



KerMinSVM for imbalanced datasets with a case study on arabic comics classification



Ammar Nayal, Hadi Jomaa, Mariette Awad*

Department of Electrical and Computer Engineering, American University of Beirut, Lebanon

ARTICLE INFO

Keywords:

Imbalance datasets
Support vector machines
Arabic comics analysis
Natural language processing
Supervised classification

ABSTRACT

Many studies have been performed to classify large-sized text documents using different classifiers, ranging from simple distance classifiers such as K-Nearest-Neighbor (KNN) to more advanced classifiers such as Support Vector Machines. Traditional approaches fail when a short text is encountered due to sparsity resulting from a limited number of words. Another common problem in text classification is class imbalance (CI). CI occurs when one class of the data contains most of the samples while the other class contains only a few. Standard classifiers, when applied to imbalanced data, result in high accuracy for the majority class and low accuracy for the minority one. We were motivated to propose a novel framework for classifying the content of Arabic comics; therefore, we propose KerMinSVM, a kernel extension of our previously proposed MinSVM coupled with a new dimensionality featuring a reduction scheme based on word root frequency ratios (WRFR). KerMinSVM was tested on multiple imbalanced benchmark datasets, and the results were verified using three measures: accuracy, F-measure, and statistical analysis. WRFR was applied to the manual construction of the Arabic comic text dataset to detect strong content in children's comic books. Test results revealed that our proposed framework outperformed most of the methods for imbalanced datasets and short text classification.

1. Introduction

Due to the rapid development and spread of the internet, different types of short texts have been produced, such as web search snippets, chat messages, comments, status updates, tweets, news feeds, books, movie synopses and reviews. Classifying short text is of great importance for several purposes and applications, such as filtering offensive comments or assessing the satisfaction of customers with a certain product. Another example of short text is found in comic books. This text is usually unstructured and takes the form of brief conversations, consisting of multiple short sentences. Since short texts tend to have a sparse feature vector and exhibit class imbalance (CI), they cannot be classified with good accuracy using standard techniques.

An imbalanced dataset is one in which the different classification categories are not equally represented. A class that comprises many samples is referred to as the 'majority class' and conversely a class that contains very few samples is known as the 'minority class'. As stated before, when performing classification on an imbalanced dataset, the classifier tends to achieve a high level of accuracy for the majority class, but low accuracy for the minority ones. This is because most of the classification algorithms focus on maximizing overall accuracy, without taking into consideration the accuracy of each class. In imbalanced

datasets, the impact of minority samples is more pronounced than the majority samples. Misclassifying these minority samples will inevitably result in misleading and inaccurate information and hence undermine the aims of the application (Awad and Khanna, 2015).

Comics are usually popular amongst children. However, a number of these comics include strong content, such as conflicts, war, weaponry, and martyrdom, which are topics unsuitable for a younger audience. In general, the number of comics that contains strong material is very small with respect to the comics that are suitable for children. Nevertheless, detecting strong content in comic books is crucial. In this direction, we propose a new framework to classify short Arabic texts. Taking into consideration the root base nature of the Arabic language, we reduce the sparsity and dimensionality of the feature vector without adding external information to the original data. This methodology allows the roots of words to be used as features. It groups words of same root in one feature, and consequently reduces data dimensionality. To reduce the feature vector length even further, roots are grouped together based on their semantic similarities. To test the feature reduction technique, a dataset of Arabic comic text is manually constructed and annotated. Grouping similar roots together gave better representation of the constructed dataset and reduced the sparsity of the feature vector. To improve the classification accuracy of

* Corresponding author.

E-mail addresses: axn01@aub.edu.lb (A. Nayal), hsj04@aub.edu.lb (H. Jomaa), mariette.awad@aub.edu.lb (M. Awad).

Support Vector Machines (SVM) on imbalanced data, a kernel extension to the Minority Support Vector Machine (MinSVM) classifier we proposed earlier (Ajeeb et al., 2013) has been developed and tested.

