwUPGwRR	17	13	RIGAnoElit	17	21
GenocopwUPGwNRR	18	14	GA+RepairwUPGwRR	18	15
GA+RepairwUPGwNR	19	16	GenocopwUPGwNRR	19	16
.GenocopwUPGwRR	20	18	GA+RepairwUPGwNR	20	18
GAnoElit	21	21	.GenocopwUPGwRR	21	19
–	–	–	GAnoElit	22	22

Owing to these tests, we may conclude that the proposed ranking method which do not use any statistic assumptions and do not utilise any information about benefits and costs provide the same results as the state-of-the-art H-TOPSIS method, which utilises the properties mentioned above. From this point of view our method is simpler, i.e. it does not require configuration of any parameters.

Credit authorship contribution statement

Marcin Michał Mirończuk: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Writing - original draft, Writing - review & editing. **Jarosław Protasiewicz:** Formal analysis, Validation, Writing - original draft, Writing - review & editing. **Witold Pedrycz:** Writing - original draft, Writing - review & editing.

References

Ali, R., Lee, S., & Chung, T. C. (2017). Accurate multi-criteria decision making methodology for recommending machine learning algorithm. *Expert Systems with Applications, 71*, 257–278. doi:10.1016/j.eswa.2016.11.034.

Basto-Fernandes, V., Yevseyeva, I., Méndez, J. R., Zhao, J., Fdez-Riverola, F., & Emmerich, M. T. (2016). A spam filtering multi-objective optimization study covering parsimony maximization and three-way classification. *Applied Soft Computing, 48*, 111–123.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on computational learning theory* (pp. 92–100). ACM.

Bordens, K., & Abbott, B. (2017). *Research design and methods a process approach* (10th ed.). McGraw-Hill Higher Education.

Calvo, B., & Santafe, G. (2015). scmamp: Statistical comparison of multiple algorithms in multiple problems. *The R Journal, Accepted for publication*.

Chang, M., & Poon, C. K. (2009). Using phrases as features in email classification. *Journal of Systems and Software, 82*(6), 1036–1045. doi:10.1016/j.jss.2009.01.013.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*. Lawrence Erlbaum Associates.

Cohen, J. (1992). A power primer. *Psychological bulletin, 112*(1), 155.

Cuzzola, J., Jovanović, J., Bagheri, E., & Gašević, D. (2015). Automated classification and localization of daily deal content from the web. *Applied Soft Computing, 31*, 241–256.

Dasigi, V., Mann, R. C., & Protopopescu, V. A. (2001). Information fusion for text classification - An experimental comparison. *Pattern Recognition, 34*(12), 2413–2425. doi:10.1016/S0031-3203(00)00171-0.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30. http://www.jmlr.org/papers/v7/demsar06a.html.

Du, Q., & Fowler, J. E. (2008). Low-complexity principal component analysis for hyperspectral image compression. *The International Journal of High Performance Computing Applications, 22*(4), 438–448.

Field, A., Miles, J., & Field, Z. (2012). *Discovering statistics using R*. Sage publications.

Forman, G., & Scholz, M. (2010). Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement. *ACM SIGKDD Explorations Newsletter, 12*(1), 49–57. doi:10.1145/1882471.1882479.

de Fortuny, E. J., Smedt, T. D., Martens, D., & Daelemans, W. (2014). Evaluating and understanding text-based stock price prediction models. *Information Processing and Management, 50*(2), 426–441. doi:10.1016/j.ipm.2013.12.002.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association, 32*(200), 675–701.

Ge, Z., & Song, Z. (2012). *Multivariate statistical process control: Process monitoring methods and applications*. Springer Science & Business Media.

Ghani, R., Slattery, S., & Yang, Y. (2001). Hypertext categorization using hyperlink patterns and meta data. In C. E. Brodley, & A. P. Danyluk (Eds.), *Proceedings of the eighteenth international conference on machine learning (ICML 2001), williams college, Williamstown, MA, USA, June 28 - July 1, 2001* (pp. 178–185). Morgan Kaufmann.

Giachanou, A., Salampasis, M., & Paltoglou, G. (2015). Multilayer source selection as a tool for supporting patent search and classification. *Information Retrieval Journal, 18*(6), 559–585. doi:10.1007/s10791-015-9270-2.

Gu, P., Zhu, Q., & Zhang, C. (2009). A multi-view approach to semi-supervised document classification with incremental naive bayes. *Computers & Mathematics with Applications, 57*(6), 1030–1036.

