Journal of Computer Languages 75 (2023) 101207

Contents lists available at ScienceDirect

Journal of Computer Languages

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

Machine learning with word embedding for detecting web-services anti-patterns

Lov Kumar [a], Sahithi Tummalapalli [b], Sonika Chandrakant Rathi [b], Lalita Bhanu Murthy [b], Aneesh Krishna [c], Sanjay Misra [d,e,*]

[a] National Institute of Technology, Kurukshetra, India
[b] BITS Pilani, India
[c] Curtin University Perth, Australia
[d] Østfold University College, Halden, Norway
[e] Institute of Energy Technology Norway, Norway

ARTICLE INFO

Dataset link: https://github.com/ouniali/WSantipatterns

Keywords:
Data sampling
Feature selection techniques
Anti-pattern
Aggregation measure
Machine learning

ABSTRACT

Software design Anti-pattern is the common feedback to a recurring problem that is ineffective and has a high risk of failure. Early prediction of these Anti-patterns helps reduce the design process's efforts, resources, and costs. In earlier research, static code or Web Service Description Language (WSDL) metrics were used to develop anti-pattern prediction models. These source code metrics are calculated at either file-level or system-level. So, the values of these metrics are frequently dependent on assumptions that are not defined or standardized and might vary depending on the tools available. This study aims to develop a machine learning-based Anti-patterns prediction model using natural language processing techniques for representing the WSDL file as an input. In this research, the four-word embedding methods have been used to process the WSDL file. The processed outputs are used as input to the models trained using thirty-three classifier techniques. This study also uses eight feature selection techniques to remove ineffective features and five data sampling techniques to handle the class imbalance nature of the datasets. The results indicate that the developed models using text metrics perform better than the static code or WSDL metrics. Additionally, the results suggest that selecting features using feature selection and balancing data using sampling techniques helps improve the models' performance.

1. Introduction

An anti-pattern is a common problem-solving strategy that fails to address the root of the issue. Software anti-patterns make the source code challenging to comprehend and retain, creating difficulties in software design that make it difficult to maintain and develop software. Additionally, research revealed that Anti-patterns in web services have been shown to cause problems with maintenance and evolution. Identifying anti-patterns at the design level helps in the reduction of efforts, resources, and costs associated with the design process. As a result, finding anti-patterns is an intriguing topic for research to pursue. This work presents a novel method for detecting anti-patterns in web services using text metrics extracted from the Web Service Description Language (WSDL) file. Based on the assumption that text metrics collected at the web service level are good indicators of anti-patterns, a novel framework is proposed in this research work. Segev et al. [1], investigated two text processing methods that are context analysis for text and term frequency/inverse document frequency (TF-IDF) in their study. The authors examined WSDL files and publicly accessible text descriptors found in service repositories to study the services. The authors established how web service usage could be increased by utilizing data from the web to create a rich context for client queries rather than burdening users with formal concepts and explanations that are unrelated to their services by looking at the WSDL and free text descriptions. In our earlier proposed chapters, object-oriented metrics and WSDL metrics that were extracted from web service description language (WSDL) files as input were used to develop strategies for detecting web service anti-patterns. However, when WSDL or object-oriented metrics are used, then these metrics must be calculated at the file or system level. Even in some situations, it is not always possible to compute metrics, as these metrics' validation and justification are based on empirical or historical data, the reliability of which is difficult to establish. The values of these metrics can change depending on the working environment and the tools available because they are frequently based on undefined or non-standardized assumptions. Computing software metrics can be

* Corresponding author at: Østfold University College, Halden, Norway.
E-mail addresses: LovKumar@nitkkr.ac.in (L. Kumar), p20170433@hyderabad.bits-pilani.ac.in (S. Tummalapalli), sonika.rathi@pilani.bits-pilani.ac.in (S.C. Rathi), bhanu@hyderabad.bits-pilani.ac.in (L.B. Murthy), A.Krishna@curtin.edu.au (A. Krishna), sanjay.misra@ife.no (S. Misra).

https://doi.org/10.1016/j.cola.2023.101207
Received 31 October 2022; Received in revised form 22 February 2023; Accepted 6 April 2023
Available online 20 April 2023

challenging and costly in some situations. In this paper, we have put forth a method for extracting text metrics from WSDL description files using four embedding techniques namely: Term Frequency-Inverse Document Frequency (TFIDF), SKipgram (SKG), GLOVE, and Continuous Bag of Words (CBOW). These metrics are then used as input for building autonomous models that detect the anti-patterns in web services. In this paper, data sampling strategies are used to address the class imbalance that the dataset has been experiencing, along with feature selection techniques to address feature redundancy to optimize further the performance of anti-patterns detecting models. Further, the most frequent classifiers like different variants of naive Bayes, logistic regression, support vector machine with different kernels, and multi-layer perceptron with varying training algorithms have been employed to train the models. Also, an advanced version of machine and ensemble learning like bagging, boosting, etc. are engaged in this research. Finally, to find the pattern for anti-pattern prediction, deep learning with various hidden layers is employed. The performance of these models trained using the above techniques is computed in terms of AUC, accuracy, F-measure, and G-mean values. These experiments are carried out with 226 different web-service and validated using a 5-fold cross-validation approach. This experiment also explored the use of data sampling to handle the class-imbalanced nature of the datasets. So, this work empirically examines the effectiveness of eight feature selection techniques in conjunction with the original features, five data sampling techniques in conjunction with the original data, four-word embedding techniques, and 33 classifier techniques for detecting web service anti-patterns.

1.1. Contributions

The study presented in this work makes the following research contributions:

- The application of word embedding techniques for extracting text metrics from the Web Service Description Language (WSDL) file, which is then utilized to detect anti-patterns in web services, is a significant, innovative contribution of this work.
- Second, in detecting web service anti-patterns, this work demonstrates the use of four-word embedding approaches, eight feature selection strategies, thirty-three classification algorithms, and five data sampling techniques to overcome class imbalance difficulties.
- Another contribution of our work to the body of knowledge on anti-pattern detection in web services is an in-depth empirical investigation of the proposed framework and approaches on a publically available dataset of 226 web services from various fields. A series of experiments were then conducted to investigate the efficacy of various machine learning parameter combinations and their relative efficacy using performance metrics such as AUC, accuracy, and statistical hypothesis-based testing.

1.2. Research objectives

The research aims of this work are presented as follows:

- To empirically examine the performance of metrics produced by utilizing various text embedding techniques on WSDL description files in the detection of web service anti-patterns.
- To investigate the correlation between the text metrics derived from the WSDL description file and the occurrence of anti-patterns.
- To examine the effectiveness of various data sampling techniques used to deal with class imbalance issues, feature selection techniques, and classifier techniques in context with the text metrics produced from the WSDL description files using statistical hypothesis testing and Area Under Curve (AUC) approaches.

1.3. Research questions

The following research questions (RQ) have been developed to identify, evaluate, and summarize the findings of the proposed experimental work:

- Are statistically significant differences between the predictive models trained using extracted sets of features computed with the help of four-word embedding techniques?
- Is there a significant performance difference between models produced using all features and those created using the subset of features chosen using the eight feature selection techniques?
- Is there a statistically significant difference between the performances demonstrated by the five data sampling techniques and those exhibited by the initial data?
- How effective are the 33 learning algorithms in terms of AUC, accuracy, F-measure, and G-mean parameters? Does the expected performance of the 33 classifier techniques differ significantly?

The rest of the paper is arranged as: Section 2 covers a literature review on web-services antipatterns using object-oriented, WSDL metrics, and the derived motivation of work. Section 3 discusses the methods that are considered for carrying out the proposed work. Sections 4 and 5 explain the proposed framework and the results and analysis, respectively. Section 6 focuses on the experimental comparative analysis of completed research and discussion of the result. The conclusion and future scope is highlighted in Section 7.

2. Related work

The presence of web-service Anti-patterns highly influences the maintainability and perception of software systems. Authors working on anti-pattern prediction proposed several methods, but detection and specification of anti-patterns in web services are still not explored too much. Maiga et al. [2] proposed one tool called SVMDetect using a machine learning technique. They used Support Vector Machines with the polynomial kernel to train the SVMDetect models. These trained models help predict anti-patterns like Blob, Swiss Army Knife, Blob, Functional Decomposition, and Spaghetti Code. They have also compared their finding with the DETEX proposed by Moha et al. [3]. Maiga et al. [4] have extended their work and proposed a new SMURF tool based on the same machine learning techniques. However, they have used practitioners' feedback as another component to improve the performance of the models. They have examined more than 300 experiments that help to predict anti-patterns like Blob, Swiss Army Knife, Spaghetti Code, and Functional Decomposition. Finally, they have given proper compression of their proposed method SMURF with DETEX [3], and BDTEX [5]. The benefit of using SMURF is not to define any rule or threshold, i.e., SMURF does not require any manual user to set rules to detect anti-patterns. Their experimental results suggested that SMURF has good performance in terms of recall and precision. Finally, they have concluded that their methods work better after integrating with the user's feedback. Furthermore, various research has been done based on different metrics to address the issue of web service anti-pattern detection.

2.1. Anti-pattern detection using object oriented metrics

There is extensive research done on detecting anti-patterns using object-oriented metrics. Travassos and his team introduced one method based on manual reviews to detect design smells. These smells do not compromise the description of smells [6]. To detect the design smells, Marinescu et al. [7] presented a metric-based approach executed in the IPLASMA tool, with some detection techniques. They have given ten different methods to identify different categories of anti-patterns. The fundamental limitations of these approaches are based on deep knowledge of metric-based rules. Their proposed approaches use metric-based rules to identify different categories of anti-patterns. Similarly, Munro and his team developed one template to characterize different anti-patterns with an objective to overcome the problems of metric-based rules descriptions [8]. They have applied heuristics methods to metrics to predict anti-patterns. Ciupke et al. [9] described one method based on code legacy to identify design problems. They have extracted the occurrences of the problems using models, and these models are developed using source code.

Simon et al. [10] have used visualization concepts to find the relationship between automated approaches. These concepts are time-consuming, productive, and systematic, and no need for manual inspections. Similarly, Rao and his team [11] proposed one model based on Design Propagation Probability Matrices(DCPP). Their developed models use change propagation between the design of artifacts to detect Divergent change and Shotgun surgery anti-patterns. Khomh et al. [5] presented Bayesian Detection Expert (BDTEX), a Goal Question Metric (GQM) approach which builds Bayesian Belief Networks(BBN) based on anti-pattern definition. This approach also allows quality analysts to use their knowledge. Ouni et al. [12] have proposed another automated approach using a cooperative parallel evolutionary algorithm(P-EA) to detect the presence of web service anti-patterns. They have executed different prediction models parallelly and noticed anti-patterns by combining the results of all models. Kumar et al. [13] developed anti-pattern prediction models using extracted features from static source code analysis. They have suggested to apply the concept of different aggregation techniques to find metrics at the WSDL level. Kumar and his team empirically investigated the use of different machine learning algorithms to find the pattern for anti-patterns and different data sampling techniques to overcome the problem of class imbalance. They have also applied different feature selection techniques such as OneR, Gain Ratio, Symmetric Uncertainty, and Information Gain to remove ineffective features.

2.2. Anti-pattern detection technique using WSDL

Rodriguez et al. [14] proposed a tool for the automatic detection of anti-patterns while creating the web service descriptions. In this paper, the author has proposed heuristics to detect complex anti-patterns and algorithms to detect simple anti-patterns. Natural Language Processing (NLP) and Machine Learning algorithms defined the proposed heuristics and algorithms. The developed tool takes the web service description language (WSDL) file as input and lists all the detected anti-patterns and their occurrences as output. Furthermore, the author has validated the proposed algorithm's effectiveness with a publicly available dataset. Veliouglu and his group [15] proposed Y-CSD tool based on structural analysis to predict the presence of code smells and anti-patterns. They observed that their proposed tools help to reduce the presence of anti-patterns. Y-CSD tool to predict two different categories of anti-patterns, such as Data class and Brain Method. They have also planned to develop a new version of the tool that helps to locate code smells and anti-patterns. Researchers also observed that it is possible to do method call analysis utilizing Call-based Dependence between Methods (CDM) [16], Information flow-based Coupling (ICP) [17], Message Passing Coupling (MCP) [18] and Coupling Between Object classes (CBO) [18]. The CDM method is the most extensively employed for extracting code structure features. It is a symbol of the integration of several ways. A variety of studies have used it as a way to detect features like feature envy [19], blob [20], denied bequest [21–23], blob [24], type checking [25], divergent change [11], and shotgun surgery [11]. Shared instance-based analysis may be carried out either by determining structural similarity between methods (SSM) [26] or by using lack of method cohesion (LCOM) [27].

2.3. Anti-pattern detection technique using word embedding and mining techniques

Anti-patterns are often discovered by examining the code's structural properties. Structural information determines if a given method should be moved to another class since it utilizes more characteristics of a particular class. Move Method Refactoring (MMR) is used by JDeodorant [25] to eliminate feature envy from programs. Type checking and blob anti-patterns may also be detected using JDeodorant [21,23,25]. The abstract syntax tree (AST) may also be used to find similarities in syntax or structure [28]. To identify feature envy, the AST is scanned to compute the referenced classes for each field. Many ways have been developed to discover syntactic or structural similarities using AST to identify duplicate or clone code [29,30]. Latent semantic indexing (LSI) [31,32] is used to extract lexical properties by searching for textual similarities. Lexical characteristics analysis has been used to identify features such as feature envy [19] and blob [16]. Software repositories are analyzed to obtain historical features using a variety of mining approaches. Borovits and his team [33] developed one model to predict linguistic anti-patterns. They have applied word embeddings on the abstract syntax tree of Iac code and deep learning techniques to train their models. Alshraiedh et al. [34] pointed out that there is a chance of anti-patterns in RESTful web services. They have observed that many RESTful web services have the problem of anti-patterns. The presence of these anti-patterns may diminish the sustainability of the services. Table 1 shows a study of the web service anti-pattern detection techniques proposed over the years.

2.4. Motivation for the study

The conducted literature survey represents that the authors have used different approaches like object-oriented, WSDL metrics, and some rule-based approaches to detect the presence of anti-patterns. These metrics are calculated at either the file level or the system level. So, the values of these metrics are frequently dependent on assumptions that are not defined or standardized and might vary depending on the tools available and the working environment. Additionally, the authors of the earlier study engaged simple machine learning techniques like Naive Bayes, and support vector machines. Further investigation reveals very few studies of text-based approaches for anti-pattern prediction using WSDL files. It leads us to endow our focus on implementing the proposed model to address the substantial gap identified to extemporize the performance and predictability of the anti-pattern prediction model by engaging different sets of word-embedding, sampling, and feature selection techniques jointly with a wide variety of machine learning techniques. This research exploits the implication of four different categories of word embedding, five sampling strategies, seven feature selection techniques, and a huge variety of machine learning techniques to develop the best automated models for finding the presence of web service anti-patterns. The performance of these developed models is analyzed using AUC, accuracy, F-measure, and G-mean metrics.

3. Used methodologies in our experimentation

This section enlightens on the components required for our study. In order to overcome the paper's size restriction, this section provides information on datasets, feature selection algorithms, sampling techniques, and classification approaches in a concise form. The references cited are available to the readers for a detailed description.

3.1. Word embedding techniques for the generation of text metrics

This research makes use of four embedding techniques to extract numeric features from the WSDL file. These extracted numeric features are used as input for anti-pattern prediction models. In the proposed study all four embedding techniques have been employed on each WSDL file and extracted numeric features. A detailed explanation of the above techniques is given below:

Table 1
SOA Anti-pattern detection techniques.

Authors	Method Used	Technique	Anti-Patterns Detected	Metrics Used	Year
Alshraiedh et al. [34]	Based on the URIs parsing process	Extraction of code structure features using Call-based Dependence between Methods (CDM)	feature envy, blob, denied bequest, blob, type checking, divergent change, and shotgun surgery	Method calls, shared variable instances, and inherited variables	2020
Borovits et al. [33]	abstract syntax tree for linguistic anti-pattern detection	Iac code units to create their code embedments	linguistic anti-patterns	Infrastructure as code (Iac) scripts	2020
Fatima Sabir et al. [35]	Scalable approach for the detection of anti-patterns using anti-pattern detection engine	Rules generated by Enterprise Architect data model	GOWS, DWS, FGWS, AN, Dup-WS, LCO, RPT, CWS, CRUD, LGWS	Data model of Enterprise architect	2017
Wang, H. et al. [36]	Web Service Refactoring opportunities as Multi-objective problems Identification	Multi-Objective Genetic Programming(MOGP)	MRPC, CRUDYI, DS, AN, FG, GOWS.	Web service interface metrics and Web service code metrics	2016
YUGOV, A [37]	Detection of anti-patterns by using graph models	Metric based approach	GOWS	Incoming call rate, Outcoming call rate, response time, number of service connections, cohesion with other services.	2016
Wang, H.et al. [38]	Web Service Evolution prediction	Artificial Neural Network(ANN) to predict anti-patterns in future releases	DS, MS, NS, CS		2016
Ouni, A. et al. [39]	Cooperative parallel evolutionary algorithm(P-EA) for detecting anti-patterns in web-services	Parallel Evolutionary Algorithms	MRPC, CRUDYI, DS, AN, FGWS, GOWS	Interface-level (WSDL) metrics, Dynamic metrics, Code-level metrics.	2015
Ouni, A et al. [40]	Genetic programming approach that is based on combination of metrics and threshold values	Genetic Programming	MS, DS, NS, AN	Web service metrics	2015
Coscia et al. [41]	Java to WSDL Mapper	Meta Programming and Text Mining	EDM, RPT, WET, AN, UFI, IC, ISM, LCOP	Object-oriented metrics from the code-implementing services	2014
Palma, F. et al. [42]	Service Oriented Detection for Anti-patterns in Web services (SODA-W) , an approach supported by a framework for specifying and detecting anti-patterns in WSs	Static and dynamic analysis for source code metrics	GOWS, FGWS, AN, CS, RPT, LCO	Static and Dynamic metrics	2014
Palma, F. et al. [43]	Specification and Detection of anti-patterns using a set of metrics by using a rule-based approach	SOFA	MS, DS, NS	Static and Dynamic Metrics	2013
Nayrolles et al. [44]	Service Oriented Mining for Anti-pattern Detection(SOMAD)-detection by mining execution traces.	Association Rule Mining	MS, DS, CS, Kt, BNS, SC	NMA, NDP, NM, COH, CID, IC, OC, TC	2013
Rodriguez et al. [45]	Development of web services with comprehensive guide lines along with tool support	Detecting violation of rules in WSDL by using EASY SOC	EDM, RPT, WET, AN, UFI, IC, ISM, LCOP	SLOC (Source Lines Of Code), Ce (Efferent Coupling), CBO (Coupling Between Objects) and RFC (Response For Class)	2013
Mateos et al. [46]	Detecting WSDL based services using EasySOC tool.	Text Mining, Component based software engineering and Machine learning	WSDL based Services	Object-oriented metrics from the code-implementing services	2013
Coscia et al. [47]	correlation analysis between source code metrics and WSDL implementation code	Detection of anti-patterns using statistical analysis	LCOP, ISM, UFI, IC, AN,RPT, WET, EDM	Object-Oriented metrics	2011
Rodriguez et al. [48]	Effective service availability by the improvement of WSDL document	Discoverability and removal of anti-patterns	LCOP, ISM, UFI, IC, AN, WET, RPT, EDM		2010

Table 2
Data-sets.

Anti-pattern	NAP	AP	%AP
CWS	205	21	9.29
FGWS	213	13	5.75
AWS	202	24	10.62
GOWS	205	21	9.29
DWS	212	14	6.19

- ***Term frequency-Inverse Document Frequency (TFIDF):*** The concept of TFIDF [49,50] is based on statistics to find the importance level of the word in a collection of documents. It is computed based on the word frequency in a document and across a set of documents.

$$TF(w) = \frac{\text{(Frequency of W in File F)}}{\text{(Number of Unique Words)}}$$
$$IDF(w) = \log_e\left(\frac{\text{Number of documents}}{\text{Number of Documents containing W}}\right) \quad (1)$$

- ***Continuous Bag Of Words (CBOW):*** The concept of CBOW [51, 51] is based on combining surrounding words to estimate the word present in the middle. It will take less time to train than skip-gram and provide better performance for frequent words.
- ***SKip Gram (SKG):*** The concept of SKG [52,53] is based on predicting the context based on the word. It will provide better performance for the training data with fewer samples. It will also provide the best representation of rare words.
- ***Global Vectors for Word Representation (GLOVE):*** The concept of GLOVE [54] is based on an unsupervised learning algorithm. It will use this algorithm to compute numerical vector representations for words. The GLOVE is trained on statistics of aggregated global word–word from a corpus. The final output of GLOVE represents the linear substructures of the word vector space.

3.2. Experimental dataset

This experiment makes the use of publicly available web-services datasets consisting of 226 WSDL files shared by Ouni et al. on GitHub.[1] Table 2 shows a detailed description of the considered datasets in terms of different types of anti-patterns. The first column of the table contains the name of anti-patterns like Fine-Grained anti-pattern (FGWS), Chatty anti-pattern (CSW), God Object anti-pattern (GOWS), Data antpattern (DWS), and Ambiguous Anti-pattern (AWS). The second column contains the number web-service not having these patterns, the third column contains the number web-service having these patterns, and the last column contains the percentage of web-service having these patterns. Table 2, indicates that the 13 web-service has FGWS anti-pattern with 5.75%.

3.3. Data sampling techniques

The number of samples for AP and ANP are not equal, as shown in Table 2 and this indicates that the datasets under consideration have a class imbalance problem. Different sampling techniques have been used in this study to address this issue [55]. The technique used to tackle the class imbalance problem of the considered datasets are Synthetic Minority Oversampling Technique (SMOTE), SVMSMOTE, Adaptive Synthetic Sampling Technique (ADASYN), Borderline SMOTE (BLSMOTE), and UP sampling Technique (UPSAM). A brief overview of each of these techniques is given in Table 3.

[1] https://github.com/ouniali/WSantipatterns

Table 3
Data sampling techniques.

Technique	Description
SMOTE	Synthetic Minority Oversampling Technique [56], is based on the concept of nearest neighbors. SMOTE generates samples as a convex combination of two samples. One sample is generated using k nearest neighbors concepts, and another sample is selected as a line segment in the feature space.
BLSMOTE	Borderline SMOTE is based on the concept of selecting new samples of the minority class [57]. These samples are the closest neighbors of the border region between classes.
SVMSMOTE	Support vector machine SMOTE generates new samples of minority classes over the border with SVM to establish a boundary line between the classes using SVM [58].
ADASYN	Adaptive Synthetic Sampling Technique concept uses the weighted distribution of minority samples based on their difficulty level during the training of models. It will generate more synthetic data for minority levels that are difficult to learn as compared to the sample that is easier to learn [59].
UPSAM	Upsampling Method concept randomly generates samples for minority levels in the range of 95 confidence interval of samples [60].

3.4. Feature selection techniques

The authors working in the area of data science revealed that the prediction ability of the classifiers depends on the selection of relevant features. The presence of redundant or ineffective features not only reduces the prediction's ability but also improves the models' complexity. Therefore, it is compulsory to remove unusable features using feature selection techniques before training the models. To find significant sets of effective features this study has applied Significant Features (SIGF) obtained by applying the Wilcoxon sign test, Information Gain (INFG), Gain Ratio (GNR), Correlation coefficient (CORR), CFS subset evaluator (CFS), OneR, Genetic Algorithm (GA), and Principal Component Analysis (PCA). The extracted effective sets of features are taken as input to design the anti-pattern prediction models. The study also compared these models using the original set of features. The details of these techniques are mentioned in Table 4.

3.5. Classifier techniques

To develop models for detecting web service anti-patterns, many classifier techniques have been used in the proposed study. The employed general classifier techniques are Support Vector Classifier (linear (SVC-LIN), radial bias (SVC-RBF), and polynomial kernel (SVC-POLY) kernel), decision tree (DT), Naive Bayes (Gaussian (GNB), Bernoulli (BNB), Multinomial (MNB)), and Logistic Regression Analysis (LOGR). Further, the deep layer technique with different architectures (1-hidden layer (DL1), 2-hidden layer (DL2), 3-hidden layer (DL3), 4-hidden layer (DL4)), advanced learning classifiers such as extreme learning machine (ELM) and weighted extreme learning machine (WELM). The ELM algorithm is implemented with three kernels, namely: Linear (ELM-LIN), Radial Bias (ELM-RBF), and Poly (ELM-POLY), whereas the WELM algorithm is implemented with four kernels, namely: Sigmoid (WELM-SIG), Radbas (WELM-RBS), Tribas (WELM-TRBS), and Sine (WELM-SIN). Furthermore, ensemble classifiers such as Bagging Classifier (BAG), Random Forest Classifier (RF), Extra Trees Classifier (EXTR), AdaBoost Classifier (AdaB), Gradient Boosting Classifier (GraB) are used for training the models. All these techniques help to find import patterns to detect web services' anti-patterns. This study has adopted the concept of 5-fold cross-validation to validate these models. The details of these techniques are mentioned in Table 5

Table 4
Feature selection techniques.