The remainder of this paper includes, under Section II, the literature review for the techniques used to improve imbalanced data classification as well as previous work on Arabic text and on short text classifications. Section III presents the proposed framework containing KerMinSVM for imbalanced data classification and WRFR, the new feature extraction approach for short Arabic text. Experimental results are presented in Section IV, followed by concluding remarks in Section V.

2. Literature review

2.1. Data imbalance

Little SVM research has investigated improving classification for imbalanced datasets. One approach is to resample the dataset to achieve class balance. This is performed by either under-sampling the majority class or over-sampling the minority class. Another technique is to modify the SVM algorithm to overcome the data imbalance. Finally, hybrid methods are designed to benefit from the advantages of both mentioned approaches.

2.1.1. Data resampling

Chawla et al. (2011) proposed the Synthetic Minority Over-Sampling Technique (SMOTE) to oversample the minority class. The original minority data sample is used as a starting point to over-populate the minority class with artificial samples to balance the difference in samples between the classes. However, SMOTE requires the fine tuning of many user-defined parameters. Alternatively, Kubat et al. (1997) balanced the classes by randomly removing samples from the majority class. There are several heuristics for removing samples from the majority class. Randomly removing samples might result in a loss of important data from the majority class. Padmaja et al. (2007) combined both techniques, using SMOTE to oversample the minority class and the random under-sampling with elimination of outliers on the majority class. Chawla et al. (2003) later proposed SMOTEBoost, an enhanced SMOTE algorithm to improve performance. Inspired by SMOTEBoost, Seiffert et al. (2010) proposed RUSBoost, which combines random under-sampling with performance enhancements. These techniques use SMOTE or random under-sampling in every boosting iteration to attain the best new resampled dataset that has class balance.

To overcome the disadvantages of random resampling, Tang et al. (2009) proposed the GSVM RU algorithm for under-sampling, instead of randomly under-sampling the majority class. The algorithm assumes that only the support vectors (SVs) are important to the classification. It forms multiple majority information granules from which local majority SVs are extracted and aggregated, and then, it performs random undersampling over these points. Napierała et al. (2010) studied re-sampling methods for learning classifiers from imbalanced data and conducted experiments to investigate the effect of noisy and borderline examples from the minority class. They concluded that when the data suffers severely from those factors, then their proposed re-sampling method, selective preprocessing of imbalanced data (SPIDER2), and nearest cleaning rule (NCR) method would outperform the known oversampling methods. Otherwise, if the overlapping area is small or if most of the minority examples are not difficult to classify, then known oversampling methods perform well on improving prediction and are comparable to their proposed oversampling schemes. All of the former listed algorithms require optimization over some user-defined parameters.

Gazzah and Amara (2008) investigated four types of polynomial fittings to rebalance the dataset by oversampling the minority class: star, polynomial curved-bus, bus and mesh topologies. These functions

are utilized to create synthetic minority-class features before applying the learning algorithm. The results showed that applying mesh and star topologies outperforms regular classification.

2.1.2. SVM modification

Many studies proposed some improvements for SVM classifier over imbalanced datasets. One of the earliest modifications proposed by Veropoulos et al. (1999) used different loss functions (the square of the L2 norm instead of the L1 norm) for the majority and the minority classes to penalize the misclassification of the minority data samples.

Imam et al. (2006) tried to correct the skew of the learned classifier by introducing a factor, z , to the SVs of the minority class samples, to reduce the bias of the learned SVM model. Batuwita and Palade (2010) proposed the fuzzy support vector machine (FSVM) as a tool for CI learning. This is performed by choosing a membership function that achieves two goals. The first is to suppress the effect of CI, and the second is to reflect the importance of different training examples within the class to suppress the effect of outliers and noise. These techniques suffer either from the need to fine tune user-defined parameters, or from the high complexity of the algorithm. MinSVM was proposed by Ajeeb et al. (2013), where the objective of MinSVM is to make the learned classifier favor the minority class samples over the majority class samples. To achieve this, the resulting hyperplane should be as close as possible to the majority class. This will allow a greater margin for the minority class samples and thus they will be favored over the majority class ones.