Hajmohammadi, M. S., Ibrahim, R., & Selamat, A. (2014). Cross-lingual sentiment classification using multiple source languages in multi-view semi-supervised learning. *Engineering Applications of Artificial Intelligence, 36*, 195–203. doi:10.1016/j.engappai.2014.07.020.

Harrell, F. E., Jr, with contributions from Charles Dupont, et al. (2016). *Hmisc: Harrell miscellaneous*. R package version 4.0-2. https://CRAN.R-project.org/package=Hmisc.

Hwang, C.-L., & Yoon, K. (1981). Methods for multiple attribute decision making. In *Multiple attribute decision making: methods and applications a state-of-the-art survey* (pp. 58–191)). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-48318-9_3.

Ittoo, A., Nguyen, L. M., & van den Bosch, A. (2016). Text analytics in industry: Challenges, desiderata and trends. *Computers in Industry, 78*, 96–107. doi:10.1016/j.compind.2015.12.001.

Kan, M.-Y., & Thi, H. O. N. (2005). Fast webpage classification using url features. In *Proceedings of the 14th ACM international conference on information and knowledge management*. In *CIKM '05* (pp. 325–326). New York, NY, USA: ACM. doi:10.1145/1099554.1099649.

Karatzoglou, A., Smola, A., Hornik, K., & Zeileis, A. (2004). kernlab – An S4 package for kernel methods in R. *Journal of Statistical Software, 11*(9), 1–20. http://www.jstatsoft.org/v11/i09/.

Kou, G., Lu, Y., Peng, Y., & Shi, Y. (2012). Evaluation of classification algorithms using MCDM and rank correlation. *International Journal of Information Technology and Decision Making, 11*(1), 197–225. doi:10.1142/S0219622012500095.

Krijthe, J. H. (2015). *Rtsne: T-distributed stochastic neighbor embedding using a barnes-hut implementation*. https://github.com/jkrijthe/Rtsne.

Krohling, R. A., Lourenzutti, R., & Campos, M. (2015). Ranking and comparing evolutionary algorithms with Hellinger-topsis. *Applied Soft Computing, 37*(C), 217–226. doi:10.1016/j.asoc.2015.08.012.

Krohling, R. A., & Pacheco, A. G. (2015). A-topsis - An approach based on topsis for ranking evolutionary algorithms. *Procedia Computer Science, 55*, 308–317. doi:10.1016/j.procs.2015.07.054.

Kumar, B. S., & Ravi, V. (2016). A survey of the applications of text mining in financial domain. *Knowledge-Based Systems, 114*, 128–147.

Kumar, V., & Minz, S. (2016). Multi-view ensemble learning: an optimal feature set partitioning for high-dimensional data classification. *Knowledge and Information Systems, 49*(1), 1–59.

Li, Y., & Shawe-Taylor, J. (2007). Advanced learning algorithms for cross-language patent retrieval and classification. *Information Processing and Management, 43*(5), 1183–1199. Patent Processing, doi: 10.1016/j.ipm.2006.11.005.

Lim, C. S., Lee, K. J., & Kim, G. (2005). Multiple sets of features for automatic genre classification of web documents. *Information Processing and Management, 41*(5), 1263–1276. doi:10.1016/j.ipm.2004.06.004.

Lin, S. (2009). A document classification and retrieval system for r&d in semiconductor industry - A hybrid approach. *Expert Systems with Applications, 36*(3), 4753–4764. doi:10.1016/j.eswa.2008.06.024.

Liu, B. (2006). *Web data mining: exploring hyperlinks, contents, and usage data (data–centric systems and applications)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. New York, NY, USA: Cambridge University Press.

Matsubara, E. T., Monard, M. C., & Batista, G. E. A. P. A. (2005). Multi-view semi-supervised learning: An approach to obtain different views from text datasets. In K. Nakamatsu, & J. M. Abe (Eds.), *Advances in logic based intelligent systems - selected papers of LAPTEC 2005, Himeji, Japan, april 2–4, 2005*. In *Frontiers in Artificial Intelligence and Applications: 132* (pp. 97–104). IOS Press. http://www.booksonline.iospress.nl/Content/View.aspx?piid=1046.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., & Leisch, F. (2015). *e1071: Misc functions of the department of statistics, probability theory group (formerly: E1071), tu wien*. R package version 1.6-7. https://CRAN.R-project.org/package=e1071.