Technique	Description
SIGF	Significant Features (SIGF) are computed using the rank-sum test. Initially, we computed the metrics value of the WSDL file having anti-pattern and not having anti-pattern. Then, we applied the rank-sum test at a 0.05 significance level to find the significant difference between the metrics values of WSDL with anti-pattern and WSDL without anti-pattern [61,62]. Finally, we have selected only those metrics that have the ability to differentiate between WSDL files having an anti-pattern present or not.
INFG	Information Gain (INFG) is a parameter to rank features. It is calculated as a difference between the parent node's entropy and the chile nodes' mean entropy [63,64]. Initially, we used information gain to rank all features, and then we selected $log_2 n$ number of features for analysis.
GNR	Gain Ratio (GNR) is also one parameter to rank features like information gain [65,66]. It is calculated as a ratio of information gain of features with the value of Split Information. The value of gain ratio also depends on the number and size of branches. Here, we have also selected the top $log_2 n$ number of features for analysis.
OneR	OneR attribute is mostly used to predict the label of the data sample by finding the most frequent labels for the feature values [67,68]. It is also called one Rule, which helps to classify different labels with high performance based on single features. Here, we have used OneR to find important features, and then we selected $log_2 n$ number of features for analysis.
PCA	Principle Component Analysis(PCA) is used to find new values of the feature that have better variances than the original features [69,70]. It is based on the concept of removing highly correlated features and selecting new values of features. In this work, we have considered all the Principle Components whose eigenvalue is greater than 1.
CORRS	Correlation Coefficient (CORRS) feature selection uses Pearson's correlation value to find the highly correlated features and select new sets of un-correlated features [71,72]. Here, we have defined highly correlated features as Pearson's correlation value >= 0.7 or <=-0.7.
CFS	Classifier Subset Evaluation (CFS) concept uses one classification technique to find the best sets of low inter-correlation features with a better prediction ability [73,74].
GA	Genetic Algorithm(GA) helps to search for the best set of features that can improve the performance of the models. The concept is based on fitness value [75,76]. Here, we have used fitness values that contain multiple objective functions. The objective of using GA is to find the minimum number of features with a better prediction ability. The fitness function for each chromosome is calculated using below equation: $$0.8 * AUC + 0.2 * \frac{No.of\,Feature - Selected\,No.of\,Features}{No.of\,Feature} \quad (2)$$ In this work, we have given 80 weightage to AUC and 20 weightage to a number of features.

4. Proposed framework

The work in this paper present the empirical framework to predict anti-pattern present in web-service using natural language processing techniques for representing the characters and words in the WSDL file as an input. The proposed framework for the experimental work is shown in Fig. 1. It is a multi-step procedure, and each step is discussed below in detail.

- **STEP 1:** The proposed framework uses WSDL description files gathered from various disciplines, including weather, education, finance, etc., to find and validate the extracted patterns for Anti-Pattern prediction. These pattern help to identify five anti-patterns like Ambiguous Anti-pattern (AWS), God Object anti-pattern (GOWS), Chatty anti-pattern (CSW), Fine-Grained anti-pattern (FGWS), Data ant-pattern (DWS) as shown in Table 2.
- **STEP 2:** In the initial step, text metrics are generated from the WSDL file by using a word embedding approach. In this step, a numerical representation of the WSDL file has been found using the embedding techniques Continuous Bag of Words (CBOW), Term Frequency-Inverse Document Frequency (Tf-IDF), Skip Gram (SKG), and Global Vectors for Word Representation (GLOVE). The considered embedding approach generates a numeric vector of size 450 to 1200 for each WSDL file. These computed numeric vectors are used as input for the proposed prediction models.
- **STEP 3:** Since the computed numeric vectors are used as input, so there is a possibility that some of these vector components will be ineffective in identifying anti-patterns. Eight feature selection algorithms have therefore been applied in this step to identify the redundant features. By removing extraneous redundant vector components, this step aids in reducing the complexity of designed models. Additionally, the Min–max normalization has been applied to scale feature values.
- **STEP 4:** The information present in Section 3.2 suggested that the considered datasets for the experiment suffer from an issue of class imbalance. Therefore, the applied five data sampling strategies to solve the problem of class imbalance are: the Synthetic Minority Oversampling Technique (SMOTE), SVMSMOTE, Borderline SMOTE (BLSMOTE), Upsampling (UPSAM), and the Adaptive Synthetic oversampling technique (ADASYN).
- **STEP 5:** In this step, the proposed study constructs models for the prediction of web service anti-patterns using a variety of classifiers, including ensemble classifiers like AdaBoost Classifier (AdaB), Random Forest Classifier (RF), Bagging Classifier (BAG), Extra Trees Classifier (EXTR), and Gradient Boosting Classifier (GraB) as well as Naive Bayes with different variants, different variants of Support Vector Classifier with various kernels, Multilayer perceptron with varying algorithms of training, different types of Extreme Learning Machine and deep learning technique with a varying number of hidden layers. The generated models are validated using a 5-fold cross-validation approach.
- **STEP 6:** The last step of the framework is to suggest the best combination of embedding technique, feature selection technique, data sampling technique, and classification techniques to find the presence of different anti-patterns in web services. The final suggestion for anti-pattern prediction is based on calculated AUC, accuracy, F-measure, and G-mean of testing data. The proposed framework also uses the Friedman Test with the Wilcoxon Signed Rank Test to find the significant impact on the performance of the models after using the specific technique.

5. Results and analysis

The primary aim of this research is to validate the text vectors generated using the word embedding techniques for finding the presence of an anti-pattern in the WSDL file. The initial step is to apply different categories of embedding techniques to find the numerical vector of the WSDL file. The TFIDF, CBOW, SKG, and GLOVE techniques are used for this objective to find numerical vectors. Following the discovery of numerical vectors, the study employs a variety of techniques to eliminate ineffective features and sampling techniques to address the class imbalance problem datasets. Finally, to achieve the best models for anti-pattern predictions, the proposed research employs thirty-three different types of machine learning algorithms. The prediction capability for each developed model is validated using 5-fold cross-validation, and their performances are analyzed in terms of AUC, accuracy, F-measure, and G-mean. The experiments were conducted using a core-i3 processor, 64 GB of RAM, and python libraries like sklearn, numpy, pandas, and keras, as well as Matlab 2021 for visualization. The outcome and analysis of the feature selection and classification techniques are presented in this section.

Table 5
Classification technique.

Classifiers	Interpretation
Naive Bayes Algorithm (NB)	Noteworthy for its multi-class prediction. We can predict the likelihood of various classes of target variables using this algorithm [77,78]. To create models for predicting web service anti-patterns, we use three variations of naive Bayes algorithms in this study: Gaussian, Bernoulli, and Multinomial.
Decision Tree (DT)	It is a decision model that uses a number of independent variables to represent the estimate of a target variable [79,80].
Logistic Regression Analysis (LOGR)	LOGR is used to compute the probability of sample belong to specific class based on one or more independent variables.
Support Vector Classifier (SVC)	It is an excellent option for creating a classification model as it operates as a non-probabilistic binary linear classifier by categorizing input data into one of two classes. In this work, SVC with three different kernels, namely linear (SVC-L), polynomial (SVC-P), and radial (SVC-R), are employed to train models [81,82].
Least Square Support Vector Machine (LSSVM)	The squared sum of errors to the objective functions is minimized by this algorithm. These supervised learning techniques examine the data to find patterns. In the proposed work the models are trained using LSSVM with linear (LSSVM-Lin), polynomial (LSSVM-Poly), and radial basis function (LSSVM-RBF). [83,84].
Extreme Learning Machine (ELM)	ELM is one of the effective neural network with single hidden layer. Feed-forward neural networks with a single hidden layer is the learning process of this algorithm. The core concept behind this method is the random generation of hidden nodes, wherein hidden node parameters are distributed independently of training samples. ELM was used to train the anti-pattern detection models using radial basis functions (ELM-RBF), polynomial basis functions (ELM-Poly), and linear basis functions (ELM-Lin) in this work [85,86].
Weighted Extreme Learning Machine (WELM)	This method gives the minority class more weight when dealing with unbalanced data and less weight to the majority class. Based on the class distribution, WELM chooses a weighting scheme, and the weights produced are inversely proportional to the number of samples in the training set. To further increase WELM's speed, we implemented four different kernel functions (Sigmoid, Radbas, Tribes, and Sine) [87,88].
Multi-Layer Perceptron (MLP)	A set of features and a target can be used by MLP to train a non-linear function approximator for classification or regression. As there can be one or more non-linear layers, known as hidden layers, between the input and output layers, it differs from logistic regression [89].
MLP with Stochastic Gradient Descent (MLP-SGD)	When using MLP, it is necessary to update the weights to lower output error [90]. SGD is used in this work for this purpose. By taking the total error function's first order derivative, the SGD method locates the minima in error space.
MLP with Quasi-Newton Method (MLP-LNF)	This technique requires the computation of the 2nd order derivatives of the total error function for each component of the gradient vector. It is a quick optimization method that can be utilized instead of conjugate gradient techniques [91].
MLP with Stochastic Gradient with Adaptive Learning Rate Method (MLP-ADAM)	The training process will take too long to converge in case of the sample size is too small. The best learning rate (α) before training can theoretically be predicted, but it is practically impossible to predict the value of changes as training progresses. As a result, ADAM is used in this study to train the prediction model [91].
K-Nearest Neighbour (KNN)	KNN is based on the nearest neighbors principle. To determine the results of the samples, it will employ the idea of vote methods. If the majority of the neighboring samples are of class A, then the sample will be categorized as A [92].
Bagging Classifier (BAG)	It is an ensemble meta-assessment tool that applies base classifiers to individual subjective subsets of the underlying dataset, aggregates remote predictions made either by voting or averaging them, and then creates the final prediction [93].
Random Forest Classifier (RF)	This algorithm creates decision trees on samples of data, waits for each one to deliver its prediction, and then uses the voting method to select the best arrangement [94].
Extra Trees Classifier (EXTR)	This algorithm creates a meta assessor that fits various randomized decision trees or additional trees on various sub-samples of the dataset and makes use of averaging to improve predictive accuracy while controlling over-fitting [95].
AdaBoost Classifier (AdaB)	It is a meta-estimator that begins by fitting a classifier to the initial dataset and then fits more copies of the classifier to the subsequent datasets. However, the weights of instances that are incorrectly classified are changed with the ultimate goal of making subsequent classifiers focus more on problematic cases [95].
Gradient Boosting Classifier (GraB)	The GraB classifier's philosophy is to minimize the difference between the predicted class esteem and the actual class estimation of the training instance. It makes it easier to build an additive model in a forward stage-by-stage manner [95].
Deep Learning Technique (DL)	Artificial neural networks, a type of machine learning that depends on the composition and functionality of the human brain, are used in deep learning. This algorithm trains the machines by using various examples from the dataset or pertinent examples [96,97]. An ANN has several advantages over other kinds of algorithms, but its unique information processing architecture is its main advantage. In this work, we used the Deep Learning (DL) technique with four different hidden layer: DL1 (hidden layer), DL2 (hidden layer), and DL3 (hidden layer) (DL4).

5.1. Feature selection techniques

The use of feature selection techniques to identify ineffective features is one important factor in the performance of the models. The presence of ineffective or irrelevant features decreases the models' prediction capability and increases the models' complexity. This research work uses different feature selection techniques to remove ineffective features and provide the best sets of features. Table 6 shows the

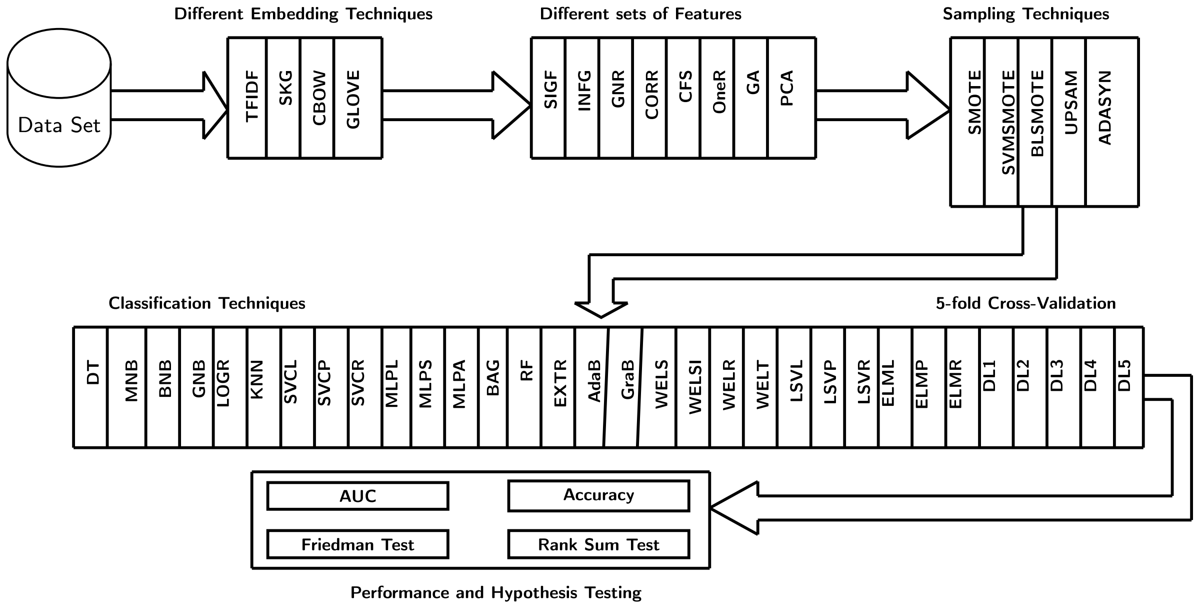


Fig. 1. WSDL text based anti-pattern prediction.

Table 6
Number of selected features: Different types feature selection techniques.

	AP1			AP2			AP3			AP4			AP5		
	CBOW	SKG	GLOVE	CBOW	SKG	GLOVE	CBOW	SKG	GLOVE	CBOW	SKG	GLOVE	CBOW	SKG	GLOVE
AF	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300
SIGF	228	208	149	15	24	80	211	197	146	235	214	157	34	36	18
INFOG	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
GNR	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
CFS	4	6	14	4	3	13	14	16	23	17	21	24	4	4	4
CORR	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
OneR	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
GA	118	121	128	116	127	121	119	120	121	121	122	124	127	125	127
PCA	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8

number of selected features after applying different feature selection techniques. The rows of Table 6 present the techniques used for feature selection, and the column represents the embedding with anti-patterns. From Table 6, it can be inferred that 228 of the 300 features computed using CBOW can distinguish whether a WSDL contains an anti-pattern or not. Similarly, the number of significant features using SIGF for AP2 on CBOW embedding have 15. The results demonstrate that, when using other feature selection techniques, the number of selected features is significantly less than the number of original features. The information of Table 6 also suggested that the number of selected features depends on feature selection techniques.

5.2. Performance value

This study uses text metrics extracted from the WSDL files using NLP techniques as input for web-service anti-pattern predictions. These models are developed using selected relevant vectors after applying eight feature selection techniques and the original text metrics. In this work, different variants of sampling techniques, like SMOTE, UPSAMPLING, BLSMOTE, etc., have been used to handle the class-imbalanced nature of datasets. Finally, a variety of classifier methods are employed including Naive Bayes with different variants, different variants of Support Vector Classifier with different kernels, different types of extreme learning machines, Multilayer perceptron with varying algorithms of training, ensemble classifiers such as AdaBoost Classifier (AdaB), Random Forest Classifier (RF), Bagging Classifier (BAG), Extra Trees Classifier (EXTR), Gradient Boosting Classifier (GraB), and deep learning technique with a varying number of hidden layers. The generated models are validated using a 5-fold cross-validation approach. The final models developed using text metrics with feature selection, data sampling, and machine learning techniques were used to predict five types of anti-patterns present in web services based on WSDL metrics. The prediction capability of these models for AP1 is measured in terms of accuracy, F-score, G-mean, and AUC, as mentioned in Tables 7 and 8. The F-score, and G-mean for other combinations are mentioned in appendix section (Tables 19 to 22). The first section of Tables 7 and 8 present the accuracy of the models with different variants of embedding techniques trained on original data and sampled data using SMOTE. Similarly, second section of Tables present the AUC of the models with different variants of embedding techniques trained on original data and sampled data using SMOTE. The bold green color of Tables represent the model's performance. The rows of both tables are used to represent the classifiers used to train the models, i.e., the models trained using MNB classifier on all features of TFIDF embedding with original data achieved 91.41% of accuracy, 0 fscore, 0 gmean, and 0.61 AUC. Similarly, the model trained using the same MNB on balanced data with selected relevant input features achieved the best performance with 0.85 AUC values. The LSSVM-RBF with SVM SMOTE sampling approach, as shown in Table 22, yields better results for all combinations of employed feature selection and embedding techniques, with F-score values consistently greater than 0.95. Tables 19 and 20 represent RF, and EXTR achieves higher G-mean values after LSSVM-RBF. The information in Table 7 suggested that the models produced using data sampling approaches outperformed by earning high G-mean, F-score, and AUC values than models developed using the original data, indicating that data sampling techniques are crucial in constructing models for finding the presence of anti-patterns. The information also

Table 7
Accuracy and AUC for Anti-pattern 1: Different types of text features.

	ORGD									SMOTE								
	TFIDF			CBOW			SKG			TFIDF			CBOW			SKG		
	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA
Accuracy																		
MNB	91.41	91.41	91.41	62.12	62.63	70.20	61.62	62.63	72.22	70.99	72.93	71.82	77.07	77.07	77.35	76.52	77.35	76.24
BNB	86.36	91.41	89.90	87.37	87.88	86.87	86.87	86.87	88.38	84.81	82.04	85.91	51.66	50.83	51.66	51.11	51.66	50.55
GNB	65.15	78.79	83.84	66.16	65.15	65.66	65.66	64.14	63.13	77.62	72.10	77.35	81.77	80.94	82.04	81.22	80.11	80.94
DT	89.90	88.89	87.37	86.36	85.86	86.36	88.38	85.86	84.34	88.67	81.49	83.15	91.44	94.2	88.95	93.65	92.54	92.54
LOGR	91.41	91.41	91.41	91.41	90.91	91.41	91.41	90.91	91.41	81.49	78.45	78.18	77.62	80.11	78.73	83.15	82.32	81.22
KNN	90.40	90.91	90.40	89.90	89.90	89.39	91.92	90.40	90.91	89.50	80.66	86.74	87.57	87.29	86.74	89.23	88.67	88.95
SVCL	79.29	79.29	80.81	67.17	66.16	64.65	77.78	73.23	69.70	75.97	61.60	68.23	70.44	70.99	72.38	73.20	66.02	72.93
SVCP	86.36	84.34	86.87	79.80	76.26	78.28	84.34	83.84	83.84	78.18	57.18	71.27	74.86	74.03	75.97	76.80	67.96	77.62
SVCR	81.82	81.82	81.31	71.72	71.21	69.70	72.73	70.20	72.73	78.73	63.26	71.27	71.55	71.55	74.03	71.55	63.54	73.48
MLPL	91.92	88.89	88.38	91.41	90.91	83.84	88.89	91.41	91.41	89.78	79.01	81.22	45.86	86.74	60.22	46.96	94.48	45.86
MLPS	91.41	91.41	91.41	91.41	91.41	91.41	91.41	91.41	91.41	83.43	49.45	80.11	83.15	82.32	79.83	50.00	83.43	50.00
MLPA	90.91	91.41	90.91	91.41	91.41	89.39	91.41	91.41	91.41	91.71	79.83	84.53	50.00	50.00	88.95	50.00	91.99	50.00
BAG	91.41	91.41	91.41	91.41	91.41	91.41	90.91	91.41	90.91	86.46	81.22	82.60	83.15	83.98	84.53	85.91	86.46	85.36
RF	89.90	91.41	90.40	89.90	89.90	88.89	91.41	89.39	91.41	90.33	82.87	90.06	92.27	92.54	91.99	94.20	94.48	94.48
EXTR	90.91	91.41	90.91	89.90	89.90	89.90	90.91	90.91	90.91	90.88	80.94	90.06	93.65	92.54	91.44	95.86	95.03	94.75
ADA	91.92	92.42	91.41	87.88	89.90	83.33	90.40	88.89	91.41	88.40	84.25	84.25	90.06	92.82	87.57	93.92	93.09	93.65
GRAB	91.41	92.42	92.42	85.86	87.37	83.33	85.86	87.37	85.86	85.91	84.81	86.74	92.54	94.2	85.64	93.09	92.27	93.37
WELS	64.14	76.77	64.65	67.68	65.66	62.12	64.65	64.65	62.12	74.86	75.41	75.41	79.56	79.28	77.62	79.56	77.90	77.62
WELSI	73.74	76.77	69.70	81.31	75.25	70.20	82.83	80.30	74.75	78.73	77.90	78.73	87.85	84.81	82.60	90.61	87.57	83.15
WELR	78.28	82.32	71.72	75.76	73.74	65.15	79.29	76.26	70.20	78.18	77.90	79.56	83.70	83.15	81.77	85.36	81.77	80.66
WELT	74.24	79.80	71.21	76.77	76.26	69.70	81.31	72.73	70.71	78.18	77.35	76.24	84.81	81.77	82.87	84.53	82.60	82.04
LSL	91.41	91.41	90.91	91.41	91.41	91.41	91.41	91.41	91.41	83.70	77.90	78.73	100	100	99.72	98.34	100	93.65
LSP	92.93	93.94	93.94	100	100	96.46	100	100	98.48	94.75	82.87	88.40	99.72	99.72	100	100	100	99.45
LSR	92.42	91.41	92.42	97.98	100	100	100	97.98	100	98.62	88.40	99.45	100	99.72	100	100	100	100
ELL	91.41	90.91	91.41	91.41	91.41	91.41	91.41	91.41	91.41	77.90	79.56	71.27	79.83	79.28	77.62	80.66	78.73	79.01
ELR	91.41	91.41	91.41	90.40	89.90	91.41	90.40	90.40	91.41	80.66	78.73	79.01	85.36	83.98	82.60	85.08	83.70	81.77
ELP	90.91	90.91	91.41	88.38	88.89	89.90	87.88	89.39	89.39	88.67	78.73	81.49	90.88	88.95	84.81	92.27	92.54	89.78
DL1	91.41	91.41	91.41	91.41	90.91	91.41	91.41	91.41	91.41	81.49	79.01	78.73	80.94	82.04	81.77	86.19	85.08	82.32
DL2	91.41	91.41	91.41	91.41	91.41	91.41	91.41	91.41	91.41	83.70	68.23	79.01	81.77	82.60	82.60	88.67	88.40	85.64
DL3	91.41	91.41	91.41	91.41	91.41	91.41	91.41	91.41	91.41	85.36	60.50	79.83	82.87	80.94	83.43	88.40	85.36	86.74
DL4	91.41	91.41	91.41	91.41	91.41	91.41	91.41	91.41	91.41	84.81	52.76	66.85	81.49	81.77	81.77	87.85	87.57	86.46
DL5	90.40	91.41	91.41	90.91	90.91	91.41	90.91	90.91	88.89	79.01	53.59	62.98	87.02	85.08	86.19	83.98	88.12	88.12
DL6	90.91	91.41	91.41	90.91	89.90	90.40	89.90	89.39	89.39	57.73	61.88	61.60	87.57	84.25	84.25	87.57	90.33	86.70
AUC																		
MNB	0.61	0.65	0.66	0.77	0.78	0.74	0.79	0.77	0.79	0.78	0.84	0.81	0.81	0.82	0.82	0.84	0.85	0.85
BNB	0.69	0.75	0.72	0.50	0.50	0.36	0.49	0.38	0.45	0.9	0.87	0.89	0.47	0.49	0.50	0.50	0.50	0.48
GNB	0.60	0.55	0.70	0.74	0.73	0.75	0.74	0.73	0.71	0.83	0.82	0.84	0.82	0.81	0.83	0.81	0.81	0.82
DT	0.62	0.62	0.56	0.55	0.63	0.55	0.62	0.52	0.59	0.88	0.80	0.82	0.91	0.94	0.89	0.94	0.93	0.93
LOGR	0.70	0.73	0.70	0.70	0.72	0.66	0.75	0.73	0.75	0.89	0.83	0.87	0.84	0.84	0.83	0.9	0.9	0.88
KNN	0.68	0.63	0.66	0.68	0.71	0.72	0.74	0.78	0.76	0.95	0.88	0.93	0.94	0.93	0.93	0.95	0.95	0.94
SVCL	0.74	0.68	0.72	0.45	0.56	0.58	0.60	0.54	0.74	0.56	0.53	0.61	0.58	0.59	0.57	0.62	0.51	0.60
SVCP	0.76	0.61	0.71	0.52	0.63	0.67	0.68	0.70	0.77	0.59	0.52	0.62	0.61	0.62	0.62	0.63	0.52	0.62
SVCR	0.73	0.75	0.74	0.71	0.75	0.72	0.79	0.8	0.79	0.58	0.53	0.63	0.61	0.61	0.61	0.63	0.52	0.62
MLPL	0.74	0.70	0.51	0.66	0.74	0.67	0.72	0.38	0.39	0.93	0.82	0.91	0.45	0.88	0.70	0.46	0.97	0.44
MLPS	0.42	0.24	0.48	0.70	0.76	0.66	0.72	0.38	0.39	0.91	0.59	0.87	0.86	0.87	0.83	0.46	0.90	0.44
MLPA	0.71	0.50	0.75	0.69	0.72	0.73	0.80	0.68	0.49	0.96	0.83	0.90	0.45	0.83	0.90	0.46	0.95	0.44
BAG	0.65	0.73	0.63	0.70	0.79	0.69	0.80	0.81	0.80	0.93	0.89	0.90	0.93	0.92	0.93	0.96	0.95	0.94
RF	0.68	0.69	0.77	0.67	0.82	0.68	0.77	0.78	0.72	0.96	0.88	0.94	0.98	0.98	0.97	0.99	0.98	0.98
EXTR	0.73	0.71	0.73	0.77	0.77	0.66	0.77	0.81	0.76	0.96	0.86	0.94	0.98	0.98	0.98	0.98	0.98	0.98
ADA	0.65	0.78	0.74	0.61	0.70	0.56	0.75	0.68	0.69	0.94	0.89	0.91	0.96	0.95	0.94	0.97	0.96	0.96

(*continued on next page*)

Table 7 (*continued*).

Accuracy																		
	ORGD									SMOTE								
	TFIDF			CBOW			SKG			TFIDF			CBOW			SKG		
	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA
GRAB	0.77	0.78	0.68	0.51	0.73	0.65	0.69	0.77	0.72	0.94	0.89	0.92	0.96	0.98 0.92	0.96	0.96	0.97	
WELS	0.68	0.74	0.63	0.73	0.73	0.74	0.76	0.77	0.79	0.81	0.82	0.80	0.88	0.86	0.84	0.9	0.88	0.87
WELSI	0.68	0.71	0.72	0.81	0.81	0.77	0.80	0.87	0.81	0.86	0.84	0.86	0.95	0.93	0.91	0.95	0.94	0.93
WELR	0.74	0.78	0.71	0.81	0.79	0.76	0.84	0.83	0.83	0.88	0.83	0.85	0.93	0.93	0.90	0.95	0.94	0.92
WELT	0.72	0.77	0.71	0.82	0.82	0.78	0.83	0.86	0.80	0.85	0.84	0.86	0.94	0.92	0.91	0.95	0.94	0.94
LSL	0.87	0.77	0.84	0.84	0.86	0.87	0.82	0.91	0.85	0.93	0.84	0.88	1	1	1	1	1	0.98
LSP	0.94	0.89	0.93	1	1	1	1	1	1	0.98	0.89	0.96	1	1	1	1	1	1
LSR	0.92	0.87	0.91	1	1	1	1	1	1	1	0.95	1	1	1	1	1	1	1
ELL	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.78	0.80	0.71	0.80	0.79	0.78	0.81	0.79	0.79
ELR	0.50	0.50	0.50	0.49	0.49	0.50	0.49	0.49	0.50	0.81	0.79	0.79	0.85	0.84	0.83	0.85 0.84	0.82	
ELP	0.50	0.50	0.50	0.48	0.49	0.49	0.53	0.54	0.49	0.89	0.79	0.81	0.91	0.89	0.85	0.92	0.93	0.90
DL1	0.58	0.44	0.47	0.72	0.71	0.65	0.74	0.71	0.75	0.88	0.84	0.87	0.89	0.88	0.88	0.93	0.93 0.91	
DL2	0.69	0.59	0.61	0.71	0.72	0.63	0.75	0.76	0.76	0.89	0.77	0.87	0.89	0.87	0.90	0.93	0.93 0.92	
DL3	0.70	0.53	0.61	0.70	0.72	0.64	0.76	0.74	0.77	0.89	0.68	0.85	0.88	0.87	0.89	0.93	0.92	0.93
DL4	0.67	0.52	0.61	0.72	0.71	0.66	0.73	0.75	0.76	0.88	0.56	0.76	0.87	0.85	0.87	0.93	0.91	0.92
DL5	0.74	0.66	0.70	0.71	0.70	0.70	0.79	0.80	0.79	0.90	0.58	0.72	0.93	0.89	0.91	0.90	0.92	0.92
DL6	0.72	0.66	0.68	0.73	0.73	0.68	0.77	0.78	0.80	0.70	0.71	0.70	0.90	0.90	0.89	0.90	0.93	0.92

confirmed that the performance of the models trained after removing ineffective features using feature selection techniques has a better capability of anti-pattern prediction than all features.

6. Comparative analysis

This section presents the significant impact of various types embedding techniques, data sampling techniques, feature selection techniques, the most popular classifiers, advanced level of classifiers, ensemble learning, and deep learning on the performance of web-service anti-pattern prediction models. These comparisons are made using a box-and-whisker diagram of AUC, F-score, G-mean, and Accuracy parameters, a descriptive statistic in terms of Q1, Q2, Q3, min, max, mean, and hypothesis testing using the Friedman Test with the Wilcoxon Signed Rank Test. The final findings of these techniques are presented in subsequent subsections.

Are statistically significant differences between the predictive models trained using extracted sets of features computed with the help of four-word embedding techniques?

6.1. Different embedding techniques

This subsection examines and discusses the model's prediction capability obtained by different types of embedding techniques. The impact of embedding techniques on the performance of models for predicting five anti-patterns is carried out using four different parameters: Accuracy, F-score, G-mean, and AUC. Finally, the experimental investigation is derived using a box-and-whisker diagram of AUC, F-score, G-mean, and Accuracy parameters, a descriptive statistic, and hypothesis testing using the Friedman Test with the Wilcoxon Signed Rank Test.

Descriptive statistic and box-and-whisker Diagram: Fig. 2 provide the box-and-whisker diagram of accuracy, F-score, G-mean, and AUC values, and Table 9 provides descriptive statistics in Q1, Q2, Q3, min, max, and mean for different word embedding techniques. The anti-pattern models were developed by taking extracted features from the WSDL file using TFIDF and three different embedding techniques. These models are trained using thirty-three different classifiers on original data as well as balanced data. Fig. 2 and Table 9 depict that the model developed using the text metrics generated using the word embedding technique GLOVE shows the best performance with a mean AUC value of 0.83, a median AUC value of 0.90. Similarly, the model developed using other embeddings also generated similar AUC values. Fig. 2 also depicts that some models get a perfect Accuracy of 100% and an AUC value of 1.

Hypothesis Testing: Friedman test with Wilcoxon signed rank test: As mentioned in Fig. 1, two hypothesis tests were conducted on performance analysis: the Friedman Test and the Wilcoxon Signed Rank Test with Bonferroni correction. First, the Friedman test has been used to examine the null hypothesis "*The performance values of Anti-Pattern Models trained using thirty-three different classifiers show no significant improvement after applying different embedding techniques*". We rejected H_0 only if p≤0.05. The 2nd and 7th columns of Table 10 present the Friedman test results on AUC and accuracy values, even the F-measure and G-mean metrics yielded similar results. Friedman test results show that the models trained using different embedding techniques are significantly different because of the p-value ≤0.05. Table 10 also shows the Friedman test results for the different applied word embedding techniques. The Friedman test helps us identify the best text metrics generated to develop the anti-pattern detection model. As indicated in Table 10, the GLOVE secures the lowest mean-rank of 2.14. This, concludes that the text metrics computed by applying the technique GLOVE are best for developing the model for web service anti-pattern detection. On the other hand, CBOW has the highest mean rank, 2.74, indicating that the model developed using metrics generated by CBOW has the worst performance. After the Friedman test results, the study also compared GLOVE with other embedding techniques Wilcoxon signed rank test with Bonferroni correction. The considered hypothesis for the Wilcoxon test is "*The performance values of Anti-Pattern Models trained using GOLVE and other techniques are significantly same*". Table 10 shows the results of the Wilcoxon test with Bonferroni correction, where - is used to present accepted hypothesis, and ∔ is used to represent rejected hypothesis. From Table 10, it is observe that the

Table 8
G-Mean and F-Score for Anti-pattern 1: Different types of text features.

	ORGD									SMOTE								
	TFIDF			CBOW			SKG			TFIDF			CBOW			SKG		
	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA
G-Mean																		
MNB	0.00	0.00	0.00	0.73	0.73	0.75	0.72	0.73	0.77	0.71	0.71	0.7	0.75	0.75	0.75	0.74	0.75	0.74
BNB	0.57	0.53	0.34	0.24	0.24	0.24	0.24	0.24	0.24	0.85	0.82	0.86	0.2	0.13	0.18	0.15	0.18	0.11
GNB	0.53	0.44	0.60	0.73	0.72	0.73	0.73	0.72	0.69	0.77	0.7	0.77	0.81	0.8	0.81	0.8	0.79	0.80
DT	0.53	0.53	0.41	0.40	0.57	0.40	0.53	0.33	0.51	0.89	0.81	0.83	0.91	0.94	0.89	0.94	0.93	0.93
LOGR	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.81	0.78	0.78	0.76	0.79	0.77	0.82	0.81	0.8
KNN	0.00	0.00	0.00	0.00	0.24	0.00	0.34	0.24	0.34	0.89	0.81	0.87	0.87	0.87	0.86	0.89	0.88	0.88
SVCL	0.75	0.69	0.76	0.71	0.71	0.70	0.74	0.77	0.73	0.75	0.53	0.67	0.7	0.7	0.72	0.72	0.62	0.72
SVCP	0.65	0.56	0.61	0.69	0.71	0.72	0.68	0.77	0.71	0.76	0.4	0.67	0.74	0.73	0.75	0.75	0.63	0.76
SVCR	0.70	0.63	0.70	0.74	0.74	0.73	0.79	0.78	0.77	0.77	0.56	0.68	0.71	0.7	0.73	0.71	0.6	0.72
MLPL	0.54	0.34	0.47	0.42	0.24	0.23	0.00	0.00	0.00	0.90	0.79	0.8	0.45	0.86	0.6	0.46	0.94	0.45
MLPS	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.83	0.00	0.8	0.82	0.82	0.79	0.00	0.83	0.00
MLPA	0.34	0.00	0.24	0.00	0.00	0.24	0.00	0.00	0.00	0.92	0.8	0.85	0.00	0.00	0.88	0.00	0.92	0.00
BAG	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.24	0.00	0.86	0.81	0.83	0.82	0.83	0.84	0.85	0.86	0.85
RF	0.24	0.34	0.34	0.24	0.34	0.24	0.42	0.00	0.34	0.9	0.83	0.9	0.92	0.93	0.92	0.94	0.94	0.94
EXTR	0.42	0.53	0.42	0.00	0.41	0.34	0.34	0.24	0.42	0.91	0.81	0.9	0.94	0.93	0.91	0.96 0.95	0.95	
ADA	0.59	0.54	0.48	0.33	0.48	0.23	0.48	0.47	0.53	0.88	0.84	0.84	0.9	0.93	0.87	0.94	0.93	0.94
GRAB	0.63	0.54	0.54	0.40	0.47	0.00	0.40	0.24	0.46	0.86	0.85	0.87	0.93	0.94	0.86	0.93	0.92	0.93
WELS	0.43	0.50	0.40	0.47	0.46	0.44	0.50	0.47	0.45	0.78	0.78	0.77	0.85	0.83	0.84	0.85	0.83	0.83
WELSI	0.82	0.93	0.89	0.90	0.90	0.90	0.85	0.93	0.91	0.89	0.95	0.92	0.9	0.91	0.89	0.89	0.9	0.92
WELR	0.43	0.50	0.40	0.47	0.46	0.44	0.50	0.47	0.45	0.78	0.78	0.77	0.85	0.83	0.84	0.85	0.83	0.83
WELT	0.76	0.89	0.90	0.81	0.93	0.93	0.84	0.80	0.87	0.93	0.96	0.92	0.91	0.92	0.95	0.9	0.89	0.98
LSL	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.84	0.77	0.78	1	1	1	0.98	1	0.93
LSP	0.42	0.59	0.54	1	1	0.77	1	1	0.91	0.95	0.83	0.88	1	1	1	1	1	0.99
LSR	0.34	0	0.34	0.87	1	1	1	0.87	1	0.99	0.88	0.99	1	1	1	1	1	1
ELL	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.77	0.79	0.69	0.78	0.78	0.76	0.79	0.77	0.77
ELR	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.8	0.78	0.79	0.84	0.83	0.82	0.84	0.83	0.81
ELP	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.34	0.00	0.89	0.78	0.81	0.9	0.88	0.84	0.92	0.92	0.89
DL1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.81	0.79	0.78	0.8	0.82	0.81	0.85	0.84	0.82
DL2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.84	0.66	0.79	0.81	0.82	0.82	0.88	0.88	0.85
DL3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.85	0.60	0.80	0.82	0.8	0.83	0.88	0.85	0.86
DL4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.85	0.50	0.67	0.81	0.81	0.81	0.87	0.87 0.86	
DL5	0.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.78	0.54	0.62	0.87	0.85	0.86	0.84	0.88	0.87
DL6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.34	0.57	0.61	0.61	0.87	0.83	0.84	0.87	0.9	0.86
F-Score																		
MNB	0.00	0.00	0.00	0.29	0.29	0.32	0.28	0.29	0.34	0.73	0.77	0.75	0.81	0.81	0.81	0.8	0.81	0.8
BNB	0.31	0.37	0.17	0.07	0.08	0.07	0.07	0.07	0.08	0.86	0.82	0.86	0.67	0.67	0.67	0.67	0.67	0.67
GNB	0.17	0.16	0.30	0.29	0.29	0.29	0.29	0.28	0.26	0.79	0.66	0.79	0.84	0.83	0.84	0.83	0.82	0.83
DT	0.33	0.31	0.19	0.18	0.30	0.18	0.30	0.13	0.24	0.89	0.81	0.83	0.92	0.94	0.89	0.94	0.93	0.93
LOGR	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.83	0.76	0.8	0.8	0.82	0.81	0.85	0.84	0.84
KNN	0.00	0.00	0.00	0.00	0.09	0.00	0.20	0.10	0.18	0.9	0.8	0.87	0.89	0.88	0.88	0.9	0.9	0.9
SVCL	0.37	0.33	0.39	0.29	0.28	0.27	0.35	0.35	0.30	0.73	0.44	0.63	0.67	0.67	0.69	0.7	0.56	0.69
SVCP	0.37	0.28	0.35	0.33	0.32	0.34	0.37	0.43	0.38	0.73	0.27	0.62	0.71	0.71	0.73	0.73	0.58	0.74
SVCR	0.36	0.31	0.35	0.32	0.31	0.30	0.36	0.34	0.34	0.74	0.48	0.63	0.68	0.67	0.7	0.68	0.53	0.70
MLPL	0.38	0.15	0.26	0.26	0.10	0.06	0.00	0.00	0.00	0.9	0.78	0.83	0.40	0.88	0.58	0.41	0.95	0.51
MLPS	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.84	0.66	0.81	0.85	0.84	0.82	0.00	0.85	0.67
MLPA	0.18	0.00	0.10	0.00	0.00	0.09	0.00	0.00	0.00	0.92	0.79	0.85	0.00	0.00	0.9	0.00	0.92	0.67
BAG	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.00	0.86	0.81	0.83	0.85	0.86	0.86	0.87	0.88	0.87

(continued on next page)

above designed hypothesis is rejected for all pairs. As a result, it can be concluded that the performance of various word embedding techniques used for generating text metrics differs significantly in the detection of web service anti-patterns.

Table 8 (*continued*).

G-Mean																		
	ORGD									SMOTE								
	TFIDF			CBOW			SKG			TFIDF			CBOW			SKG		
	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA	AF	SIGF	GA
RF	0.09	0.19	0.17	0.09	0.17	0.08	0.26	0.00	0.19	0.9	0.82	0.9	0.92	0.93	0.92	0.94	0.95	0.95
EXTR	0.25	0.37	0.25	0.00	0.23	0.17	0.18	0.10	0.25	0.91	0.8	0.9	0.94	0.93	0.92	0.96	0.95	0.95
ADA	0.43	0.40	0.32	0.14	0.29	0.06	0.30	0.27	0.37	0.89	0.84	0.85	0.90	0.93	0.88	0.94	0.93	0.94
GRAB	0.45	0.40	0.40	0.18	0.24	0.00	0.18	0.07	0.22	0.86	0.85	0.87	0.93	0.94	0.86	0.93	0.93	0.93
WELS	0.30	0.38	0.26	0.34	0.34	0.32	0.37	0.36	0.33	0.79	0.75	0.78	0.86	0.83	0.84	0.85	0.84	0.84
WELSI	0.72	0.88	0.82	0.80	0.80	0.86	0.67	0.88	0.87	0.91	0.95	0.93	0.93	0.94	0.94	0.92	0.94	0.95
WELR	0.30	0.38	0.26	0.34	0.34	0.32	0.37	0.36	0.33	0.79	0.75	0.78	0.86	0.83	0.84	0.85	0.84	0.84
WELT	0.67	0.82	0.85	0.76	0.89	0.92	0.78	0.78	0.85	0.94	0.95	0.94	0.93	0.94	0.97	0.91	0.92	0.98
LSL	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.85	0.75	0.8	1	1	1	0.98	1	0.94
LSP	0.30	0.50	0.45	1	1	0.74	1	1 0.90	0.95	0.82	0.89	1	1	1	1	1	0.99	
LSR	0.21	0.00	0.21	0.87	1	1	1	0.87	1	0.99	0.88	0.99	1	1	1	1	1	1
ELL	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.81	0.81	0.76	0.82	0.82	0.81	0.83	0.82	0.82
ELR	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.82	0.77	0.81	0.87	0.86	0.85	0.87	0.86	0.84
ELP	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.16	0.00	0.89	0.77	0.82	0.92	0.9	0.86	0.93	0.93	0.91
DL1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.82	0.77	0.8	0.83	0.84	0.84	0.88	0.86	0.84
DL2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.84	0.73	0.8	0.84	0.84	0.84	0.9	0.89	0.87
DL3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.86	0.63	0.81	0.85	0.83	0.85	0.89	0.87	0.88
DL4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.85	0.6	0.67	0.83	0.84	0.83	0.89	0.89	0.88
DL5	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.81	0.53	0.66	0.88	0.86	0.87	0.85	0.89	0.89
DL6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.53	0.57	0.57	0.88	0.86	0.85	0.89	0.91	0.88

Table 9
Descriptive statistics: Different embedding techniques.

	Median	Mean	Max	Min	Q3	Q1
Accuracy						
TFIDF	90.28	84.97	100.00	27.78	95.79	76.77
CBOW	85.11	82.35	100.00	40.05	91.80	75.27
SKG	84.45	82.01	100.00	32.45	91.94	74.73
GLOVE	88.19	84.20	100.00	37.54	93.82	78.46
AUC						
TFIDF	0.91	0.82	1.00	0.17	0.97	0.67
CBOW	0.86	0.81	1.00	0.13	0.94	0.73
SKG	0.85	0.81	1.00	0.12	0.95	0.71
GLOVE	0.90	0.83	1.00	0.14	0.96	0.75

Table 10
Hypothesis testing: AUC and Accuracy: Different embedding techniques.

	Accuracy					AUC				
	Rank : p-v≤0.05	TFIDF	CBOW	SKG	GLOVE	Rank : p-v≤0.05	TFIDF	CBOW	SKG	GLOVE
TFIDF	2.27	-	‡	‡	‡	2.48	-	‡	‡	‡
CBOW	2.71	‡	-	‡	‡	2.74	‡	-	‡	‡
SKG	2.75	‡	‡	-	‡	2.63	‡	‡	-	‡
GLOVE	2.27	‡	‡	‡	-	2.14	‡	‡	‡	-

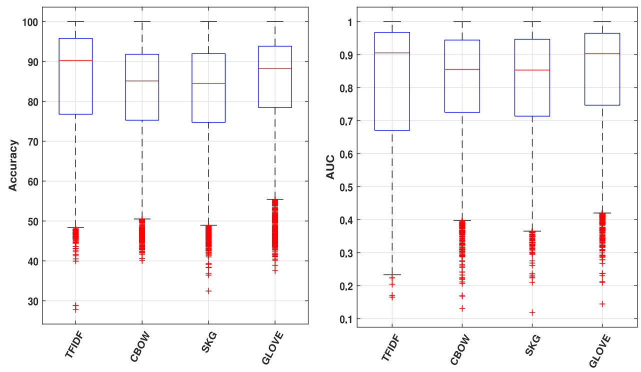
Is there a significant performance difference between models produced using all features and those created using the subset of features chosen using the eight feature selection techniques?

6.2. Different features selection techniques

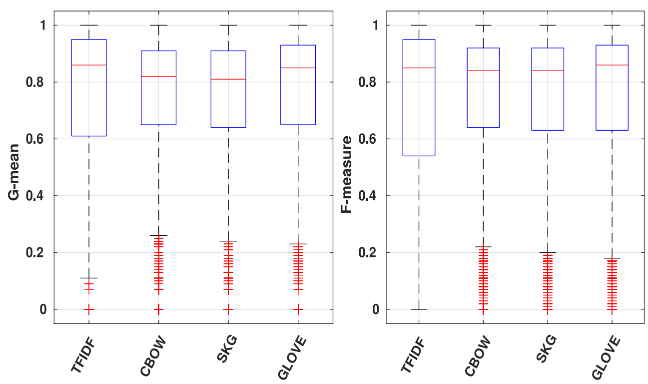
This subsection presents, the analysis and discusses the model's prediction capability obtained after removing ineffective features using eight feature selection techniques. All these features sets are deduced after applying feature selection techniques, and the original features (AF) are used as input for developing the models for detecting web service anti-patterns. The impact of these techniques on models is compared using accuracy, AUC, F-measure, and G-mean value. The experimental investigation is derived using a box-and-whisker diagram of AUC and Accuracy parameters, a descriptive statistic, and hypothesis testing using the Friedman Test with the Wilcoxon Signed Rank Test.

Descriptive statistic and box-and-whisker Diagram:

Fig. 3(a) and (b) represent the box-and-whisker diagram of accuracy and AUC values, G-mean and F-measure respectively. And Table 11 epitomizes the descriptive statistics in Q1, Q3, min, max, and mean for



(a) Accuracy and AUC



(b) F-measure and G-mean

Fig. 2. Box-and-whisker diagram: Different embedding techniques.

different feature selection techniques. Fig. 3 and Table 11 depict that the models trained after removing ineffective features using SIGF show the best performance with a mean AUC value of 0.84, a median AUC value of 0.91, and a median G-mean value of 0.86. Additionally, by achieving a median F-score value of 0.87, the SIGF and CORR feature selection techniques demonstrate superior performance. Similarly, the model trained after removing ineffective features using other feature selection techniques has a similar prediction ability than AF features. This improvement in their prediction ability is due to the removal of ineffective features. In the case of original features, some features were ineffective for anti-pattern prediction, so it is difficult to ignore during the training process of ML models. Removal of ineffective features makes it easy to find a pattern that has a better ability for Anti-pattern prediction.

Hypothesis Testing: Friedman test with Wilcoxon signed rank test: Similar to the previous subsections, the Friedman Test and Wilcoxon Signed Rank Test with Bonferroni correction have been applied to analyze performance. The Friedman test was applied first, to evaluate the null hypothesis "*The performance values of Anti-Pattern Models trained using thirty-three different classifiers show no significant improvement after removing ineffective features using feature selection techniques*". We rejected H_0 only if p≤0.05. The 2nd and 12th columns of Table 12 present the Friedman test results on AUC and accuracy values, similar results were revealed even for F-measure and G-mean metrics. The results confirm that the models trained after removing ineffective features using feature selection techniques significantly differ in their ability to predict the Anti-pattern present in web services. The results of Table 12 also confirm that the SIGF computed features has the lowest mean rank of 3.58. The analysis thus concludes that, the models trained after removing ineffective features are best for web service anti-pattern detection. The study also compares the SIGF with other feature selection techniques following the results of the Friedman test using the Wilcoxon signed rank test with Bonferroni correction. The considered hypothesis for the Wilcoxon test is "*The performance values of Anti-Pattern Models trained using SIGF and other techniques are significantly same*". Table 12 shows the results of the Wilcoxon test with Bonferroni correction, where - is used to present the accepted hypothesis, and ‡ is used to represent rejected hypothesis. From Table 12, it is observed that the above-designed hypothesis is rejected for all pairs. As a result, the investigation's findings indicate that removing ineffective features significantly improves the performance of anti-pattern prediction models.

Is there a statistically significant difference between the performances demonstrated by the five data sampling techniques and those exhibited by the initial data?

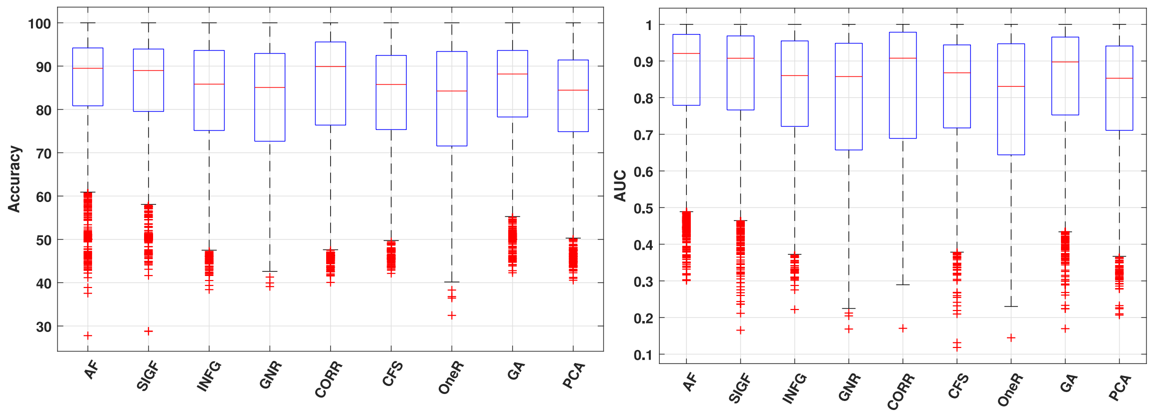
6.3. Different data sampling approaches

This subsection analyzes and discusses the model's prediction capability obtained after training on balanced data using different sampling approaches. In the proposed work, five different variants of data sampling approaches have been used to overcome the class imbalance problem of datasets. The final benefit of using these techniques is compared using accuracy, AUC, F-measure, and G-mean values, and their investigation is carried out using a box-and-whisker diagram of these values with hypothesis testing using the Friedman Test with the Wilcoxon Signed Rank Test.