Mironczuk, M., & Protasiewicz, J. (2016). A diversified classification committee for recognition of innovative internet domains. In S. Kozielski, D. Mrozek, P. Kasprowski, B. MaÅiak-Mrozek, & D. Kostrzewa (Eds.), *Beyond databases, architectures and structures. advanced technologies for data mining and knowledge discovery: 12th international conference, BDAS 2016, UstroÅPoland, May 31–June 3, 2016, Proceedings* (pp. 368–383)). Cham: Springer International Publishing.

Mostafa, J., & Lam, W. (2000). Automatic classification using supervised learning in a medical document filtering application. *Information Processing and Management, 36*(3), 415–444. doi:10.1016/S0306-4573(99)00033-3.

Myles, H., Douglas, A. W., & Eric, C. (2014). *Nonparametric statistical methods* (3rd ed.). Wiley.

Parlak, B., & Uysal, A. K. (2015). Classification of medical documents according to diseases. In *2015 23nd signal processing and communications applications conference (siu)* (pp. 1635–1638). doi:10.1109/SIU.2015.7130164.

Protasiewicz, J. (2017a). Inventorum - A recommendation system connecting business and academia. *2017 IEEE international conference on systems, man, and cybernetics (smc)*. Banff, Canada: IEEE.

Protasiewicz, J. (2017b). Inventorum: A platform for open innovation. *2017 IEEE international conference on systems, man, and cybernetics (SMC)*. Banff, Canada: IEEE.

Qi, X., & Davison, B. D. (2009). Web page classification: Features and algorithms. *ACM Computing Surveys, 41*(2), 12:1–12:31. doi:10.1145/1459352.1459357.

R Core Team (2016). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Salkind, N. (2010). *Encyclopaedia of research design*: 1. Sage Publications.

(2010a). Cross-validation. In C. Sammut, & G. I. Webb (Eds.), *Encyclopedia of machine learning* (pp. 249–249)). Boston, MA: Springer US. doi:10.1007/978-0-387-30164-8_190.

Sammut, C., & Webb, G. I. (Eds.). (2010b). *Encyclopedia of machine learning*. Boston, MA: Springer US 912–912 doi:10.1007/978-0-387-30164-8_778.

Shen, L., Chen, H., Yu, Z., Kang, W., Zhang, B., Li, H., et al. (2016). Evolving support vector machines using fruit fly optimization for medical data classification. *Knowledge-Based Systems, 96*, 61–75.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management, 45*(4), 427–437.

Sun, S. (2013). A survey of multi-view machine learning. *Neural Computing and Applications, 23*(7–8), 2031–2038.

Therneau, T., Atkinson, B., & Ripley, B. (2015). *rpart: Recursive partitioning and regression trees*. R package version 4.1-10. https://CRAN.R-project.org/package=rpart.

Trappey, A. J. C., Hsu, F., Trappey, C. V., & Lin, C. (2006). Development of a patent document classification and search platform using a back-propagation network. *Expert Systems with Applications, 31*(4), 755–765. doi:10.1016/j.eswa.2006.01.013.

Venables, W. N., & Ripley, B. D. (2002). *Modern Applied Statistics with S* (4th). New York: Springer. ISBN 0-387-95457-0. http://www.stats.ox.ac.uk/pub/MASS4.

Wong, T.-T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition, 48*(9), 2839–2846. doi:10.1016/j.patcog.2015.03.009.

Woźniak, M., Graña, M., & Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems. *Information Fusion, 16*, 3–17.

Xu, C., Tao, D., & Xu, C. (2013). A survey on multi-view learning. arXiv:1304.5634.

Zavadskas, E. K., Zakarevicius, A., & Antucheviciene, J. (2006). Evaluation of ranking accuracy in multi-criteria decisions. *Informatica, Lithuanian Academy of Sciences, 17*(4), 601–618. http://content.iospress.com/articles/informatica/inf17-4-10.

Zhang, C., Liu, C., Zhang, X., & Almpanidis, G. (2017). An up-to-date comparison of state-of-the-art classification algorithms. *Expert Systems with Applications, 82*, 128–150. doi:10.1016/j.eswa.2017.04.003.

Zhao, J., Xie, X., Xu, X., & Sun, S. (2017). Multi-view learning overview: Recent progress and new challenges. *Information Fusion, 38*, 43–54.