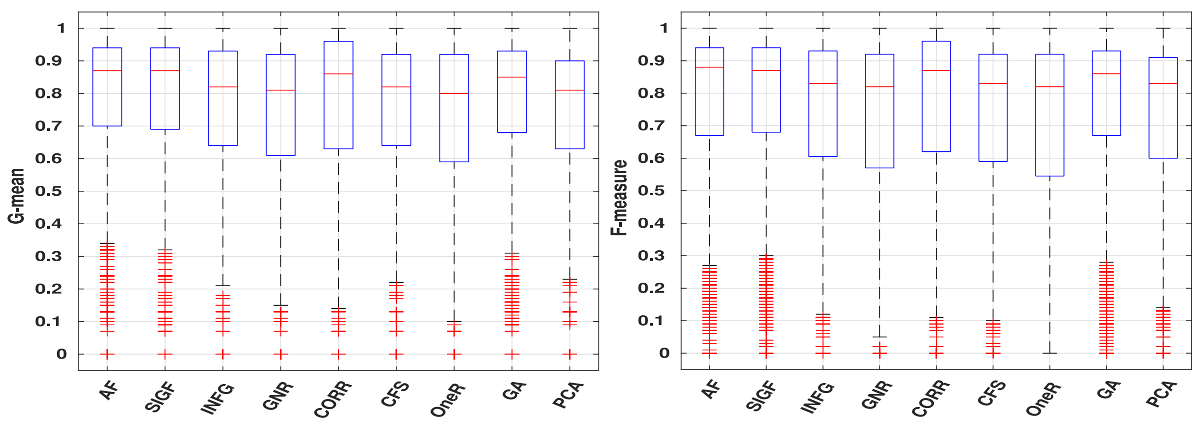
Descriptive statistic and box-and-whisker Diagram: Fig. 4(a) and (b) provide the box-and-whisker diagram of accuracy and AUC values, G-mean and F-measure respectively, and Table 14 provides descriptive statistics in Q1, Q2, Q3, min, max, and mean for different data sampling approaches. The results of Fig. 4 and Table 14 confirmed that the models trained on sampled data have the best performance, with a mean AUC value of 0.86, a median AUC value of 0.91, a median F-score value of 0.92 and a median G-mean value of 0.83. The results also confirmed that the models trained on original data have 0.70 as the mean AUC and 0.73 as Median AUC. Thus, following the use sampling approach the AUC values of models improved from 0.70 to 0.86, i.e., 22.8% improvement in their prediction ability to predict anti-patterns. Also, the F-measure and G-mean values improved post-employment of sampling approaches. Therefore, balancing data using a sampling approach is more helpful for analyzing the anti-patterns in web services.

Hypothesis Testing: Friedman test with Wilcoxon signed rank test: The same Friedman Test and Wilcoxon Signed Rank Test have been used here with Bonferroni correction, to find the significant impact of using sampling approaches. The null hypothesis has been initially investigated using the Friedman test "*performance values of Anti-Pattern Models trained using thirty-three different classifiers show no significant improvement after training on balanced data*". We rejected H_0 only if p≤0.05. The 2nd and 9th columns of Table 13 present the Friedman test results on AUC and Accuracy values, even the F-measure and G-mean metrics produced similar outcomes. The above hypothesis was found significant because of the p-value ≤ 0.05. The Friedman Test depicts that the models trained on sampled data are not significantly similar to the models trained on original data.

As mentioned in Table 13, the Friedman Test confirms that the models trained on balanced data using ADSYN have the lowest mean rank of 2.93. Therefore, it can be concluded that the models trained on sampled data using ADSYN are best for web service anti-pattern detection. Unlike other sampling approaches, ADSYN does not follow the concept of an equal number of samples for all minority classes. The ADSYN concept focuses more on samples that lie in the safe region and removes samples with noise. Following this test, a pairwise comparison of other sampling techniques with ADSYN using the Wilcoxon signed rank test has been conducted. The considered hypothesis for the Wilcoxon test is "*The performance values of Anti-Pattern Models*



(a) Accuracy and AUC



(b) G-mean and F-measure

Fig. 3. Box-and-whisker diagram: Different features aelection techniques.

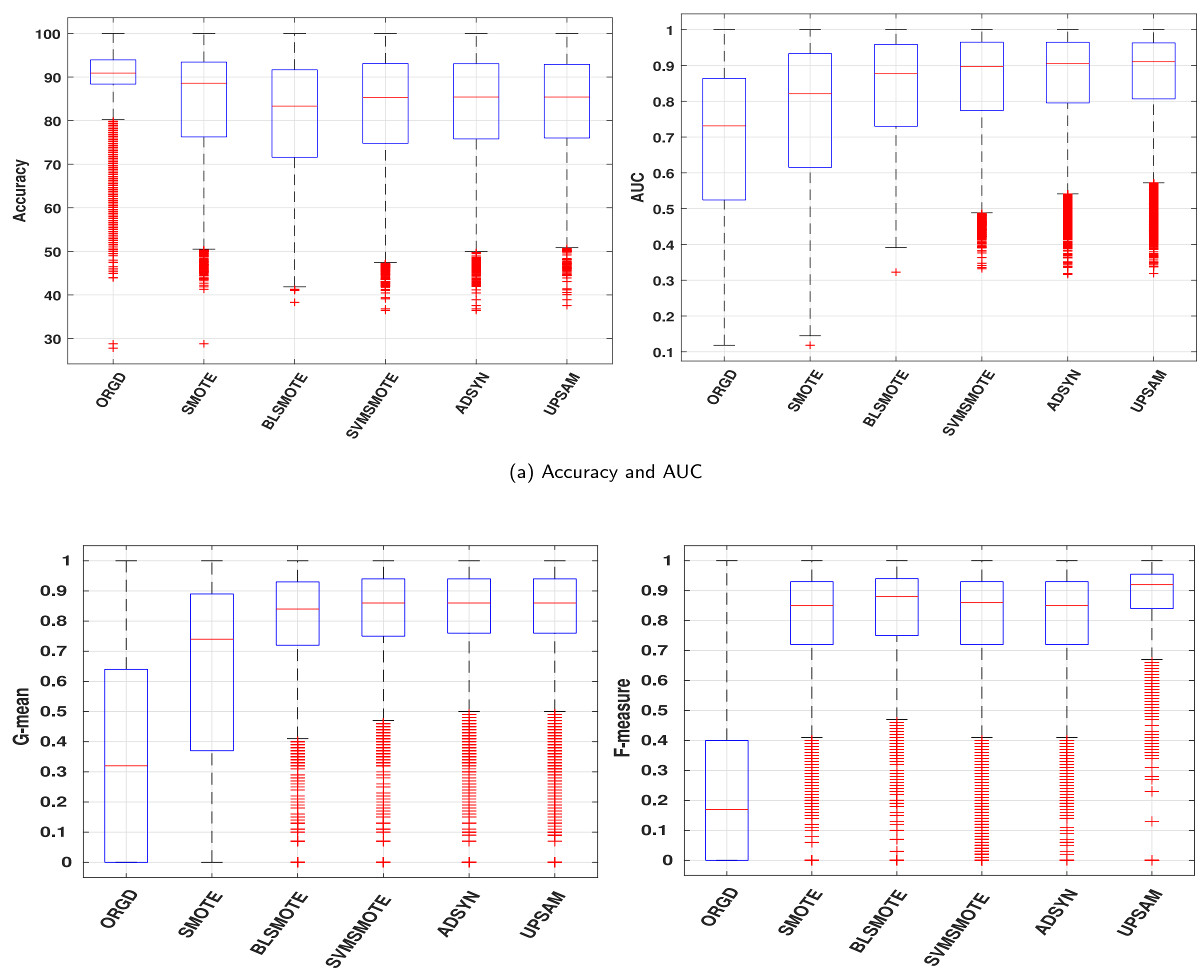
Table 11
Descriptive statistics: Different features selection techniques.

	Accuracy					
	Median	Mean	Max	Min	Q3	Q1
AF	89.50	84.64	100.00	27.78	94.20	80.83
SIGF	88.98	85.42	100.00	28.79	93.94	79.54
INFG	85.83	82.82	100.00	38.40	93.62	75.14
GNR	85.08	81.99	100.00	39.10	92.93	72.65
CORR	89.90	84.10	100.00	40.05	95.58	76.38
CFS	85.76	82.67	100.00	42.13	92.47	75.35
OneR	84.25	81.51	100.00	32.45	93.38	71.56
GA	88.18	84.33	100.00	42.29	93.62	78.25
PCA	84.44	81.92	100.00	40.56	91.41	74.86
	AUC					
AF	0.91	0.83	1.00	0.30	0.97	0.78
SIGF	0.91	0.84	1.00	0.17	0.97	0.77
INFG	0.86	0.82	1.00	0.22	0.95	0.72
GNR	0.86	0.80	1.00	0.17	0.95	0.66
CORR	0.91	0.82	1.00	0.17	0.98	0.69
CFS	0.87	0.81	1.00	0.12	0.94	0.72
OneR	0.83	0.79	1.00	0.14	0.95	0.64
GA	0.90	0.83	1.00	0.17	0.97	0.75
PCA	0.85	0.81	1.00	0.21	0.94	0.71

do not differ significantly when other sampling approaches are used for balancing the dataset". Table 13 shows the results of the Wilcoxon test with Bonferroni correction on accuracy, F-measure, AUC, and G-mean values, where - is used to present the accepted hypothesis, and ‡ is used to represent rejected hypothesis. From Table 13, it is observed that the ADSYN outperforms significantly as compared to other sampling approaches, while in the case of UPSAM, the models are significantly similar. As a result, the study advises the application of a sampling approach for anti-pattern prediction. In the case of original data, a very small number of WSDL files have anti-patterns, so it becomes very

Table 12
Hypothesis testing: Different feature selection techniques.

	Accuracy										AUC									
	Rank : p-v≤0.05	AF	SIGF	INFG	GNR	CORR	CFS	OneR	GA	PCA	Rank : p-v≤0.05	AF	SIGF	INFG	GNR	CORR	CFS	OneR	GA	PCA
AF	4.14	-	‡	‡	‡	‡	‡	‡	‡	‡	3.98	-	‡	‡	‡	‡	‡	‡	‡	‡
SIGF	3.86	‡	-	‡	‡	‡	‡	‡	‡	‡	3.58	‡	-	‡	‡	‡	‡	‡	‡	‡
INFG	5.37	‡	‡	-	-	‡	-	‡	‡	‡	5.35	‡	‡	-	‡	‡	-	‡	‡	‡
GNR	5.42	‡	‡	-	-	‡	‡	‡	‡	‡	5.59	‡	‡	‡	-	‡	‡	‡	‡	‡
CORR	4.56	‡	‡	‡	‡	-	‡	‡	‡	‡	4.53	‡	‡	‡	‡	-	‡	‡	‡	‡
CFS	5.45	‡	‡	-	‡	‡	-	‡	‡	‡	5.54	‡	‡	-	‡	‡	-	‡	‡	‡
OneR	5.91	‡	‡	‡	‡	‡	‡	-	‡	-	6.18	‡	‡	‡	‡	‡	‡	-	‡	‡
GA	4.37	‡	‡	‡	‡	‡	‡	‡	-	‡	4.27	‡	‡	‡	‡	‡	‡	‡	-	‡
PCA	5.92	‡	‡	‡	‡	‡	‡	-	‡	-	5.97	‡	‡	‡	‡	‡	‡	‡	‡	-



(a) Accuracy and AUC

(b) G-mean and F-measure

Fig. 4. Box-and-whisker diagram: Different techniques to deal with classes imbalanced problem.

difficult for machine learning techniques to find important patterns. The primary reason is that machine learning techniques concepts are based on the assumption of training data.

How effective are the 33 learning algorithms in terms of AUC and Accuracy parameters? Does the expected performance of the 33 classifier techniques differ significantly?

6.4. Classification techniques

This subsection analyzes and discusses the performance of models trained using different variants of machine learning algorithms. To develop models for detecting web service anti-patterns, a variety of classifier techniques have been employed in this study, including general classifier techniques such as Gaussian Naive Bayes (GNB),

Table 13
Hypothesis testing: AUC: Different techniques to deal with Classes Imbalanced Problem.

	Accuracy							AUC						
	Rank : p-$v \leq 0.05$	ORGD	SMOTE	BLSMOTE	SVMSMOTE	ADSYN	UPSAM	Rank : p-$v \leq 0.05$	ORGD	SMOTE	BLSMOTE	SVMSMOTE	ADSYN	UPSAM
ORGD	2.90	-	ǂ	ǂ	ǂ	ǂ	ǂ	4.89	-	ǂ	ǂ	ǂ	ǂ	ǂ
SMOTE	3.55	ǂ	-	ǂ	ǂ	-	-	3.93	ǂ	-	ǂ	ǂ	ǂ	ǂ
BLSMOTE	3.92	ǂ	ǂ	-	ǂ	ǂ	ǂ	3.25	ǂ	ǂ	-	ǂ	ǂ	ǂ
SVMSMOTE	3.56	ǂ	ǂ	ǂ	-	-	ǂ	3.02	ǂ	ǂ	ǂ	-	ǂ	ǂ
ADSYN	3.51	ǂ	-	ǂ	-	-	-	2.93	ǂ	ǂ	ǂ	ǂ	-	-
UPSAM	3.56	ǂ	-	ǂ	ǂ	-	-	2.97	ǂ	ǂ	ǂ	ǂ	-	-

Table 14
Descriptive statistics: Different techniques to deal with Classes Imbalanced Problem.

	Median	Mean	Max	Min	Q3	Q1
	Accuracy					
ORGD	90.91	88.03	100.00	27.78	93.94	88.38
SMOTE	88.59	83.45	100.00	28.79	93.43	76.26
SVMSMOTE	85.28	82.39	100.00	36.44	93.09	74.80
BLSMOTE	83.33	80.71	100.00	38.30	91.67	71.59
UPSAM	85.41	83.20	100.00	37.54	92.91	76.01
ADSYN	85.42	82.85	100.00	36.44	93.06	75.80
	AUC					
ORGD	0.73	0.70	1.00	0.12	0.86	0.52
SMOTE	0.82	0.77	1.00	0.12	0.93	0.62
SVMSMOTE	0.90	0.84	1.00	0.33	0.97	0.77
BLSMOTE	0.88	0.83	1.00	0.32	0.96	0.73
UPSAM	0.91	0.86	1.00	0.32	0.96	0.81
ADSYN	0.90	0.85	1.00	0.32	0.96	0.80

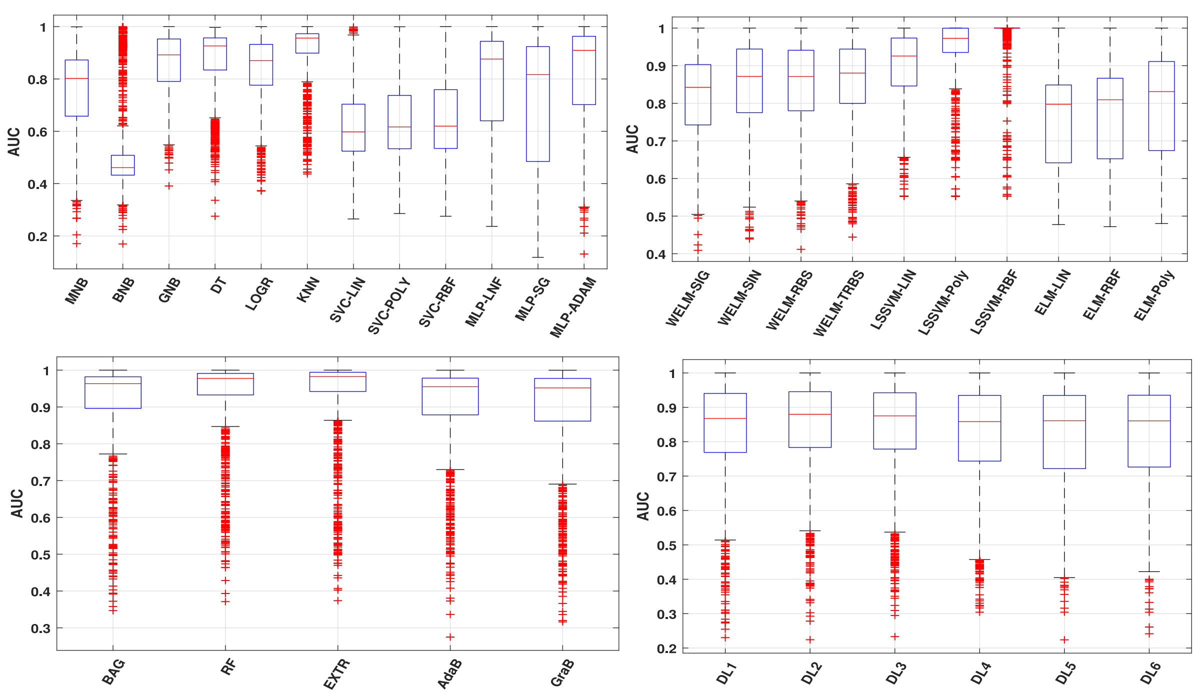


Fig. 5. AUC: box-and-whisker diagram: Different machine learning techniques.

Multinomial Naive Bayes (MNB), Bernoulli Naive Bayes (BNB), Logistic Regression Analysis (LOGR), Support Vector Classifier with linear Kernel (SVC-LIN), SVC with a polynomial kernel (SVC-POLY), and SVC with radial bias kernel (SVC-RBF). Further, the deep layer technique with a varying number of hidden layers, i.e., DL with one hidden layer (DL1), DL with two hidden layers (DL2), DL with three hidden layers (DL3), and DL with four hidden layers (DL4) have been used. Apart from this, advanced learning classifiers such as extreme learning

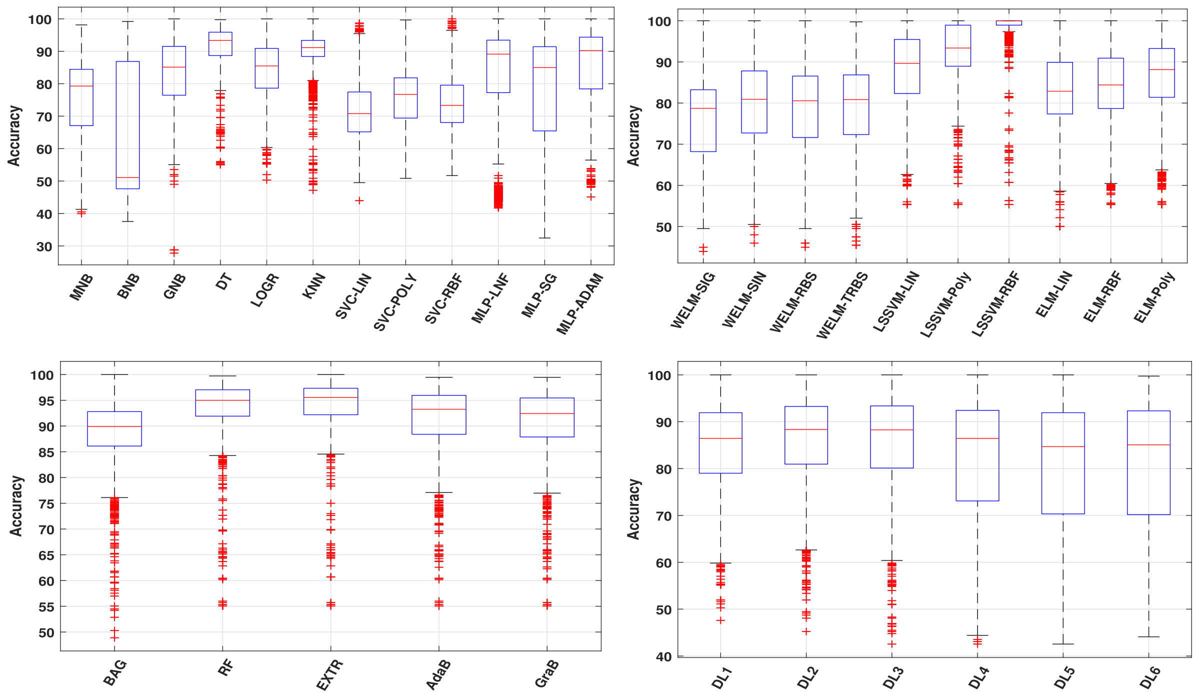


Fig. 6. Accuracy: box-and-whisker diagram: Different machine learning techniques.

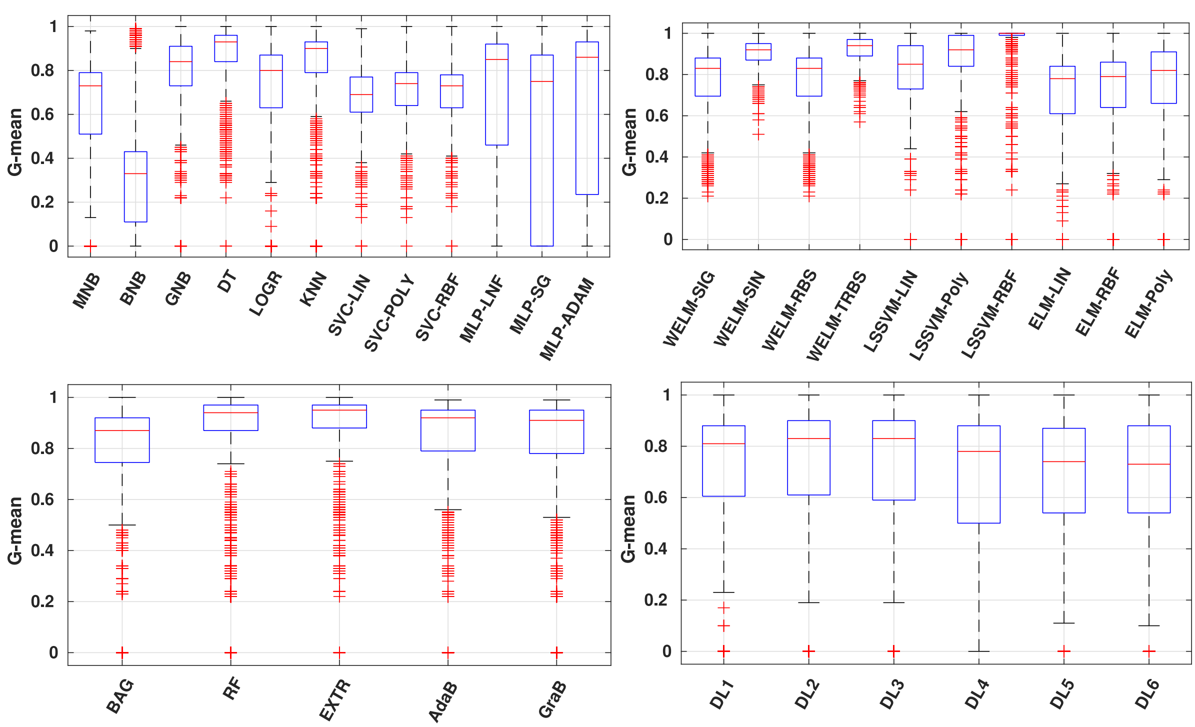


Fig. 7. G-Mean: box-and-whisker diagram: Different machine learning techniques.

machine (ELM) and weighted extreme learning machine (WELM) are employed. The ELM algorithm is implemented with three kernels, namely: Linear (ELM-LIN), Radial Bias (ELM-RBF), and Poly (ELM-POLY), whereas the WELM algorithm is implemented with four kernels, namely: Sigmoid (WELM-SIG), Radbas (WELM-RBS), Tribas (WELM-TRBS), and Sine (WELM-SIN). Furthermore, ensemble classifiers such as Bagging Classifier (BAG), Random Forest Classifier (RF), Extra Trees Classifier (EXTR), AdaBoost Classifier (AdaB), Gradient Boosting Classifier (GraB) are used for training the models for the detection of web service anti-patterns using text metrics as input. The anti-pattern prediction models built are validated using the 5-fold cross-validation approach. The final benefit of using these techniques is compared using accuracy, AUC, F-measure, and G-mean values, and their investigation is carried out using a box-and-whisker diagram of these values with

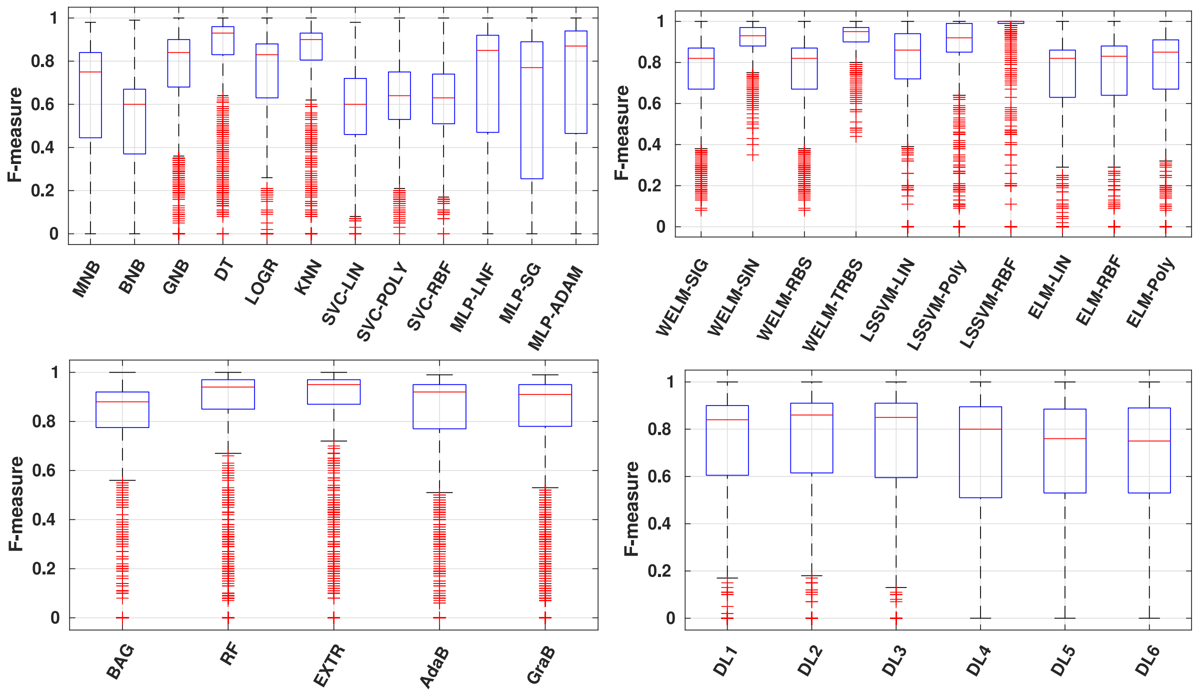


Fig. 8. F-Measure: box-and-whisker diagram: Different machine learning techniques.

Table 15
Descriptive statistics: Different machine learning techniques.

	Accuracy						AUC					
	Median	Mean	Max	Min	Q3	Q1	Median	Mean	Max	Min	Q3	Q1
MNB	79.28	76.46	98.14	39.94	84.44	67.08	0.80	0.76	1.00	0.17	0.87	0.66
BNB	51.08	63.01	99.20	37.54	86.87	47.62	0.46	0.53	1.00	0.17	0.51	0.43
GNB	85.11	82.89	100.00	27.78	91.52	76.46	0.89	0.86	1.00	0.39	0.95	0.79
DT	93.38	91.46	99.73	55.06	95.89	88.67	0.93	0.87	1.00	0.28	0.96	0.83
LOGR	85.48	83.73	100.00	50.29	90.91	78.63	0.87	0.84	1.00	0.37	0.93	0.78
KNN	91.12	89.46	100.00	47.14	93.37	88.40	0.96	0.91	1.00	0.44	0.97	0.90
SVCL	70.79	71.71	98.65	43.94	77.46	65.17	0.60	0.63	1.00	0.26	0.70	0.52
SVCP	76.72	76.34	99.66	50.84	81.82	69.41	0.62	0.65	1.00	0.29	0.74	0.53
SVCR	73.33	74.17	100.00	51.69	79.55	68.04	0.62	0.65	1.00	0.28	0.76	0.53
MLPL	89.13	81.93	100.00	41.67	93.43	77.25	0.88	0.79	1.00	0.24	0.94	0.64
MLPS	84.97	78.79	100.00	32.45	91.41	65.46	0.82	0.72	1.00	0.12	0.92	0.48
MLPA	90.20	83.99	100.00	45.07	94.35	78.41	0.91	0.81	1.00	0.13	0.96	0.70
BAG	89.90	88.09	100.00	48.88	92.80	86.11	0.96	0.91	1.00	0.35	0.98	0.90
RF	95.00	93.38	99.73	55.06	97.04	91.92	0.98	0.93	1.00	0.37	0.99	0.93
EXTR	95.56	93.79	100.00	55.06	97.34	92.19	0.98	0.93	1.00	0.37	0.99	0.94
ADA	93.26	90.78	99.46	55.06	95.96	88.40	0.96	0.90	1.00	0.28	0.98	0.88
GRAB	92.42	90.30	99.46	55.06	95.45	87.88	0.95	0.89	1.00	0.32	0.98	0.86
WELS	78.73	76.63	100.00	43.94	83.23	68.18	0.84	0.82	1.00	0.41	0.90	0.74
WELSI	80.91	79.65	100.00	45.96	87.81	72.73	0.87	0.85	1.00	0.44	0.94	0.77
WELR	80.56	78.88	100.00	44.95	86.56	71.63	0.87	0.85	1.00	0.41	0.94	0.78
WELT	80.83	79.27	99.73	45.45	86.85	72.34	0.88	0.85	1.00	0.44	0.94	0.80
LSL	89.65	87.67	100.00	55.34	95.45	82.32	0.93	0.90	1.00	0.55	0.97	0.85
LSP	93.38	92.13	100.00	55.34	98.93	88.95	0.97	0.95	1.00	0.55	1.00	0.94
LSR	100.00	98.16	100.00	55.34	100.00	98.93	1.00	0.98	1.00	0.55	1.00	1.00
ELL	82.87	81.93	100.00	50.00	89.90	77.35	0.80	0.75	1.00	0.48	0.85	0.64
ELR	84.41	83.50	100.00	55.34	90.91	78.68	0.81	0.76	1.00	0.47	0.87	0.65
ELP	88.13	85.78	100.00	55.34	93.27	81.40	0.83	0.79	1.00	0.48	0.91	0.67
DL1	86.45	84.17	100.00	47.61	91.92	79.01	0.87	0.83	1.00	0.23	0.94	0.77
DL2	88.33	85.27	100.00	45.22	93.26	80.95	0.88	0.84	1.00	0.22	0.95	0.78
DL3	88.27	84.85	100.00	42.55	93.37	80.11	0.88	0.83	1.00	0.23	0.94	0.78
DL4	86.43	82.11	100.00	42.55	92.42	73.10	0.86	0.81	1.00	0.30	0.93	0.74
DL5	84.68	80.54	100.00	42.55	91.92	70.33	0.86	0.81	1.00	0.22	0.93	0.72
DL6	85.07	80.66	99.73	44.10	92.32	70.18	0.86	0.81	1.00	0.24	0.94	0.73

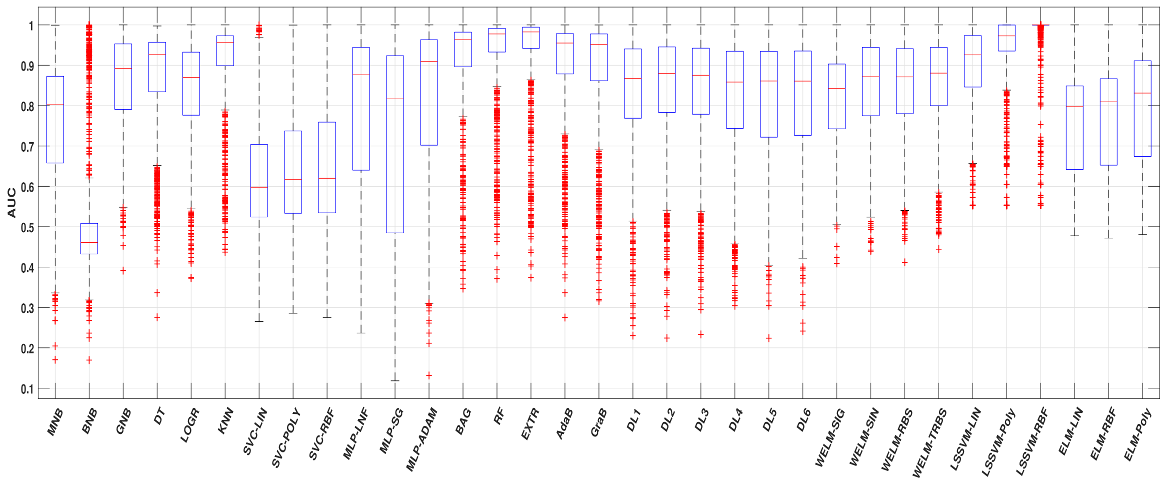


Fig. 9. Accuracy: box-and-whisker diagram: Different machine learning techniques.

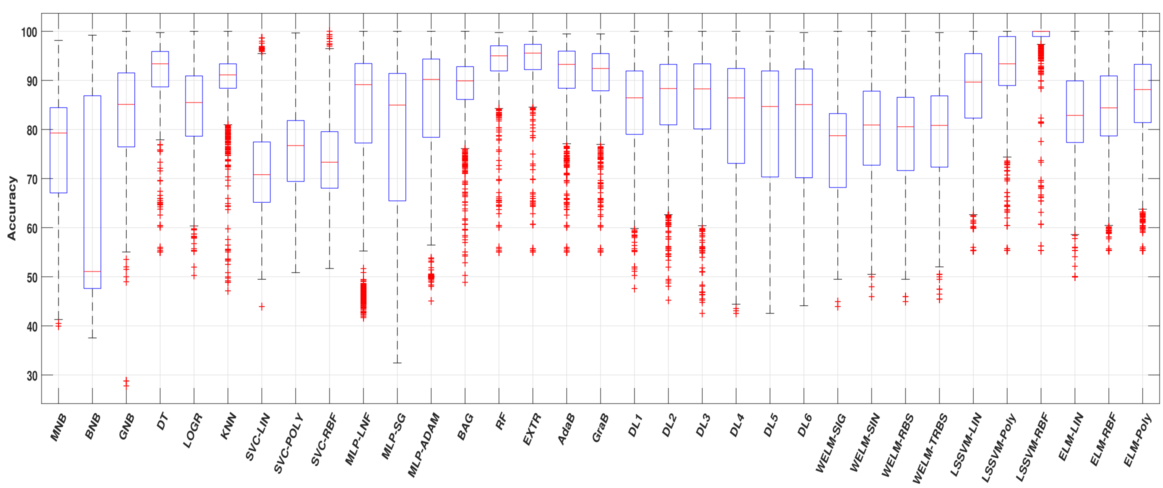


Fig. 10. AUC: box-and-whisker diagram: Different machine learning techniques.

hypothesis testing using the Friedman Test with the Wilcoxon Signed Rank Test.

Descriptive statistic and box-and-whisker Diagram:

Table 15 shows the descriptive statistics for all the classifier techniques. Figs. 5 to 10 show the AUC, accuracy, G-mean and F-measure box-and-whiskers for the different categories of classifier techniques. The examination of Figs. 5 to 10 and Table 15 led to the following conclusion:

- Among the general classifiers category, DT outperformed by earning mean AUC value of 0.87. In contrast, BNB offers the worst performance, with a mean AUC value of 0.53.
- Among the ensemble classifiers employed for training the models for anti-patterns detection, the Extra tree classifier (EXTR) outperformed by earning mean AUC value of 0.93, and GraB shows the worst performance with a mean AUC value of 0.89.
- Of all the deep learning algorithms with varying hidden layers, DL2 outperformed by earning mean AUC value of 0.88. DL4, DL5, and DL6 are offering the worst performance, with a mean AUC value of 0.86.
- Over the advanced ML classifiers used for developing models for detecting web service anti-patterns, LSSVM-RBF outperformed by earning mean AUC value of 0.98. In contrast, the models trained using ELM-LIN show the worst performance with a mean AUC value of 0.75.
- From Figs. 9, 10, we infer that the models trained using LSSVM-RBF outperformed by earning mean AUC value of 0.98. LSSVM-POLY, RF, and EXTR show better performance after LSSVM-RBF with a mean AUC value of 0.95, 0.93, and 0.93, respectively. BNB is showing the worst performance with a mean AUC value of 0.53.
- The performance analysis indicates the LSSVM-RBF achieved better results consistently. Fig. 11, illustrates that LSSVM-RBF attains the highest F-score. As shown in Fig. 12, LSSVM-RBF also achieves the maximum G-mean value.

Hypothesis Testing: Friedman test with Wilcoxon signed rank test: Similar to the previous sub-sections, to assess the significant impact of using different training algorithms for finding an essential pattern for predicting anti-patterns present in web services, the Friedman and Wilcoxon Signed Rank Test with Bonferroni correction has

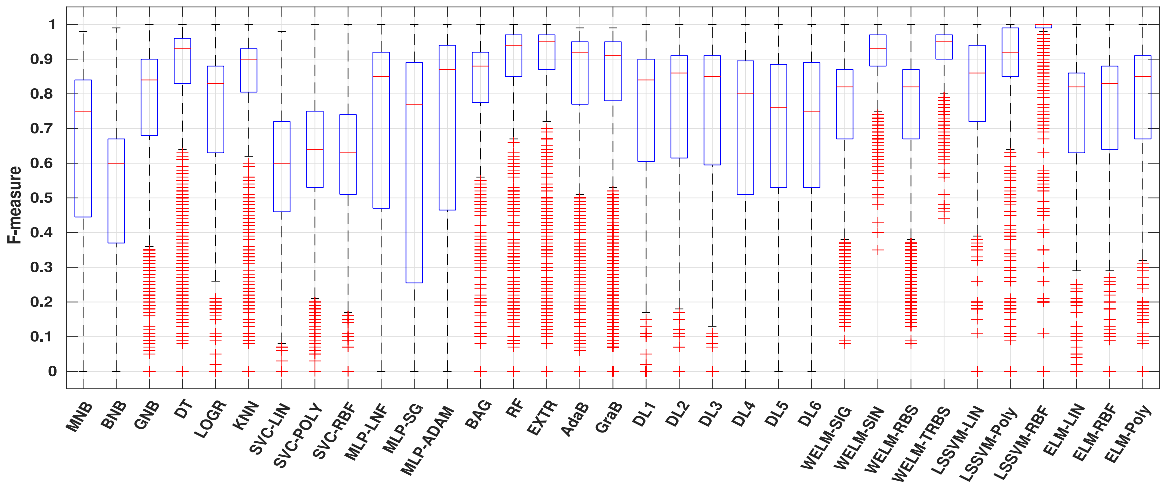


Fig. 11. F-Measure: box-and-whisker diagram: Different machine learning techniques.

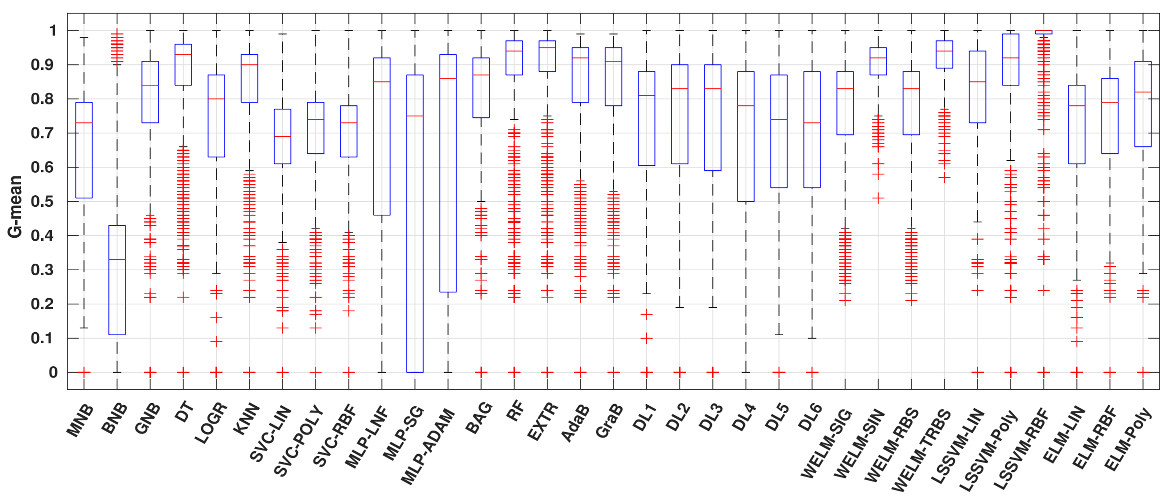


Fig. 12. G-Mean: box-and-whisker diagram: Different machine learning techniques.

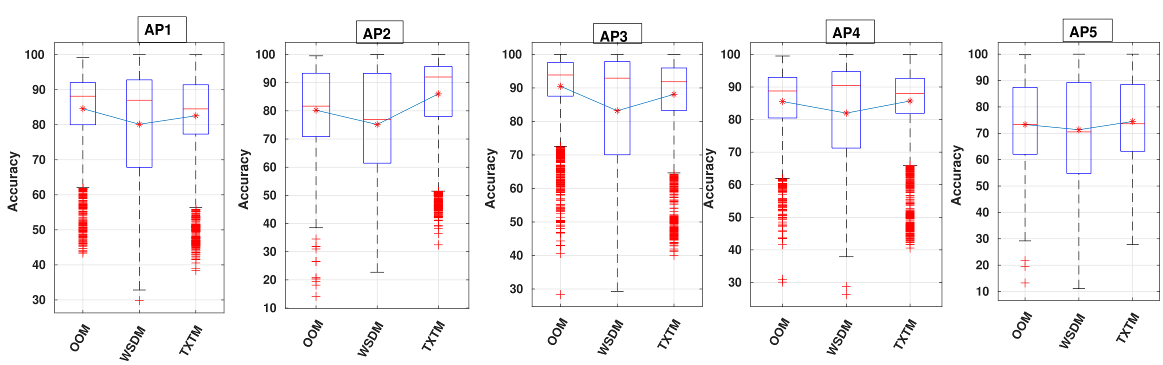


Fig. 13. Accuracy: box-and-whisker diagram: Different metrics sets.

been used. In order to test the null hypothesis, the Friedman test is used first, "*performance values of Anti-Pattern Models show no significant improvement after changing training algorithms*". Table 16 presents the Friedman test results on AUC and Accuracy values. The Friedman Test depicts that the models trained using different algorithms are not significantly similar. From Table 16, with a mean AUC rank of 1.37, the analysis of the experiment concludes that LSSVM-RBF performs best, while the BNB classifier technique performs worst, with a mean AUC rank of 28.49. After finding the best training algorithms, the investigation used the Wilcoxon signed rank test with Bonferroni correction to compare different training methods with LSSVM-RBF pairwise. The considered hypothesis for the Wilcoxon test is "*The performance values of Anti-Pattern Models using LSSVM-RBF do not differ significantly when other training methods are used*". Table 16 shows the results of the Wilcoxon test with Bonferroni correction on accuracy and AUC values, where - is used to present the accepted hypothesis, and ⧧ is used

Table 16
Hypothesis testing: Accuracy and AUC: Different machine learning techniques.

Accuracy	Rank	MNB	BNB	GNB	DT	LOGR	KNN	SVCL	SVCP	SVCR	MLPL	MLPS	MLPA	BAG	RF	EXTR	ADA	GRAB	DL1	DL2	DL3	DL4	DL5	DL6	WELS	WELSI	WELR	WELT	LSL	LSP	LSR	ELL	ELR	ELP
MNB	25.60	-	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
BNB	27.36	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
GNB	18.19	+	+	-	+	-	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
DT	9.57	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
LOGR	18.37	+	+	-	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
KNN	11.86	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
SVCL	28.30	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
SVCP	24.20	-	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+
SVCR	26.07	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
MLPL	15.34	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
MLPS	20.32	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	-	+	+
MLPA	13.63	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+
BAG	13.70	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RF	6.14	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
EXTR	5.52	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
ADA	9.20	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
GRAB	10.54	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
DL1	17.06	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
DL2	14.98	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	+	+	+	+	+	+	+	+	+	+	+	+	-
DL3	15.55	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+
DL4	18.77	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+
DL5	20.09	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	-	-	+	-	+	+	+	+	+	+	+	+
DL6	19.67	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	+	-	+	+	+	+	+	+	+	+
WELS	26.24	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+
WELSI	22.58	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	+	-	+	+	+	+	+	+	+	+
WELR	23.90	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+
WELT	23.13	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+
LSL	12.22	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+
LSP	6.29	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+
LSR	1.48	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+
ELL	20.76	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+
ELR	19.38	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+
ELP	14.99	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	-

AUC

	Rank	MNB	BNB	GNB	DT	LOGR	KNN	SVCL	SVCP	SVCR	MLPL	MLPS	MLPA	BAG	RF	EXTR	ADA	GRAB	DL1	DL2	DL3	DL4	DL5	DL6	WELS	WELSI	WELR	WELT	LSL	LSP	LSR	ELL	ELR	ELP
MNB	25.21	-	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+
BNB	28.49	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
GNB	15.51	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
DT	15.20	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
LOGR	18.11	+	+	+	+	-	+	+	+	+	-	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+
KNN	9.52	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
SVCL	27.49	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
SVCP	25.94	+	+	+	+	+	+	+	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
SVCR	25.17	+	+	+	+	+	+	+	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
MLPL	17.39	+	+	+	+	-	+	+	+	+	-	+	+	+	+	+	+	+	+	-	-	+	+	+	+	-	-	+	+	+	+	+	+	+
MLPS	22.97	-	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	+
MLPA	15.37	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+
BAG	8.08	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RF	6.42	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
EXTR	5.86	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+

(continued on next page)

Table 16 (*continued*).

ADA	10.25	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ
GRAB	11.01	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ
DL1	18.54	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ
DL2	17.08	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	-	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ
DL3	18.11	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ
DL4	20.56	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ
DL5	19.59	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ
DL6	19.23	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ
WELS	21.55	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ
WELSI	16.93	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	-	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ
WELR	17.10	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	-	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ
WELT	15.82	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ
LSL	9.66	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ
LSP	4.31	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ
LSR	1.37	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ
ELL	25.51	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ
ELR	25.20	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-	ǂ
ELP	22.45	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	ǂ	-

Table 17
Friedman test for each anti-pattern.

	Accuracy				AUC			
Rank	CLST	WORD	FST	DST	CLST	WORD	FST	DST
AP1								
Rank-1	LSSVM-RBF	GLOVE	OD	UPSAM	LSSVM-RBF	GLOVE	SIGF	BLSMOTE
Rank-2	LSSVM-Poly	TFIDF	GA	BLSMOTE	LSSVM-Poly	SKG	GA	SVMSMOTE
Rank-3	EXTR	SKG	SIGF	ORGD	EXTR	TFIDF	OD	UPSAM
Rank-4	RF	CBOW	CORR	SVMSMOTE	RF	CBOW	CORR	SMOTE
Rank-5	KNN		INFG	SMOTE	BAG		INFG	ADSYN
AP2								
Rank-1	LSSVM-RBF	GLOVE	SIGF	UPSAM	LSSVM-RBF	GLOVE	OD	UPSAM
Rank-2	EXTR	TFIDF	OD	ORGD	LSSVM-Poly	TFIDF	SIGF	BLSMOTE
Rank-3	LSSVM-Poly	SKG	GA	BLSMOTE	EXTR	SKG	GA	SMOTE
Rank-4	RF	CBOW	CORR	ADSYN	RF	CBOW	CORR	ADSYN
Rank-5	AdaB		INFG	SMOTE	BAG		INFG	SVMSMOTE
AP3								
Rank-1	LSSVM-RBF	TFIDF	CORR	UPSAM	LSSVM-RBF	TFIDF	CORR	UPSAM
Rank-2	EXTR	GLOVE	OD	ORGD	EXTR	GLOVE	OD	SVMSMOTE
Rank-3	RF	CBOW	SIGF	BLSMOTE	LSSVM-Poly	CBOW	SIGF	BLSMOTE
Rank-4	AdaB	SKG	GNR	ADSYN	RF	SKG	GA	SMOTE
Rank-5	LSSVM-Poly		GA	SMOTE	BAG		GNR	ADSYN
AP4								
Rank-1	LSSVM-RBF	TFIDF	CORR	UPSAM	LSSVM-RBF	GLOVE	CORR	UPSAM
Rank-2	EXTR	GLOVE	SIGF	ORGD	LSSVM-Poly	TFIDF	OD	SVMSMOTE
Rank-3	RF	CBOW	OD	SVMSMOTE	EXTR	CBOW	SIGF	SMOTE
Rank-4	LSSVM-Poly	SKG	GA	BLSMOTE	RF	SKG	GA	BLSMOTE
Rank-5	AdaB		CFS	SMOTE	BAG		CFS	ADSYN
AP5								
Rank-1	LSSVM-RBF	GLOVE	OD	UPSAM	LSSVM-RBF	GLOVE	SIGF	UPSAM
Rank-2	LSSVM-Poly	SKG	GA	ORGD	LSSVM-Poly	SKG	OD	BLSMOTE
Rank-3	EXTR	CBOW	SIGF	BLSMOTE	RF	CBOW	GA	SVMSMOTE
Rank-4	RF	TFIDF	CFS	SVMSMOTE	EXTR	TFIDF	CFS	SMOTE
Rank-5	DT		PCA	SMOTE	KNN		PCA	ADSYN

to represent rejected hypothesis. From Table 16, it is observed that the LSSVM-RBF outperforms significantly compared to other training methods. Therefore, the investigation recommends using an advanced version of machine learning to predict anti-patterns.

6.5. Comparison of the performance of the proposed framework with other models

The literature survey shows that the authors have used different approaches, like object-oriented methods, WSDL metrics, and some rule-based techniques, to detect the presence of anti-patterns. Hence, in this section, a comparative study has been done between the anti-pattern detection models developed using text, object-oriented, and WSDL metrics. All the anti-pattern detection models used in comparison have been trained with the same machine learning classifiers, feature selection techniques, and sampling techniques by only changing the input sets of features. Table 18 exhibits the performance variance of the developed models using text metrics in comparison to the static code and WSDL metrics approaches, and Figs. 13 and 14 demonstrate the box plot of the accuracy and AUC values for these metrics sets.

Table 18
Descriptive statistics: Different metrics sets.

	Median	Mean	Max	Min	Q3	Q1
Accuracy						
AP1						
OOM	88.17	84.59	99.27	43.36	92.04	80.00
WSDM	87.02	80.15	100.00	29.80	92.82	67.87
TXTM	84.53	82.57	100.00	38.40	91.41	77.35
AP2						
OOM	81.69	80.23	99.53	14.16	93.36	70.89
WSDM	77.01	75.15	100.00	22.73	93.33	61.44
TXTM	92.02	86.02	100.00	32.45	95.76	78.04
AP3						
OOM	93.87	90.51	100.00	28.32	97.64	87.56
WSDM	92.93	83.18	100.00	29.29	97.85	70.05
TXTM	91.84	88.11	100.00	40.05	95.96	83.33
AP4						
OOM	88.78	85.55	99.52	30.09	92.93	80.49
WSDM	90.43	81.98	100.00	26.26	94.72	71.26
TXTM	88.06	85.72	100.00	40.56	92.68	81.94
AP5						
OOM	73.41	73.32	99.76	13.27	87.38	62.04
WSDM	70.52	71.34	100.00	11.11	89.27	54.78
TXTM	73.60	74.48	100.00	27.78	88.48	63.19
AUC						
AP1						
OOM	0.90	0.85	1.00	0.22	0.96	0.79
WSDM	0.85	0.78	1.00	0.02	0.95	0.59
TXTM	0.86	0.82	1.00	0.21	0.94	0.75
AP2						
OOM	0.82	0.80	1.00	0.23	0.94	0.72
WSDM	0.71	0.72	1.00	0.11	0.93	0.50
TXTM	0.92	0.83	1.00	0.13	0.98	0.73
AP3						
OOM	0.97	0.90	1.00	0.05	0.99	0.89
WSDM	0.93	0.81	1.00	0.18	0.99	0.57
TXTM	0.94	0.88	1.00	0.12	0.98	0.85
AP4						
OOM	0.91	0.86	1.00	0.19	0.97	0.81
WSDM	0.92	0.80	1.00	0.26	0.97	0.62
TXTM	0.90	0.86	1.00	0.14	0.96	0.83
AP5						
OOM	0.69	0.69	1.00	0.28	0.87	0.52
WSDM	0.62	0.66	1.00	0.19	0.86	0.50
TXTM	0.70	0.71	1.00	0.17	0.86	0.56

Table 18 shows that, in the majority of cases, text metrics-based models perform equally or better than object-oriented and WSDL metrics with optimized computational costs as the text metrics approach eliminates the need to compute the metrics. The enhanced performance for AP2 anti-pattern data with the use of text metrics in comparison to object-oriented and WSDL metrics is shown in Figs. 13 and 14. The performed extensive experiments have also been evaluated using F-measure and G-mean parameters. The performance outcome evaluated using F-measure and G-mean for all employed 33 including machine learning and deep learning techniques with the implication of feature selection and sampling approaches is represented in Table 19, 20, 21, and 22. The conclusion drawn from the Tables is that using the text metrics-based models LSSVM-RB, EXTR, LSSVM-POLY, RF, and AdaBoost performed well after the application of sampling and feature selection methods.

6.6. Discussion of results

An approach for predicting the anti-patterns with improved performance and predictability power is proposed in the current work and involves extensive experimentation on publicly available web-services datasets consisting of 226 WSDL. Four embedding techniques were used to investigate the efficacy of metrics produced by applying text embedding techniques to WSDL description files in identifying web service anti-patterns. Additionally, the study examined the impact on the accuracy of the anti-pattern detection model with the application of 54 different combinations of various feature selection and sampling techniques (8 feature selection methods + 1 original feature * 5 data sampling approaches + 1 original data). To suggest the best combination of embedding, feature selection, data sampling, and classification techniques to detect the presence of various anti-patterns in web services, the investigation computed the top 5 ranks of all the developed models against the five data using the Friedman test. Table 17 displays the results of the Friedman test, with ranks determined based on obtained Accuracy and AUC values. The combination of LSSV RBF, GLOVE, OD, and UPSAM of classifier, embedding, sampling, and feature selection techniques scored the best rank for the data AP1 according to the Accuracy value. As the data has a class imbalance problem, the conclusion will be based on the area under the curve. Hence the effective combination of classifier, embedding, feature selection, and sampling approaches that scored the first rank based on AUC values for data AP1 are LSSV RBF, GLOVE, SIGF, and BLSMOTE. Here, the significant feature selection method yields the best results based on AUC. In both instances, it is evident that the best results were attained post-application of embedding and sampling approaches.

Table 17 depicts that the LSSV RBF consistently produced the best results in contrast to other employed 32 classification techniques for all five anti-pattern data. Overall the GLOVE embedding technique outperformed the other embedding approaches, with three out of five cases based on Accuracy, whereas four out of five cases based on AUC. Table 17 shows that the best results were obtained after applying feature selection techniques in three cases based on Accuracy and four cases based on AUC. For AP3 and AP4, applying the correlation coefficient feature selection technique led to the best results based on accuracy and AUC. Additionally, extensive experimentation indicates that sampling techniques led to the best performance in every instance. Table 17, demonstrates UPSAM sampling strategy performed better than others overall.

7. Conclusion and future scope

Early detection of anti-patterns helps in reducing efforts, expenses, and costs that ultimately improve the calibre of the software. The proposed work exemplifies how classifiers, embedding, feature selection, and sampling approaches can be combined effectively that enhance the performance and efficacy of the predicting anti-pattern model. Accuracy and AUC are used to validate the effectiveness of each technique that is used, and the Wilcoxon Sign Rank test and Friedman test are used to statistically analyze the findings of the research. The results of the research lead to the following conclusion:

- The empirical finding demonstrates that for AP3 and AP4 out of five anti-pattern data, upsampling with correlation coefficient feature selection combination outperformed
- For 80% of the input anti-pattern data, according to the Friedman test based on AUC values, the GLOVE embedding approach outperformed by earning the highest rank
- According to statistical analysis using the Friedman test, the best mean rank was obtained for all five anti-pattern data i.e. 100%, with the application LSSV RBF classification algorithm
- The TFIDF embedding approach attains the best result with the median AUC value of 0.91, whereas CBOW performed worst with the highest mean rank of 2.74
- The experimental finding demonstrates improved performance post-employment of sampling techniques

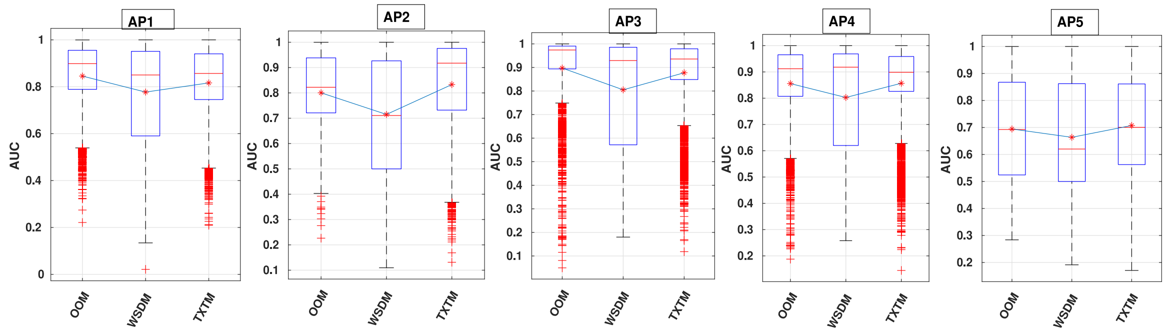


Fig. 14. AUC: box-and-whisker diagram: Different metrics sets.

Table 19
G-Mean for Anti-pattern 1: Different types of text features.

		MNB	BNB	GNB	DT	LOGR	KNN	SVCL	SVCP	SVCR	MLPL	MLPS	MLPA	BAG	RF	EXTR	ADA	GRAB
SMOTE																		
AF	TFIDF	0.71	0.85	0.77	0.89	0.81	0.89	0.75	0.76	0.77	0.90	0.83	0.92	0.86	0.90	0.91	0.88	0.86
AF	CBOW	0.75	0.20	0.81	0.91	0.76	0.87	0.70	0.74	0.71	0.45	0.82	0.00	0.82	0.92	0.94	0.90	0.93
AF	SKG	0.74	0.15	0.80	0.94	0.82	0.89	0.72	0.75	0.71	0.46	0.00	0.00	0.85	0.94	0.96	0.94	0.93
AF	GLOVE	0.75	0.13	0.79	0.91	0.89	0.87	0.75	0.77	0.71	0.45	0.00	0.00	0.80	0.94	0.94	0.93	0.92
SIGF	TFIDF	0.71	0.82	0.70	0.81	0.78	0.81	0.53	0.40	0.56	0.79	0.00	0.80	0.81	0.83	0.81	0.84	0.85
SIGF	CBOW	0.75	0.13	0.80	0.94	0.79	0.87	0.70	0.73	0.70	0.86	0.82	0.00	0.83	0.93	0.93	0.93	0.94
SIGF	SKG	0.75	0.18	0.79	0.93	0.81	0.88	0.62	0.63	0.60	0.94	0.83	0.92	0.86	0.94	0.95	0.93	0.92
SIGF	GLOVE	0.76	0.13	0.79	0.90	0.83	0.87	0.47	0.48	0.46	0.53	0.87	0.00	0.81	0.94	0.94	0.88	0.87
INFOG	TFIDF	0.75	0.82	0.61	0.86	0.77	0.86	0.60	0.56	0.61	0.88	0.77	0.85	0.84	0.86	0.86	0.87	0.88
INFOG	CBOW	0.75	0.43	0.78	0.89	0.77	0.87	0.75	0.78	0.77	0.84	0.78	0.80	0.84	0.91	0.91	0.88	0.87
INFOG	SKG	0.76	0.11	0.80	0.94	0.77	0.90	0.68	0.72	0.70	0.88	0.77	0.90	0.89	0.96	0.94	0.94	0.94
INFOG	GLOVE	0.75	0.15	0.82	0.89	0.76	0.86	0.69	0.73	0.71	0.87	0.77	0.82	0.84	0.93	0.92	0.86	0.86
GA	TFIDF	0.70	0.86	0.77	0.83	0.78	0.87	0.67	0.67	0.68	0.80	0.80	0.85	0.83	0.90	0.90	0.84	0.87
GA	CBOW	0.75	0.18	0.81	0.89	0.77	0.86	0.72	0.75	0.73	0.60	0.79	0.88	0.84	0.92	0.91	0.87	0.86
GA	SKG	0.74	0.11	0.80	0.93	0.80	0.88	0.72	0.76	0.72	0.45	0.00	0.00	0.85	0.94	0.95	0.94	0.93
GA	GLOVE	0.76	0.13	0.77	0.93	0.82	0.88	0.82	0.86	0.81	0.37	0.84	0.74	0.82	0.95	0.96	0.91	0.91
BLSMOTE																		
AF	TFIDF	0.83	0.88	0.83	0.93	0.85	0.92	0.74	0.77	0.76	0.93	0.87	0.94	0.92	0.94	0.94	0.94	0.93
AF	CBOW	0.77	0.18	0.84	0.96	0.85	0.88	0.62	0.63	0.62	0.46	0.87	0.00	0.85	0.94	0.96	0.94	0.94
AF	SKG	0.77	0.00	0.83	0.96	0.86	0.91	0.61	0.63	0.61	0.46	0.00	0.00	0.86	0.96	0.98	0.96	0.94
AF	GLOVE	0.79	0.15	0.83	0.90	0.91	0.90	0.88	0.89	0.85	0.35	0.00	0.00	0.84	0.97	0.98	0.94	0.92
SIGF	TFIDF	0.83	0.91	0.85	0.92	0.87	0.93	0.60	0.61	0.61	0.93	0.82	0.89	0.91	0.96	0.93	0.91	0.94
SIGF	CBOW	0.77	0.13	0.84	0.96	0.85	0.87	0.71	0.74	0.72	0.93	0.86	0.00	0.84	0.96	0.96	0.95	0.93
SIGF	SKG	0.77	0.15	0.83	0.95	0.84	0.90	0.65	0.67	0.65	0.98	0.89	0.94	0.86	0.97	0.98	0.95	0.94
SIGF	GLOVE	0.79	0.13	0.82	0.95	0.87	0.90	0.64	0.65	0.64	0.55	0.90	0.00	0.85	0.97	0.97	0.90	0.93
INFOG	TFIDF	0.79	0.85	0.59	0.90	0.82	0.84	0.61	0.51	0.61	0.88	0.81	0.86	0.83	0.90	0.88	0.89	0.88
INFOG	CBOW	0.78	0.42	0.83	0.94	0.81	0.91	0.73	0.75	0.73	0.92	0.82	0.86	0.88	0.95	0.95	0.92	0.93
INFOG	SKG	0.78	0.37	0.85	0.95	0.79	0.93	0.70	0.75	0.73	0.90	0.80	0.90	0.92	0.98	0.97	0.96	0.96
INFOG	GLOVE	0.79	0.38	0.85	0.94	0.81	0.92	0.72	0.76	0.75	0.92	0.84	0.87	0.89	0.96	0.97	0.91	0.91
GA	TFIDF	0.83	0.90	0.85	0.90	0.85	0.90	0.47	0.46	0.45	0.93	0.87	0.91	0.89	0.95	0.93	0.91	0.91
GA	CBOW	0.77	0.07	0.86	0.94	0.83	0.89	0.71	0.76	0.73	0.44	0.83	0.90	0.85	0.95	0.95	0.94	0.90
GA	SKG	0.77	0.11	0.83	0.95	0.83	0.90	0.61	0.63	0.61	0.44	0.00	0.00	0.85	0.96	0.95	0.96	0.95
GA	GLOVE	0.79	0.13	0.80	0.95	0.87	0.90	0.63	0.65	0.62	0.46	0.88	0.95	0.84	0.96	0.97	0.94	0.92
SVMSMOTE																		
AF	TFIDF	0.73	0.87	0.81	0.92	0.85	0.91	0.87	0.92	0.91	0.91	0.87	0.89	0.89	0.93	0.92	0.94	0.92
AF	CBOW	0.77	0.16	0.82	0.95	0.86	0.88	0.85	0.89	0.86	0.46	0.87	0.00	0.85	0.94	0.94	0.92	0.93
AF	SKG	0.77	0.15	0.82	0.92	0.88	0.92	0.87	0.92	0.85	0.00	0.90	0.00	0.89	0.95	0.95	0.94	0.91
AF	GLOVE	0.79	0.30	0.81	0.89	0.90	0.90	0.91	0.96	0.90	0.00	0.63	0.00	0.87	0.94	0.95	0.93	0.92
SIGF	TFIDF	0.65	0.90	0.82	0.90	0.84	0.91	0.85	0.90	0.90	0.91	0.00	0.89	0.90	0.93	0.93	0.89	0.94
SIGF	CBOW	0.77	0.13	0.83	0.93	0.86	0.88	0.85	0.89	0.85	0.88	0.86	0.00	0.84	0.95	0.95	0.93	0.91
SIGF	SKG	0.77	0.15	0.82	0.93	0.86	0.91	0.84	0.91	0.83	0.93	0.88	0.89	0.89	0.95	0.95	0.94	0.91
SIGF	GLOVE	0.80	0.15	0.81	0.92	0.89	0.89	0.91	0.95	0.86	0.00	0.00	0.00	0.87	0.94	0.95	0.89	0.92
INFOG	TFIDF	0.75	0.80	0.78	0.92	0.77	0.87	0.75	0.71	0.75	0.91	0.77	0.85	0.88	0.91	0.91	0.90	0.91
INFOG	CBOW	0.81	0.00	0.80	0.91	0.84	0.90	0.79	0.85	0.83	0.89	0.85	0.83	0.88	0.93	0.94	0.92	0.88
INFOG	SKG	0.78	0.45	0.85	0.94	0.79	0.93	0.71	0.77	0.74	0.91	0.81	0.89	0.92	0.97	0.97	0.94	0.94
INFOG	GLOVE	0.80	0.34	0.87	0.94	0.84	0.92	0.63	0.63	0.60	0.91	0.85	0.87	0.91	0.96	0.96	0.91	0.91
GA	TFIDF	0.61	0.88	0.83	0.88	0.89	0.89	0.87	0.91	0.91	0.90	0.88	0.91	0.87	0.93	0.92	0.92	0.91
GA	CBOW	0.77	0.09	0.84	0.91	0.83	0.89	0.83	0.88	0.85	0.92	0.81	0.85	0.84	0.92	0.94	0.89	0.89
GA	SKG	0.77	0.13	0.83	0.93	0.87	0.91	0.83	0.93	0.84	0.00	0.00	0.43	0.89	0.95	0.95	0.94	0.94
GA	GLOVE	0.78	0.09	0.78	0.90	0.88	0.92	0.90	0.96	0.85	0.00	0.70	0.54	0.86	0.95	0.97	0.90	0.90

(*continued on next page*)

Table 19 (*continued*).

ADSYN																		
AF	TFIDF	0.70	0.72	0.69	0.89	0.77	0.85	0.72	0.75	0.74	0.79	0.73	0.87	0.82	0.88	0.88	0.82	0.82
AF	CBOW	0.72	0.20	0.77	0.92	0.77	0.88	0.72	0.78	0.73	0.42	0.83	0.00	0.82	0.93	0.94	0.92	0.92
AF	SKG	0.72	0.10	0.78	0.93	0.82	0.90	0.47	0.48	0.47	0.36	0.00	0.00	0.86	0.96	0.96	0.93	0.93
AF	GLOVE	0.72	0.13	0.75	0.89	0.86	0.87	0.65	0.66	0.63	0.44	0.00	0.00	0.81	0.93	0.96	0.91	0.90
SIGF	TFIDF	0.65	0.76	0.67	0.84	0.71	0.81	0.58	0.53	0.60	0.80	0.35	0.80	0.79	0.85	0.84	0.78	0.77
SIGF	CBOW	0.73	0.11	0.76	0.92	0.74	0.87	0.58	0.62	0.59	0.82	0.83	0.00	0.80	0.94	0.94	0.89	0.89
SIGF	SKG	0.72	0.17	0.75	0.93	0.82	0.90	0.78	0.84	0.76	0.92	0.78	0.92	0.86	0.94	0.96	0.91	0.92
SIGF	GLOVE	0.73	0.13	0.76	0.88	0.82	0.88	0.72	0.76	0.71	0.45	0.87	0.59	0.80	0.92	0.96	0.83	0.84
INFOG	TFIDF	0.79	0.81	0.53	0.83	0.76	0.78	0.67	0.63	0.67	0.86	0.75	0.84	0.80	0.85	0.82	0.85	0.84
INFOG	CBOW	0.73	0.41	0.74	0.92	0.75	0.87	0.68	0.70	0.71	0.81	0.76	0.77	0.82	0.91	0.91	0.87	0.86
INFOG	SKG	0.70	0.37	0.74	0.94	0.72	0.89	0.63	0.66	0.64	0.85	0.72	0.86	0.87	0.95	0.96	0.91	0.91
INFOG	GLOVE	0.72	0.13	0.80	0.88	0.72	0.86	0.58	0.64	0.61	0.81	0.73	0.83	0.81	0.91	0.95	0.84	0.84
GA	TFIDF	0.72	0.82	0.74	0.88	0.78	0.85	0.60	0.58	0.62	0.89	0.80	0.85	0.83	0.91	0.91	0.86	0.86
GA	CBOW	0.72	0.13	0.77	0.90	0.77	0.87	0.74	0.80	0.77	0.83	0.78	0.90	0.82	0.93	0.94	0.87	0.90
GA	SKG	0.71	0.11	0.77	0.93	0.79	0.89	0.61	0.64	0.60	0.45	0.00	0.00	0.85	0.95	0.95	0.93	0.90
GA	GLOVE	0.72	0.15	0.75	0.91	0.81	0.88	0.74	0.79	0.73	0.44	0.82	0.73	0.81	0.96	0.97	0.91	0.89

Table 20
G-Mean for Anti-pattern 1: Different types of text features.

		DL1	DL2	DL3	DL4	DL5	DL6	WELS	WELSI	WELR	WELT	LSL	LSP	LSR	ELL	ELR	ELP
SMOTE																	
AF	TFIDF	0.81	0.84	0.85	0.85	0.78	0.57	0.78	0.89	0.78	0.93	0.84	0.95	0.99	0.77	0.80	0.89
AF	CBOW	0.80	0.81	0.82	0.81	0.87	0.87	0.85	0.90	0.85	0.91	1.00	1.00	1.00	0.78	0.84	0.90
AF	SKG	0.85	0.88	0.88	0.87	0.84	0.87	0.85	0.89	0.85	0.90	0.98	1.00	1.00	0.79	0.84	0.92
AF	GLOVE	0.88	0.91	0.86	0.88	0.90	0.92	0.86	0.92	0.86	0.86	1.00	1.00	1.00	0.82	0.84	0.95
SIGF	TFIDF	0.79	0.66	0.60	0.50	0.54	0.61	0.78	0.95	0.78	0.96	0.77	0.83	0.88	0.79	0.78	0.78
SIGF	CBOW	0.82	0.82	0.80	0.81	0.85	0.83	0.83	0.91	0.83	0.92	1.00	1.00	1.00	0.78	0.83	0.88
SIGF	SKG	0.84	0.88	0.85	0.87	0.88	0.90	0.83	0.90	0.83	0.89	1.00	1.00	1.00	0.77	0.83	0.92
SIGF	GLOVE	0.85	0.86	0.87	0.85	0.92	0.90	0.86	0.91	0.86	0.92	0.99	1.00	1.00	0.78	0.82	0.90
INFOG	TFIDF	0.79	0.81	0.81	0.81	0.74	0.78	0.78	0.88	0.78	0.92	0.77	0.85	0.92	0.76	0.77	0.78
INFOG	CBOW	0.78	0.80	0.81	0.69	0.76	0.79	0.81	0.98	0.81	0.97	0.78	0.83	1.00	0.75	0.76	0.77
INFOG	SKG	0.77	0.81	0.83	0.68	0.67	0.60	0.80	0.95	0.80	0.97	0.76	0.83	0.99	0.76	0.76	0.76
INFOG	GLOVE	0.79	0.79	0.80	0.80	0.67	0.65	0.80	0.93	0.80	0.96	0.76	0.84	1.00	0.74	0.76	0.76
GA	TFIDF	0.78	0.79	0.80	0.67	0.62	0.61	0.77	0.92	0.77	0.92	0.78	0.88	0.99	0.69	0.79	0.81
GA	CBOW	0.81	0.82	0.83	0.81	0.86	0.84	0.84	0.89	0.84	0.95	1.00	1.00	1.00	0.76	0.82	0.84
GA	SKG	0.82	0.85	0.86	0.86	0.87	0.86	0.83	0.92	0.83	0.98	0.93	0.99	1.00	0.77	0.81	0.89
GA	GLOVE	0.87	0.87	0.86	0.85	0.93	0.88	0.86	0.92	0.86	0.93	0.97	1.00	1.00	0.80	0.82	0.91
BLSMOTE																	
AF	TFIDF	0.86	0.89	0.88	0.89	0.67	0.93	0.84	0.89	0.84	0.92	0.89	0.93	1.00	0.82	0.86	0.91
AF	CBOW	0.87	0.88	0.87	0.88	0.87	0.87	0.89	0.90	0.89	0.94	0.99	1.00	1.00	0.81	0.84	0.89
AF	SKG	0.90	0.87	0.87	0.87	0.91	0.90	0.91	0.87	0.91	0.95	0.98	1.00	1.00	0.80	0.85	0.92
AF	GLOVE	0.92	0.92	0.92	0.90	0.94	0.95	0.91	0.88	0.91	0.93	0.98	1.00	1.00	0.84	0.88	0.94
SIGF	TFIDF	0.88	0.88	0.69	0.69	0.56	0.46	0.87	0.91	0.87	0.94	0.88	0.91	0.99	0.88	0.88	0.88
SIGF	CBOW	0.87	0.87	0.87	0.85	0.87	0.88	0.89	0.93	0.89	0.97	1.00	0.96	1.00	0.81	0.84	0.87
SIGF	SKG	0.88	0.89	0.90	0.88	0.86	0.87	0.90	0.89	0.90	0.95	0.97	1.00	1.00	0.79	0.85	0.91
SIGF	GLOVE	0.91	0.90	0.88	0.91	0.95	0.94	0.89	0.93	0.89	0.93	0.98	1.00	1.00	0.82	0.86	0.94
INFOG	TFIDF	0.82	0.82	0.83	0.83	0.86	0.71	0.82	0.84	0.82	0.92	0.80	0.84	0.93	0.79	0.78	0.79
INFOG	CBOW	0.82	0.84	0.85	0.85	0.77	0.62	0.85	0.97	0.85	0.97	0.82	0.86	1.00	0.79	0.79	0.81
INFOG	SKG	0.80	0.85	0.87	0.89	0.46	0.69	0.85	0.97	0.85	0.97	0.81	0.84	1.00	0.78	0.78	0.78
INFOG	GLOVE	0.86	0.86	0.78	0.86	0.64	0.69	0.83	0.96	0.83	0.96	0.79	0.89	1.00	0.79	0.80	0.81
GA	TFIDF	0.85	0.89	0.89	0.88	0.81	0.81	0.85	0.93	0.85	0.94	0.87	0.92	0.98	0.82	0.85	0.88
GA	CBOW	0.85	0.89	0.89	0.88	0.87	0.89	0.88	0.91	0.88	0.98	1.00	1.00	1.00	0.80	0.83	0.84
GA	SKG	0.89	0.89	0.89	0.90	0.90	0.90	0.87	0.92	0.87	0.96	0.94	1.00	1.00	0.80	0.83	0.91
GA	GLOVE	0.90	0.92	0.91	0.91	0.93	0.93	0.88	0.90	0.88	0.93	0.96	1.00	1.00	0.81	0.83	0.92
SVMSMOTE																	
AF	TFIDF	0.84	0.86	0.88	0.87	0.90	0.91	0.81	0.89	0.81	0.91	0.90	0.94	1.00	0.85	0.86	0.88
AF	CBOW	0.88	0.87	0.86	0.85	0.83	0.88	0.83	0.92	0.83	0.93	1.00	0.99	1.00	0.83	0.85	0.91
AF	SKG	0.88	0.90	0.91	0.88	0.88	0.86	0.85	0.89	0.85	0.90	0.98	1.00	1.00	0.84	0.87	0.93
AF	GLOVE	0.88	0.90	0.90	0.89	0.93	0.91	0.87	0.90	0.87	0.91	0.98	1.00	1.00	0.89	0.86	0.93
SIGF	TFIDF	0.83	0.85	0.73	0.33	0.87	0.69	0.85	0.91	0.85	0.95	0.87	0.92	0.98	0.82	0.85	0.89
SIGF	CBOW	0.87	0.85	0.86	0.83	0.85	0.87	0.83	0.93	0.83	0.94	1.00	1.00	1.00	0.84	0.85	0.91
SIGF	SKG	0.90	0.87	0.86	0.87	0.89	0.91	0.84	0.90	0.84	0.92	0.99	1.00	1.00	0.85	0.88	0.92
SIGF	GLOVE	0.92	0.89	0.88	0.86	0.93	0.86	0.80	0.91	0.80	0.90	0.97	1.00	1.00	0.87	0.89	0.93
INFOG	TFIDF	0.80	0.83	0.81	0.84	0.74	0.78	0.79	0.78	0.79	0.85	0.83	0.87	0.95	0.82	0.78	0.83
INFOG	CBOW	0.80	0.84	0.84	0.76	0.85	0.87	0.77	0.97	0.77	0.99	0.84	0.85	1.00	0.84	0.84	0.82
INFOG	SKG	0.82	0.83	0.87	0.89	0.89	0.91	0.84	0.96	0.84	0.98	0.81	0.88	1.00	0.77	0.78	0.78
INFOG	GLOVE	0.86	0.85	0.86	0.78	0.64	0.86	0.82	0.90	0.82	0.92	0.82	0.93	1.00	0.81	0.84	0.85
GA	TFIDF	0.89	0.90	0.88	0.89	0.85	0.90	0.78	0.92	0.78	0.94	0.89	0.93	1.00	0.87	0.88	0.90
GA	CBOW	0.85	0.87	0.86	0.86	0.89	0.86	0.81	0.91	0.81	0.97	0.99	0.98	1.00	0.83	0.85	0.88
GA	SKG	0.87	0.90	0.89	0.87	0.91	0.90	0.82	0.91	0.82	0.95	0.98	1.00	1.00	0.85	0.85	0.92
GA	GLOVE	0.89	0.86	0.89	0.88	0.87	0.94	0.84	0.89	0.84	0.93	0.95	1.00	1.00	0.83	0.88	0.93

(*continued on next page*)

Table 20 (*continued*).

ADSYN																	
AF	TFIDF	0.76	0.78	0.78	0.78	0.81	0.80	0.76	0.88	0.76	0.92	0.81	0.95	1.00	0.72	0.77	0.82
AF	CBOW	0.82	0.83	0.83	0.82	0.83	0.83	0.84	0.91	0.84	0.91	1.00	1.00	1.00	0.77	0.82	0.90
AF	SKG	0.88	0.87	0.86	0.83	0.91	0.92	0.82	0.92	0.82	0.88	0.98	1.00	1.00	0.76	0.84	0.92
AF	GLOVE	0.88	0.87	0.86	0.80	0.91	0.86	0.85	0.89	0.85	0.88	0.99	1.00	1.00	0.81	0.83	0.95
SIGF	TFIDF	0.71	0.75	0.65	0.69	0.47	0.71	0.75	0.86	0.75	0.94	0.73	0.82	0.93	0.55	0.73	0.73
SIGF	CBOW	0.82	0.83	0.85	0.83	0.86	0.84	0.81	0.91	0.81	0.88	1.00	0.91	1.00	0.75	0.84	0.89
SIGF	SKG	0.84	0.89	0.82	0.84	0.87	0.88	0.82	0.91	0.82	0.89	0.99	0.98	1.00	0.76	0.81	0.91
SIGF	GLOVE	0.86	0.86	0.84	0.82	0.90	0.93	0.82	0.91	0.82	0.91	0.98	1.00	1.00	0.78	0.81	0.92
INFOG	TFIDF	0.79	0.81	0.80	0.78	0.77	0.75	0.79	0.89	0.79	0.93	0.77	0.86	0.88	0.74	0.75	0.78
INFOG	CBOW	0.76	0.77	0.77	0.76	0.59	0.72	0.76	0.97	0.76	0.98	0.76	0.90	1.00	0.74	0.74	0.76
INFOG	SKG	0.72	0.74	0.74	0.68	0.80	0.77	0.74	0.96	0.74	0.97	0.76	0.81	0.99	0.72	0.72	0.74
INFOG	GLOVE	0.73	0.75	0.75	0.74	0.63	0.68	0.78	0.96	0.78	0.95	0.76	0.81	1.00	0.72	0.72	0.73
GA	TFIDF	0.76	0.79	0.81	0.74	0.83	0.55	0.75	0.90	0.75	0.91	0.81	0.84	0.99	0.73	0.78	0.81
GA	CBOW	0.80	0.82	0.84	0.83	0.87	0.86	0.80	0.90	0.80	0.91	1.00	1.00	1.00	0.76	0.83	0.83
GA	SKG	0.82	0.86	0.89	0.83	0.88	0.91	0.80	0.93	0.80	0.89	0.98	1.00	1.00	0.76	0.81	0.91
GA	GLOVE	0.84	0.86	0.86	0.86	0.93	0.94	0.83	0.87	0.83	0.87	0.98	1.00	1.00	0.78	0.82	0.90

Table 21
F-Score for Anti-pattern 1: Different types of text features.

		MNB	BNB	GNB	DT	LOGR	KNN	SVCL	SVCP	SVCR	MLPL	MLPS	MLPA	BAG	RF	EXTR	ADA	GRAB
SMOTE																		
AF	TFIDF	0.73	0.86	0.79	0.89	0.83	0.90	0.73	0.73	0.74	0.90	0.84	0.92	0.86	0.90	0.91	0.89	0.86
AF	CBOW	0.81	0.67	0.84	0.92	0.80	0.89	0.67	0.71	0.68	0.40	0.85	0.00	0.85	0.92	0.94	0.90	0.93
AF	SKG	0.80	0.67	0.83	0.94	0.85	0.90	0.70	0.73	0.68	0.41	0.00	0.00	0.87	0.94	0.96	0.94	0.93
AF	GLOVE	0.80	0.67	0.81	0.91	0.90	0.89	0.73	0.75	0.68	0.40	0.00	0.00	0.84	0.94	0.94	0.93	0.92
SIGF	TFIDF	0.77	0.82	0.66	0.81	0.76	0.80	0.44	0.27	0.48	0.78	0.66	0.79	0.81	0.82	0.80	0.84	0.85
SIGF	CBOW	0.81	0.67	0.83	0.94	0.82	0.88	0.67	0.71	0.67	0.88	0.84	0.00	0.86	0.93	0.93	0.93	0.94
SIGF	SKG	0.81	0.67	0.82	0.93	0.84	0.90	0.56	0.58	0.53	0.95	0.85	0.92	0.88	0.95	0.95	0.93	0.93
SIGF	GLOVE	0.81	0.67	0.82	0.90	0.85	0.89	0.36	0.38	0.35	0.64	0.88	0.00	0.84	0.94	0.94	0.88	0.87
INFOG	TFIDF	0.75	0.85	0.55	0.86	0.79	0.87	0.54	0.48	0.54	0.89	0.80	0.87	0.85	0.86	0.86	0.88	0.89
INFOG	CBOW	0.81	0.50	0.82	0.89	0.81	0.88	0.77	0.79	0.78	0.86	0.82	0.83	0.85	0.91	0.91	0.88	0.87
INFOG	SKG	0.81	0.67	0.84	0.95	0.81	0.91	0.66	0.69	0.67	0.89	0.81	0.91	0.90	0.96	0.95	0.94	0.94
INFOG	GLOVE	0.80	0.67	0.85	0.89	0.80	0.87	0.67	0.70	0.68	0.88	0.81	0.83	0.86	0.93	0.92	0.86	0.86
GA	TFIDF	0.75	0.86	0.79	0.83	0.80	0.87	0.63	0.62	0.63	0.83	0.81	0.85	0.83	0.90	0.90	0.85	0.87
GA	CBOW	0.81	0.67	0.84	0.89	0.81	0.88	0.69	0.73	0.70	0.58	0.82	0.90	0.86	0.92	0.92	0.88	0.86
GA	SKG	0.80	0.67	0.83	0.93	0.84	0.90	0.69	0.74	0.70	0.51	0.67	0.67	0.87	0.95	0.95	0.94	0.93
GA	GLOVE	0.81	0.67	0.80	0.93	0.85	0.90	0.81	0.85	0.81	0.25	0.86	0.72	0.85	0.95	0.96	0.92	0.92
BLSMOTE																		
AF	TFIDF	0.85	0.89	0.85	0.93	0.87	0.92	0.72	0.74	0.74	0.93	0.88	0.94	0.92	0.94	0.94	0.95	0.93
AF	CBOW	0.83	0.67	0.87	0.96	0.87	0.89	0.56	0.57	0.56	0.52	0.89	0.00	0.87	0.95	0.96	0.94	0.95
AF	SKG	0.83	0.67	0.86	0.96	0.88	0.91	0.54	0.57	0.55	0.52	0.00	0.00	0.88	0.96	0.98	0.96	0.94
AF	GLOVE	0.84	0.67	0.86	0.90	0.91	0.91	0.88	0.89	0.85	0.23	0.00	0.00	0.87	0.97	0.98	0.94	0.92
SIGF	TFIDF	0.84	0.92	0.86	0.92	0.88	0.93	0.53	0.54	0.54	0.93	0.82	0.90	0.91	0.96	0.93	0.91	0.94
SIGF	CBOW	0.83	0.67	0.86	0.96	0.87	0.89	0.69	0.71	0.69	0.94	0.88	0.00	0.87	0.96	0.96	0.95	0.93
SIGF	SKG	0.83	0.67	0.86	0.95	0.87	0.91	0.60	0.61	0.60	0.98	0.91	0.94	0.88	0.97	0.98	0.95	0.94
SIGF	GLOVE	0.84	0.67	0.85	0.95	0.89	0.91	0.58	0.60	0.58	0.55	0.91	0.00	0.87	0.97	0.97	0.91	0.93
INFOG	TFIDF	0.81	0.88	0.51	0.90	0.85	0.85	0.55	0.42	0.55	0.89	0.85	0.89	0.84	0.90	0.88	0.90	0.89
INFOG	CBOW	0.83	0.49	0.86	0.94	0.85	0.92	0.71	0.73	0.71	0.93	0.86	0.88	0.90	0.95	0.95	0.92	0.93
INFOG	SKG	0.83	0.60	0.87	0.95	0.84	0.93	0.68	0.73	0.70	0.91	0.85	0.91	0.93	0.98	0.97	0.96	0.96
INFOG	GLOVE	0.84	0.60	0.87	0.94	0.84	0.92	0.70	0.73	0.72	0.92	0.86	0.88	0.90	0.96	0.97	0.91	0.91
GA	TFIDF	0.86	0.90	0.87	0.90	0.87	0.91	0.36	0.34	0.34	0.94	0.88	0.91	0.90	0.95	0.93	0.92	0.91
GA	CBOW	0.83	0.67	0.88	0.94	0.86	0.90	0.69	0.73	0.71	0.39	0.86	0.91	0.88	0.95	0.95	0.94	0.91
GA	SKG	0.83	0.67	0.86	0.95	0.86	0.91	0.55	0.57	0.55	0.51	0.67	0.67	0.88	0.96	0.95	0.96	0.95
GA	GLOVE	0.84	0.67	0.82	0.95	0.89	0.91	0.57	0.59	0.56	0.52	0.90	0.95	0.87	0.96	0.97	0.95	0.92
SVMSMOTE																		
AF	TFIDF	0.67	0.83	0.77	0.90	0.81	0.88	0.83	0.90	0.88	0.89	0.83	0.86	0.86	0.92	0.90	0.92	0.90
AF	CBOW	0.74	0.04	0.78	0.94	0.82	0.84	0.81	0.86	0.82	0.34	0.83	0.00	0.81	0.92	0.91	0.89	0.91
AF	SKG	0.74	0.04	0.78	0.89	0.84	0.89	0.83	0.89	0.81	0.00	0.87	0.00	0.85	0.93	0.93	0.91	0.88
AF	GLOVE	0.76	0.18	0.76	0.86	0.87	0.86	0.89	0.94	0.87	0.00	0.56	0.00	0.82	0.92	0.94	0.91	0.90
SIGF	TFIDF	0.57	0.86	0.77	0.87	0.80	0.88	0.82	0.87	0.87	0.88	0.00	0.86	0.87	0.92	0.91	0.87	0.92
SIGF	CBOW	0.74	0.03	0.78	0.92	0.81	0.84	0.81	0.85	0.81	0.84	0.82	0.00	0.79	0.93	0.94	0.90	0.89
SIGF	SKG	0.74	0.04	0.78	0.91	0.81	0.87	0.79	0.88	0.78	0.92	0.84	0.86	0.85	0.94	0.94	0.92	0.88
SIGF	GLOVE	0.76	0.04	0.77	0.89	0.85	0.85	0.88	0.93	0.82	0.00	0.00	0.00	0.83	0.93	0.94	0.86	0.89
INFOG	TFIDF	0.72	0.80	0.75	0.92	0.77	0.88	0.74	0.68	0.76	0.91	0.77	0.86	0.88	0.91	0.91	0.90	0.91
INFOG	CBOW	0.76	0.00	0.76	0.88	0.80	0.87	0.76	0.81	0.79	0.85	0.81	0.79	0.85	0.90	0.92	0.90	0.84
INFOG	SKG	0.83	0.51	0.87	0.94	0.84	0.93	0.69	0.75	0.72	0.92	0.85	0.90	0.92	0.97	0.97	0.94	0.94
INFOG	GLOVE	0.83	0.58	0.89	0.94	0.86	0.92	0.57	0.57	0.53	0.91	0.86	0.88	0.92	0.96	0.96	0.92	0.91
GA	TFIDF	0.53	0.85	0.78	0.85	0.85	0.86	0.83	0.88	0.88	0.87	0.85	0.88	0.84	0.91	0.90	0.89	0.89
GA	CBOW	0.74	0.02	0.80	0.88	0.78	0.85	0.78	0.84	0.81	0.89	0.76	0.80	0.80	0.89	0.92	0.85	0.86
GA	SKG	0.74	0.03	0.78	0.91	0.83	0.87	0.79	0.90	0.80	0.00	0.00	0.30	0.85	0.94	0.94	0.91	0.91
GA	GLOVE	0.74	0.01	0.74	0.87	0.84	0.88	0.87	0.94	0.81	0.00	0.63	0.45	0.82	0.94	0.95	0.88	0.87

(*continued on next page*)

Table 21 (*continued*).

ADSYN																		
AF	TFIDF	0.69	0.71	0.70	0.89	0.78	0.86	0.70	0.72	0.72	0.83	0.74	0.88	0.83	0.88	0.88	0.81	0.82
AF	CBOW	0.77	0.67	0.80	0.93	0.80	0.90	0.69	0.76	0.71	0.37	0.86	0.00	0.84	0.93	0.94	0.93	0.93
AF	SKG	0.78	0.67	0.81	0.93	0.85	0.91	0.36	0.37	0.36	0.59	0.00	0.00	0.88	0.96	0.96	0.94	0.94
AF	GLOVE	0.77	0.67	0.78	0.89	0.88	0.89	0.60	0.61	0.58	0.50	0.00	0.00	0.84	0.93	0.96	0.92	0.90
SIGF	TFIDF	0.66	0.76	0.63	0.84	0.72	0.82	0.52	0.44	0.54	0.80	0.25	0.81	0.79	0.85	0.84	0.78	0.78
SIGF	CBOW	0.77	0.67	0.80	0.92	0.78	0.89	0.51	0.56	0.52	0.86	0.86	0.00	0.83	0.94	0.95	0.90	0.90
SIGF	SKG	0.77	0.67	0.79	0.93	0.85	0.91	0.78	0.83	0.77	0.93	0.80	0.93	0.88	0.94	0.96	0.91	0.92
SIGF	GLOVE	0.77	0.67	0.78	0.88	0.85	0.90	0.69	0.74	0.69	0.40	0.89	0.52	0.84	0.92	0.96	0.84	0.85
INFOG	TFIDF	0.83	0.83	0.44	0.83	0.80	0.79	0.64	0.58	0.65	0.88	0.80	0.87	0.82	0.85	0.82	0.86	0.87
INFOG	CBOW	0.78	0.36	0.78	0.92	0.79	0.89	0.65	0.68	0.68	0.84	0.79	0.81	0.83	0.91	0.91	0.87	0.86
INFOG	SKG	0.75	0.45	0.78	0.94	0.76	0.90	0.57	0.62	0.59	0.86	0.76	0.87	0.88	0.95	0.96	0.91	0.91
INFOG	GLOVE	0.77	0.67	0.82	0.88	0.77	0.88	0.51	0.58	0.55	0.83	0.77	0.85	0.83	0.91	0.95	0.84	0.85
GA	TFIDF	0.74	0.82	0.73	0.88	0.79	0.87	0.54	0.50	0.55	0.90	0.80	0.85	0.83	0.91	0.91	0.86	0.86
GA	CBOW	0.77	0.67	0.80	0.90	0.81	0.88	0.75	0.80	0.77	0.82	0.81	0.91	0.85	0.93	0.94	0.87	0.90
GA	SKG	0.77	0.67	0.80	0.93	0.83	0.91	0.54	0.59	0.54	0.51	0.67	0.67	0.87	0.95	0.95	0.94	0.90
GA	GLOVE	0.76	0.67	0.78	0.91	0.83	0.90	0.72	0.77	0.70	0.38	0.84	0.70	0.85	0.96	0.97	0.92	0.89

Table 22
F-Score for Anti-pattern 1: Different types of text features.

SMOTE																	
		DL1	DL2	DL3	DL4	DL5	DL6	WELS	WELSI	WELR	WELT	LSL	LSP	LSR	ELL	ELR	ELP
AF	TFIDF	0.82	0.84	0.86	0.85	0.81	0.53	0.79	0.91	0.79	0.94	0.85	0.95	0.99	0.81	0.82	0.89
AF	CBOW	0.83	0.84	0.85	0.83	0.88	0.88	0.86	0.93	0.86	0.93	1.00	1.00	1.00	0.82	0.87	0.92
AF	SKG	0.88	0.90	0.89	0.89	0.85	0.89	0.85	0.92	0.85	0.91	0.98	1.00	1.00	0.83	0.87	0.93
AF	GLOVE	0.89	0.91	0.88	0.88	0.90	0.93	0.86	0.94	0.86	0.88	1.00	1.00	1.00	0.85	0.87	0.95
SIGF	TFIDF	0.77	0.73	0.63	0.60	0.53	0.57	0.75	0.95	0.75	0.95	0.75	0.82	0.88	0.81	0.77	0.77
SIGF	CBOW	0.84	0.84	0.83	0.84	0.86	0.86	0.83	0.94	0.83	0.94	1.00	1.00	1.00	0.82	0.86	0.90
SIGF	SKG	0.86	0.89	0.87	0.89	0.89	0.91	0.84	0.94	0.84	0.92	1.00	1.00	1.00	0.82	0.86	0.93
SIGF	GLOVE	0.86	0.87	0.87	0.85	0.92	0.90	0.86	0.94	0.86	0.93	0.99	1.00	1.00	0.83	0.85	0.92
INFOG	TFIDF	0.81	0.83	0.83	0.84	0.79	0.78	0.79	0.91	0.79	0.93	0.82	0.87	0.92	0.81	0.81	0.82
INFOG	CBOW	0.82	0.83	0.84	0.78	0.76	0.78	0.82	0.99	0.82	0.98	0.82	0.85	1.00	0.81	0.81	0.81
INFOG	SKG	0.81	0.83	0.85	0.70	0.63	0.58	0.81	0.97	0.81	0.98	0.81	0.85	0.99	0.81	0.81	0.81
INFOG	GLOVE	0.82	0.82	0.81	0.82	0.67	0.66	0.81	0.96	0.81	0.97	0.80	0.86	1.00	0.80	0.81	0.81
GA	TFIDF	0.80	0.80	0.81	0.67	0.66	0.57	0.78	0.93	0.78	0.94	0.80	0.89	0.99	0.76	0.81	0.82
GA	CBOW	0.84	0.84	0.85	0.83	0.87	0.85	0.84	0.94	0.84	0.97	1.00	1.00	1.00	0.81	0.85	0.86
GA	SKG	0.84	0.87	0.88	0.88	0.89	0.88	0.84	0.95	0.84	0.98	0.94	0.99	1.00	0.82	0.84	0.91
GA	GLOVE	0.88	0.88	0.86	0.86	0.93	0.88	0.86	0.95	0.86	0.94	0.98	1.00	1.00	0.84	0.85	0.92
BLSMOTE																	
AF	TFIDF	0.87	0.89	0.89	0.89	0.77	0.93	0.84	0.92	0.84	0.94	0.90	0.93	1.00	0.85	0.87	0.91
AF	CBOW	0.88	0.89	0.89	0.90	0.89	0.89	0.89	0.94	0.89	0.95	0.99	1.00	1.00	0.85	0.87	0.90
AF	SKG	0.91	0.89	0.88	0.89	0.92	0.91	0.91	0.92	0.91	0.96	0.98	1.00	1.00	0.84	0.87	0.93
AF	GLOVE	0.92	0.92	0.92	0.90	0.94	0.95	0.91	0.92	0.91	0.93	0.98	1.00	1.00	0.87	0.89	0.94
SIGF	TFIDF	0.89	0.89	0.77	0.65	0.56	0.51	0.88	0.91	0.88	0.94	0.88	0.91	0.99	0.89	0.88	0.88
SIGF	CBOW	0.89	0.89	0.89	0.88	0.88	0.90	0.89	0.95	0.89	0.97	1.00	0.97	1.00	0.85	0.87	0.89
SIGF	SKG	0.89	0.90	0.91	0.90	0.87	0.88	0.90	0.93	0.90	0.96	0.98	1.00	1.00	0.84	0.87	0.92
SIGF	GLOVE	0.92	0.90	0.90	0.91	0.95	0.94	0.89	0.96	0.89	0.94	0.98	1.00	1.00	0.85	0.88	0.95
INFOG	TFIDF	0.85	0.86	0.87	0.86	0.88	0.73	0.81	0.87	0.81	0.95	0.85	0.87	0.93	0.84	0.84	0.84
INFOG	CBOW	0.86	0.87	0.88	0.88	0.82	0.60	0.84	0.98	0.84	0.98	0.86	0.88	1.00	0.84	0.84	0.85
INFOG	SKG	0.85	0.88	0.89	0.90	0.52	0.79	0.84	0.98	0.84	0.98	0.85	0.87	1.00	0.83	0.83	0.83
INFOG	GLOVE	0.87	0.87	0.77	0.87	0.62	0.70	0.82	0.97	0.82	0.97	0.84	0.90	1.00	0.84	0.84	0.84
GA	TFIDF	0.87	0.89	0.89	0.89	0.83	0.84	0.85	0.94	0.85	0.95	0.89	0.92	0.98	0.86	0.87	0.89
GA	CBOW	0.88	0.90	0.90	0.89	0.89	0.90	0.87	0.95	0.87	0.98	1.00	1.00	1.00	0.85	0.86	0.87
GA	SKG	0.90	0.90	0.91	0.91	0.91	0.90	0.87	0.95	0.87	0.97	0.95	1.00	1.00	0.84	0.86	0.91
GA	GLOVE	0.91	0.93	0.92	0.91	0.94	0.93	0.88	0.94	0.88	0.94	0.96	1.00	1.00	0.85	0.86	0.93
SVMSMOTE																	
AF	TFIDF	0.80	0.83	0.85	0.83	0.87	0.89	0.79	0.88	0.79	0.91	0.87	0.92	1.00	0.81	0.82	0.85
AF	CBOW	0.84	0.83	0.82	0.81	0.79	0.84	0.82	0.93	0.82	0.93	1.00	0.98	1.00	0.79	0.81	0.87
AF	SKG	0.84	0.87	0.88	0.84	0.84	0.83	0.84	0.90	0.84	0.90	0.97	1.00	1.00	0.80	0.83	0.90
AF	GLOVE	0.86	0.87	0.88	0.86	0.91	0.89	0.86	0.89	0.86	0.90	0.98	1.00	1.00	0.85	0.82	0.91
SIGF	TFIDF	0.79	0.81	0.67	0.20	0.84	0.64	0.84	0.90	0.84	0.94	0.83	0.89	0.97	0.78	0.82	0.86
SIGF	CBOW	0.82	0.81	0.81	0.78	0.81	0.83	0.82	0.94	0.82	0.94	1.00	1.00	1.00	0.80	0.81	0.87
SIGF	SKG	0.87	0.83	0.82	0.83	0.85	0.88	0.82	0.91	0.82	0.91	0.99	1.00	1.00	0.81	0.84	0.89
SIGF	GLOVE	0.89	0.86	0.84	0.83	0.91	0.82	0.79	0.92	0.79	0.90	0.96	1.00	1.00	0.82	0.86	0.90
INFOG	TFIDF	0.80	0.84	0.82	0.86	0.81	0.77	0.79	0.76	0.79	0.84	0.84	0.89	0.95	0.85	0.78	0.84
INFOG	CBOW	0.75	0.79	0.80	0.70	0.81	0.82	0.75	0.98	0.75	0.99	0.79	0.81	1.00	0.80	0.80	0.78
INFOG	SKG	0.86	0.86	0.89	0.90	0.91	0.92	0.83	0.98	0.83	0.99	0.85	0.90	1.00	0.83	0.83	0.83
INFOG	GLOVE	0.87	0.86	0.86	0.81	0.74	0.86	0.83	0.93	0.83	0.95	0.85	0.93	1.00	0.85	0.86	0.87
GA	TFIDF	0.86	0.87	0.85	0.86	0.82	0.88	0.76	0.92	0.76	0.94	0.86	0.91	1.00	0.82	0.84	0.86
GA	CBOW	0.81	0.83	0.82	0.81	0.85	0.82	0.80	0.92	0.80	0.97	0.99	0.96	1.00	0.78	0.80	0.84
GA	SKG	0.83	0.87	0.86	0.83	0.88	0.87	0.80	0.93	0.80	0.96	0.98	1.00	1.00	0.81	0.81	0.89
GA	GLOVE	0.86	0.82	0.86	0.85	0.84	0.93	0.82	0.91	0.82	0.93	0.93	1.00	1.00	0.78	0.84	0.90

(*continued on next page*)

Table 22 (*continued*).

ADSYN																	
AF	TFIDF	0.76	0.78	0.78	0.78	0.81	0.78	0.77	0.89	0.77	0.93	0.82	0.95	1.00	0.75	0.77	0.82
AF	CBOW	0.85	0.85	0.85	0.84	0.85	0.86	0.85	0.93	0.85	0.93	1.00	1.00	1.00	0.81	0.86	0.91
AF	SKG	0.90	0.90	0.88	0.86	0.92	0.93	0.83	0.95	0.83	0.90	0.98	1.00	1.00	0.81	0.87	0.93
AF	GLOVE	0.89	0.88	0.86	0.84	0.91	0.88	0.86	0.91	0.86	0.90	0.99	1.00	1.00	0.84	0.86	0.95
SIGF	TFIDF	0.71	0.76	0.72	0.68	0.62	0.70	0.75	0.86	0.75	0.94	0.74	0.83	0.94	0.72	0.74	0.73
SIGF	CBOW	0.84	0.86	0.87	0.86	0.88	0.87	0.82	0.94	0.82	0.92	1.00	0.92	1.00	0.79	0.87	0.90
SIGF	SKG	0.86	0.90	0.85	0.87	0.89	0.89	0.83	0.94	0.83	0.92	0.99	0.98	1.00	0.81	0.84	0.92
SIGF	GLOVE	0.86	0.87	0.86	0.85	0.90	0.93	0.83	0.93	0.83	0.92	0.98	1.00	1.00	0.82	0.84	0.93
INFOG	TFIDF	0.82	0.83	0.82	0.82	0.78	0.81	0.80	0.94	0.80	0.95	0.82	0.88	0.90	0.80	0.80	0.82
INFOG	CBOW	0.79	0.80	0.80	0.79	0.56	0.71	0.77	0.98	0.77	0.98	0.80	0.91	1.00	0.78	0.78	0.80
INFOG	SKG	0.76	0.77	0.76	0.66	0.80	0.79	0.76	0.97	0.76	0.98	0.80	0.84	0.99	0.77	0.77	0.79
INFOG	GLOVE	0.77	0.78	0.78	0.77	0.65	0.75	0.79	0.97	0.79	0.97	0.81	0.83	1.00	0.78	0.78	0.78
GA	TFIDF	0.77	0.79	0.80	0.77	0.82	0.49	0.76	0.91	0.76	0.92	0.81	0.84	0.99	0.76	0.79	0.81
GA	CBOW	0.83	0.84	0.87	0.86	0.89	0.87	0.81	0.93	0.81	0.93	1.00	1.00	1.00	0.80	0.86	0.86
GA	SKG	0.85	0.88	0.91	0.86	0.90	0.92	0.81	0.95	0.81	0.92	0.98	1.00	1.00	0.80	0.84	0.92
GA	GLOVE	0.86	0.87	0.86	0.88	0.93	0.94	0.83	0.91	0.83	0.90	0.98	1.00	1.00	0.82	0.84	0.92

- Among all the feature selection techniques the significant and correlation coefficient attains the best result with a median AUC value of 0.91

By accelerating e-business and catapulting it to the next stage of its growth cycle, the inclusion of Web-Services, a group of emerging technologies, has the potential to take service-oriented distributed computing to a completely new level in the future. Web services would make it easy for enterprises to link their information systems and commercial operations with their partners and customers, ushering in a new "service-oriented" distributed computing architecture as a result of its deployment. As a result, the cost of unsuccessful online transactions can have a significant influence on a company's capacity to meet market demand. A single hour of downtime may cost a retail firm thousands of dollars in lost sales. As a result, it is frequently necessary to maintain web service quality by tracking defects, code smells, and anti-patterns. It is estimated that human mistake accounts for 80% of software failures. Using word embedding techniques, this study offered methods for detecting five web service anti-patterns. This technique can be expanded in the future to assist forecast the remaining anti-patterns [98] in web services. We also intend to extend this work to discover cross-anti patterns using the federated learning technique [99].

CRediT authorship contribution statement

Lov Kumar: Conceptualization, Methodology, Investigation, Validation, Resource, Writing – original draft, Writing – review & editing, Supervision. **Sahithi Tummalapalli:** Conceptualization, Methodology, Investigation, Validation, Writing – original draft. **Sonika Chandrakant Rathi:** Methodology, Investigation, Validation, Writing – original draft, Writing – review & editing. **Lalita Bhanu Murthy:** Methodology, Investigation, Resource, Writing – review & editing, Supervision. **Aneesh Krishna:** Conceptualization, Methodology, Resource, Writing – review & editing, Supervision. **Sanjay Misra:** Conceptualization, Methodology, Validation, Resource, Writing – review & editing, Supervision.

Declaration of competing 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.

Data availability

https://github.com/ouniali/WSantipatterns.

Appendix

See Tables 19–22.

References

[1] A. Segev, E. Toch, Context-based matching and ranking of web services for composition, IEEE Trans. Serv. Comput. 2 (3) (2009) 210–222.

[2] A. Maiga, N. Ali, N. Bhattacharya, A. Sabané, Y.-G. Guéhéneuc, G. Antoniol, E. Aïmeur, Support vector machines for anti-pattern detection, in: 2012 Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering, IEEE, 2012, pp. 278–281.

[3] N. Moha, Y.-G. Guéhéneuc, L. Duchien, A.-F. Le Meur, Decor: A method for the specification and detection of code and design smells, IEEE Trans. Softw. Eng. 36 (1) (2009) 20–36.

[4] A. Maiga, N. Ali, N. Bhattacharya, A. Sabané, Y.-G. Guéhéneuc, E. Aimeur, Smurf: A svm-based incremental anti-pattern detection approach, in: 2012 19th Working Conference on Reverse Engineering, IEEE, 2012, pp. 466–475.

[5] F. Khomh, S. Vaucher, Y.-G. Guéhéneuc, H. Sahraoui, BDTEX: A GQM-based Bayesian approach for the detection of antipatterns, J. Syst. Softw. 84 (4) (2011) 559–572.

[6] G. Travassos, F. Shull, M. Fredericks, V.R. Basili, Detecting defects in object-oriented designs: using reading techniques to increase software quality, ACM Sigplan Notices 34 (10) (1999) 47–56.

[7] R. Marinescu, Detection strategies: Metrics-based rules for detecting design flaws, in: 20th IEEE International Conference on Software Maintenance, 2004. Proceedings, IEEE, 2004, pp. 350–359.

[8] M.J. Munro, Product metrics for automatic identification of" bad smell" design problems in java source-code, in: 11th IEEE International Software Metrics Symposium (METRICS'05), IEEE, 2005, p. 15.

[9] O. Ciupke, Automatic detection of design problems in object-oriented reengineering, in: Proceedings of Technology of Object-Oriented Languages and Systems-TOOLS 30 (Cat. No. PR00278), IEEE, 1999, pp. 18–32.

[10] F. Simon, F. Steinbruckner, C. Lewerentz, Metrics based refactoring, in: Proceedings Fifth European Conference on Software Maintenance and Reengineering, IEEE, 2001, pp. 30–38.

[11] A.A. Rao, K.N. Reddy, Detecting bad smells in object oriented design using design change propagation probability matrix 1, 2007.

[12] A. Ouni, M. Kessentini, K. Inoue, M.O. Cinnéide, Search-based web service antipatterns detection, IEEE Trans. Serv. Comput. 10 (4) (2017) 603–617.

[13] L. Kumar, A. Sureka, An empirical analysis on web service anti-pattern detection using a machine learning framework, in: 2018 IEEE 42nd Annual Computer Software and Applications Conference, COMPSAC, 1, IEEE, 2018, pp. 2–11.

[14] J.M. Rodriguez, M. Crasso, A. Zunino, An approach for web service discoverability anti-pattern detection for journal of web engineering, J. Web Eng. (2013) 131–158.

[15] S. Velioğlu, Y.E. Selçuk, An automated code smell and anti-pattern detection approach, in: 2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications, SERA, IEEE, 2017, pp. 271–275.

[16] G. Bavota, A. De Lucia, R. Oliveto, Identifying extract class refactoring opportunities using structural and semantic cohesion measures, J. Syst. Softw. 84 (3) (2011) 397–414.

[17] Y. Lee, Measuring the coupling and cohesion of an object-oriented program based on information flow, in: Proc. Int'L Conf. Software Quality, 1995, 1995.

[18] W. Li, S. Henry, Maintenance metrics for the object oriented paradigm, in: Software Metrics Symposium, 1993. Proceedings., First International, IEEE, 1993, pp. 52–60.

[19] G. Bavota, R. Oliveto, M. Gethers, D. Poshyvanyk, A. De Lucia, Methodbook: Recommending move method refactorings via relational topic models, IEEE Trans. Softw. Eng. 40 (7) (2013) 671–694.

[20] N. Tsantalis, A. Chatzigeorgiou, Identification of move method refactoring opportunities, IEEE Trans. Softw. Eng. 35 (3) (2009) 347–367.

[21] M. Fokaefs, N. Tsantalis, E. Stroulia, A. Chatzigeorgiou, Identification and application of extract class refactorings in object-oriented systems, J. Syst. Softw. 85 (10) (2012) 2241–2260.
[22] F. Khomh, S. Vaucher, Y.-G. Guéhéneuc, H. Sahraoui, A bayesian approach for the detection of code and design smells, in: 2009 Ninth International Conference on Quality Software, IEEE, 2009, pp. 305–314.
[23] R. Oliveto, F. Khomh, G. Antoniol, Y.-G. Guéhéneuc, Numerical signatures of antipatterns: An approach based on b-splines, in: 2010 14th European Conference on Software Maintenance and Reengineering, IEEE, 2010, pp. 248–251.
[24] E. Ligu, A. Chatzigeorgiou, T. Chaikalis, N. Ygeionomakis, Identification of refused bequest code smells, in: 2013 IEEE International Conference on Software Maintenance, IEEE, 2013, pp. 392–395.
[25] N. Tsantalis, T. Chaikalis, A. Chatzigeorgiou, Jdeodorant: Identification and removal of type-checking bad smells, in: 2008 12th European Conference on Software Maintenance and Reengineering, IEEE, 2008, pp. 329–331.
[26] G. Gui, P.D. Scott, Coupling and cohesion measures for evaluation of component reusability, in: Proceedings of the 2006 International Workshop on Mining Software Repositories, 2006, pp. 18–21.
[27] S.R. Chidamber, C.F. Kemerer, A metrics suite for object oriented design, IEEE Trans. Softw. Eng. 20 (6) (1994) 476–493.
[28] C. De Roover, T. D'Hondt, J. Brichau, C. Noguera, L. Duchien, Behavioral similarity matching using concrete source code templates in logic queries, in: Proceedings of the 2007 ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation, 2007, pp. 92–101.
[29] I.D. Baxter, C. Pidgeon, M. Mehlich, DMS/spl reg: program transformations for practical scalable software evolution, in: Proceedings. 26th International Conference on Software Engineering, IEEE, 2004, pp. 625–634.
[30] V. Wahler, D. Seipel, J. Wolff, G. Fischer, Clone detection in source code by frequent itemset techniques, in: Source Code Analysis and Manipulation, Fourth IEEE International Workshop on, IEEE, 2004, pp. 128–135.
[31] A. Marcus, D. Poshyvanyk, R. Ferenc, Using the conceptual cohesion of classes for fault prediction in object-oriented systems, IEEE Trans. Softw. Eng. 34 (2) (2008) 287–300.
[32] D. Poshyvanyk, A. Marcus, R. Ferenc, T. Gyimóthy, Using information retrieval based coupling measures for impact analysis, Empir. Softw. Eng. 14 (1) (2009) 5–32.
[33] N. Borovits, I. Kumara, P. Krishnan, S.D. Palma, D. Di Nucci, F. Palomba, D.A. Tamburri, W.-J. van den Heuvel, DeepIaC: deep learning-based linguistic anti-pattern detection in IaC, in: Proceedings of the 4th ACM SIGSOFT International Workshop on Machine-Learning Techniques for Software-Quality Evaluation, 2020, pp. 7–12.
[34] F.S. Alshraiedeh, N. Katuk, A URI parsing technique and algorithm for anti-pattern detection in RESTful Web services, Int. J. Web Inf. Syst. (2020).
[35] F. Sabir, G. Rasool, M. Yousaf, A lightweight approach for specification and detection of SOAP anti-patterns, Int. J. Adv. Comput. Sci. Appl. 8 (5) (2017) 455–467.
[36] H. Wang, A. Ouni, M. Kessentini, B. Maxim, W.I. Grosky, Identification of web service refactoring opportunities as a multi-objective problem, in: Web Services (ICWS), 2016 IEEE International Conference on, IEEE, 2016, pp. 586–593.
[37] A. Yugov, Approach to anti-pattern detection in service-oriented software systems, 28, (2) 2016.
[38] H. Wang, M. Kessentini, A. Ouni, Prediction of web services evolution, in: International Conference on Service-Oriented Computing, Springer, 2016, pp. 282–297.
[39] A. Ouni, M. Kessentini, K. Inoue, M.O. Cinnéide, Search-based web service antipatterns detection, IEEE Trans. Serv. Comput. 10 (4) (2017) 603–617.
[40] A. Ouni, R. Gaikovina Kula, M. Kessentini, K. Inoue, Web service antipatterns detection using genetic programming, in: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, 2015, pp. 1351–1358.
[41] J.L.O. Coscia, C. Mateos, M. Crasso, A. Zunino, Refactoring code-first web services for early avoiding wsdl anti-patterns: Approach and comprehensive assessment, Sci. Comput. Program. 89 (2014) 374–407.
[42] F. Palma, N. Moha, G. Tremblay, Y.-G. Guéhéneuc, Specification and detection of SOA antipatterns in web services, in: European Conference on Software Architecture, Springer, 2014, pp. 58–73.
[43] F. Palma, M. Nayrolles, N. Moha, Y.-G. Guéhéneuc, B. Baudry, J.-M. Jézéquel, Soa antipatterns: An approach for their specification and detection, Int. J. Coop. Inf. Syst. 22 (04) (2013) 1341004.
[44] M. Nayrolles, N. Moha, P. Valtchev, Improving SOA antipatterns detection in service based systems by mining execution traces, in: Reverse Engineering (WCRE), 2013 20th Working Conference on, IEEE, 2013, pp. 321–330.
[45] J.M. Rodriguez, M. Crasso, C. Mateos, A. Zunino, Best practices for describing, consuming, and discovering web services: a comprehensive toolset, Softw. - Pract. Exp. 43 (6) (2013) 613–639.
[46] C. Mateos, M. Crasso, A. Zunino, J.O. Coscia, Revising wsdl documents: Why and how-part ii, IEEE Internet Comput. 17 (5) (2013) 46–53.
[47] J.L.O. Coscia, C. Mateos, M. Crasso, A. Zunino, Avoiding wsdl bad practices in code-first web services, in: Proceedings of the 12th Argentine Symposium on Software Engineering (ASSE2011)-40th JAIIO, 2011, pp. 1–12.
[48] J.M. Rodriguez, M. Crasso, A. Zunino, M. Campo, Improving web service descriptions for effective service discovery, Sci. Comput. Program. 75 (11) (2010) 1001–1021.
[49] J. Ramos, et al., Using tf-idf to determine word relevance in document queries, in: Proceedings of the First Instructional Conference on Machine Learning, Vol. 242, (1) Citeseer, 2003, pp. 29–48.
[50] S. Qaiser, R. Ali, Text mining: use of TF-IDF to examine the relevance of words to documents, Int. J. Comput. Appl. 181 (1) (2018) 25–29.
[51] L. Ma, Y. Zhang, Using Word2Vec to process big text data, in: 2015 IEEE International Conference on Big Data (Big Data), IEEE, 2015, pp. 2895–2897.
[52] D. Guthrie, B. Allison, W. Liu, L. Guthrie, Y. Wilks, A closer look at skip-gram modelling, in: LREC, Vol. 6, 2006, pp. 1222–1225.
[53] C. McCormick, Word2vec tutorial-the skip-gram model, 2016, Apr-2016.[Online]. Available: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model.
[54] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1532–1543.
[55] N. Japkowicz, S. Stephen, The class imbalance problem: A systematic study, Intell. Data Anal. 6 (5) (2002) 429–449.
[56] A. Fernández, S. Garcia, F. Herrera, N.V. Chawla, SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary, J. Artificial Intelligence Res. 61 (2018) 863–905.
[57] X. Gu, P.P. Angelov, E.A. Soares, A self-adaptive synthetic over-sampling technique for imbalanced classification, Int. J. Intell. Syst. 35 (6) (2020) 923–943.
[58] R. Ghorbani, R. Ghousi, Comparing different resampling methods in predicting students' performance using machine learning techniques, IEEE Access 8 (2020) 67899–67911.
[59] Z. Hu, L. Wang, L. Qi, Y. Li, W. Yang, A novel wireless network intrusion detection method based on adaptive synthetic sampling and an improved convolutional neural network, IEEE Access 8 (2020) 195741–195751.
[60] E. Utami, I. Oyong, S. Raharjo, A.D. Hartanto, S. Adi, Supervised learning and resampling techniques on DISC personality classification using Twitter information in bahasa Indonesia, Appl. Comput. Inform. (2021).
[61] M.S. Amin, Y.K. Chiam, K.D. Varathan, Identification of significant features and data mining techniques in predicting heart disease, Telemat. Inform. 36 (2019) 82–93.
[62] Y.-Y. Pan, Y.-C. Chen, W.C.-W. Chang, M.-C. Ma, P.-C. Liao, Visualization of statistically processed LC-ms-based metabolomics data for identifying significant features in a multiple-group comparison, Chemometr. Intell. Lab. Syst. 210 (2021) 104271.
[63] E.O. Omuya, G.O. Okeyo, M.W. Kimwele, Feature selection for classification using principal component analysis and information gain, Expert Syst. Appl. 174 (2021) 114765.
[64] R. Lamba, T. Gulati, A. Jain, A hybrid feature selection approach for parkinson's detection based on mutual information gain and recursive feature elimination, Arab. J. Sci. Eng. (2022) 1–14.
[65] J. Chen, J. Hu, G. Zhang, Feature selection based on gain ratio in hybrid incomplete information systems, in: 2021 16th International Conference on Intelligent Systems and Knowledge Engineering, ISKE, IEEE, 2021, pp. 728–735.
[66] S.A. Gorji, A. Mostaan, H. Tran My, M. Ektesabi, Non-isolated buck–boost dc–dc converter with quadratic voltage gain ratio, IET Power Electron 12 (6) (2019) 1425–1433.
[67] O.N. Al Sayaydeha, M.F. Mohammad, Diagnosis of the Parkinson disease using enhanced fuzzy min-max neural network and OneR attribute evaluation method, in: 2019 International Conference on Advanced Science and Engineering, ICOASE, IEEE, 2019, pp. 64–69.
[68] S. Shaikh, L. Changan, M.R. Malik, Attribute rule performance in data mining for software deformity prophecy datasets models, in: 2019 International Conference on Advances in the Emerging Computing Technologies, AECT, IEEE, 2020, pp. 1–6.
[69] S. Asante-Okyere, C. Shen, Y.Y. Ziggah, M.M. Rulegeya, X. Zhu, Principal component analysis (PCA) based hybrid models for the accurate estimation of reservoir water saturation, Comput. Geosci. 145 (2020) 104555.
[70] N. Subba Rao, B. Sunitha, N. Adimalla, M. Chaudhary, Quality criteria for groundwater use from a rural part of Wanaparthy District, Telangana State, India, through ionic spatial distribution (ISD), entropy water quality index (EWQI) and principal component analysis (PCA), Environ. Geochem. Health 42 (2) (2020) 579–599.
[71] Q. Li, X. Han, Z. Liu, P. He, P. Shi, Q. Chen, F. Du, A novel information changing rate and conditional mutual information-based input feature selection method for artificial intelligence drought prediction models, Clim. Dynam. 58 (11) (2022) 3405–3425.
[72] D. Pérez-Guaita, G. Quintás, J. Kuligowski, Discriminant analysis and feature selection in mass spectrometry imaging using constrained repeated random sampling-cross validation (CORRS-CV), Anal. Chim. Acta 1097 (2020) 30–36.
[73] S. Mishra, P.K. Mallick, H.K. Tripathy, A.K. Bhoi, A. González-Briones, Performance evaluation of a proposed machine learning model for chronic disease datasets using an integrated attribute evaluator and an improved decision tree classifier, Appl. Sci. 10 (22) (2020) 8137.

[74] S. Singh, A.K. Singh, Web-spam features selection using CFS-PSO, Procedia Comput. Sci. 125 (2018) 568–575.
[75] F. Amini, G. Hu, A two-layer feature selection method using genetic algorithm and elastic net, Expert Syst. Appl. 166 (2021) 114072.
[76] S. Sayed, M. Nassef, A. Badr, I. Farag, A nested genetic algorithm for feature selection in high-dimensional cancer microarray datasets, Expert Syst. Appl. 121 (2019) 233–243.
[77] S. Chen, G.I. Webb, L. Liu, X. Ma, A novel selective naïve Bayes algorithm, Knowl.-Based Syst. 192 (2020) 105361.
[78] D.A. Putri, D.A. Kristiyanti, E. Indrayuni, A. Nurhadi, D.R. Hadinata, Comparison of naive bayes algorithm and support vector machine using pso feature selection for sentiment analysis on e-wallet review, in: Journal of Physics: Conference Series, Vol. 1641, (1) IOP Publishing, 2020, 012085.
[79] B. Charbuty, A. Abdulazeez, Classification based on decision tree algorithm for machine learning, J. Appl. Sci. Technol. Trends 2 (01) (2021) 20–28.
[80] C.S. Lee, P.Y.S. Cheang, M. Moslehpour, Predictive analytics in business analytics: decision tree, Adv. Decis. Sci. 26 (1) (2022) 1–29.
[81] Y. Pu, D.B. Apel, H. Xu, Rockburst prediction in kimberlite with unsupervised learning method and support vector classifier, Tunn. Undergr. Space Technol. 90 (2019) 12–18.
[82] M.M. Nishat, F. Faisal, T. Hasan, M.F.B. Karim, Z. Islam, M.R.K. Shagor, An investigative approach to employ support vector classifier as a potential detector of brain cancer from MRI dataset, in: 2021 International Conference on Electronics, Communications and Information Technology, ICECIT, IEEE, 2021, pp. 1–4.
[83] S. Singh, K.S. Parmar, S.J.S. Makkhan, J. Kaur, S. Peshoria, J. Kumar, Study of ARIMA and least square support vector machine (LS-SVM) models for the prediction of SARS-CoV-2 confirmed cases in the most affected countries, Chaos Solitons Fractals 139 (2020) 110086.
[84] L. Kumar, S.K. Sripada, A. Sureka, S.K. Rath, Effective fault prediction model developed using least square support vector machine (LSSVM), J. Syst. Softw. 137 (2018) 686–712.
[85] S. Tummalapalli, L. Kumar, L.B.M. Neti, A. Krishna, Detection of web service anti-patterns using weighted extreme learning machine, Comput. Stand. Interfaces 82 (2022) 103621.
[86] L. Kumar, C. Hota, A. Mahindru, L.B.M. Neti, Android malware prediction using extreme learning machine with different kernel functions, in: Proceedings of the Asian Internet Engineering Conference, 2019, pp. 33–40.
[87] W. Zong, G.-B. Huang, Y. Chen, Weighted extreme learning machine for imbalance learning, Neurocomputing 101 (2013) 229–242.
[88] O. Kisi, M. Alizamir, Modelling reference evapotranspiration using a new wavelet conjunction heuristic method: wavelet extreme learning machine vs wavelet neural networks, Agricult. Forest Meteorol. 263 (2018) 41–48.
[89] S. Shadkani, A. Abbaspour, S. Samadianfard, S. Hashemi, A. Mosavi, S.S. Band, Comparative study of multilayer perceptron-stochastic gradient descent and gradient boosted trees for predicting daily suspended sediment load: The case study of the Mississippi River, US, Int. J. Sediment Res. 36 (4) (2021) 512–523.
[90] S. Tummalapalli, L. Bhanu Murthy, A. Krishna, et al., Empirical Evaluation of the Performance of Data Sampling, Feature Selection Techniques, and Different Variants of Machine Learning for Web Service Anti-Pattern Prediction.
[91] T. Takase, S. Oyama, M. Kurihara, Effective neural network training with adaptive learning rate based on training loss, Neural Netw. 101 (2018) 68–78.
[92] M.R. Islam, A.R.M. Kamal, N. Sultana, R. Islam, M.A. Moni, et al., Detecting depression using k-nearest neighbors (knn) classification technique, in: 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), IEEE, 2018, pp. 1–4.
[93] S. Asadi, S.E. Roshan, A bi-objective optimization method to produce a near-optimal number of classifiers and increase diversity in Bagging, Knowl.-Based Syst. 213 (2021) 106656.
[94] Z. Khan, N. Gul, N. Faiz, A. Gul, W. Adler, B. Lausen, Optimal trees selection for classification via out-of-bag assessment and sub-bagging, IEEE Access 9 (2021) 28591–28607.
[95] S. Vijayvargiya, L. Kumar, A. Malapati, L.B. Murthy, S. Misra, Software functional requirements classification using ensemble learning, in: International Conference on Computational Science and Its Applications, Springer, 2022, pp. 678–691.
[96] M.P. McBee, O.A. Awan, A.T. Colucci, C.W. Ghobadi, N. Kadom, A.P. Kansagra, S. Tridandapani, W.F. Auffermann, Deep learning in radiology, Acad. Radiol. 25 (11) (2018) 1472–1480.
[97] S. Ali, A. Haider, M. Rahman, M. Sohail, Y.B. Zikria, Deep learning (DL) based joint resource allocation and RRH association in 5G-multi-tier networks, IEEE Access 9 (2021) 118357–118366.
[98] W.J. Brown, H.W. McCormick, S.W. Thomas, Anti-Patterns Project Management, John Wiley & Sons, Inc., 2000.
[99] R. Gupta, T. Alam, Survey on federated-learning approaches in distributed environment, Wirel. Pers. Commun. 125 (2) (2022) 1631–1652.