2.1.3. Hybrid approaches

Hybrid approaches combine data resampling techniques with the modified SVM algorithm. Akbani et al., (2004) used an SVM that has different loss functions for the minority and the majority classes. Wang and Japkowicz (2010) used an ensemble of different SVM classifiers with different loss functions to improve the margin of error over a single classifier. Tax and Duin (2004) worked on forming a description of the training dataset so that new objects that resemble the training set are detected. They suggested a spherically shaped boundary around the target set characterized by a center and a radius whose values were determined through solving a constrained optimization problem seeking to minimize the volume of the sphere containing all the training objects.

2.1.4. Cost-sensitive learning

Cost-sensitive learning is another well-known method used on imbalanced datasets. As the name indicates, re-balancing the dataset is performed by adjusting the costs of the learning algorithms. Several approaches exist in cost-sensitive learning, which can be summarized into data weighting, augmenting cost-sensitive features and cost-minimizing techniques to combination schemes. In (Liu et al., 2010), a class confidence proportion decision tree was devised to create statistically significant rules regardless of the class size. Cieslak et al., (2012) proved that the use of Hellinger trees can compensate for CI. By penalizing the errors of different samples with different weights in a Regularized Least Squares method, (Vo and Won, 2007) showed an improvement in accuracy. Another approach introduced by Sun et al. (2007) added cost items within the Adaboost learning framework

2.2. Arabic text classification

There are many studies on the classification of Arabic texts. Such studies differ in the choice of the classifier as well as in the preprocessing of the text. Sawaf et al., (2001) skipped preprocessing by using a pure statistical approach that depends only on the N-grams of the words. Thabtah et al., (2008) applied a supervised approach using a maximum entropy classifier to classify documents into known categories, and an unsupervised learning approach to cluster unlabeled documents into groups. The feature vector consists of the raw words

along with their N-grams.

Khreisat (2006) used a simple K-Nearest-Neighbor (KNN) classifier which was used with three different distance measures (Cosine, Dice, and Jaccard). Khreisat (El Kourdi et al., 2004) suggested an Arabic text classification approach based on the N-gram method, and using distance and dice measures, the category of the classified text was determined. The preprocessing was performed through replacing the HAMZA letter with ALEF in the beginning of the words. The problem with such classifiers is their inefficiency and inability to scale to large datasets since they do not build a classification model. The computations hence have to be repeated for every new testing sample.

El Kourdi et al. (2004) suggest an automatic Arabic document categorization method using the Naïve Bayes algorithm. The data preprocessing includes parsing the text, removing the stop words and finding the roots of the words. Moh'd and Mesleh (2007) proposed using the χ^2 method for feature extraction and SVM for the classifier. In this work, data preprocessing was applied by removing digits and punctuation marks, normalizing some letters like (HAMZA to ALEF), filtering non-Arabic text, removing stop words, and finally removing rarely used terms. χ^2 statistics was used to select the feature. If the feature and class are independent, then χ^2 has a value of zero. An SVM classifier was adopted in this work due to the properties of the text classification problem. This problem has high dimensional space, few features are irrelevant, and the document space vector is sparse. Al-Harbi et al. (2008) also applied χ^2 statistics to feature extraction, and they used SVM and C5.0 as two different classifiers. Their studies showed that C5.0 outperformed SVM with only little accuracy improvement. Performing an advanced morphological analysis for the text is performed by El-Halees (2007) where a maximum entropy framework is proposed. Here, more morphological analysis is performed: during preprocessing, punctuations and non-letters are removed, some letters are normalized, and stop words and infrequent words are also removed. Finally, stemming and finding the root and part of speech (POS) of words was also performed.

2.3. Short text classification

Sahami and Heilman (2006) proposed a Kernel based method to compare the similarity of short text snippets. The approach treats each snippet as a query for a search engine and then by computing the TFIDF (Term Frequency–Inverse Document Frequency) vector for each of the retrieved documents. The kernel was calculated to retrieve the category of each document. Yih et al. Yih and Meek (2007) improved upon the work of (Sahami and Heilman, 2006) by using the relevance weighted inner-product of term occurrences instead of TFIDF. Bollegala et al. (2007) also used snippets returned from search engines and proposed a method to find the semantic similarity between words. First, the authors define a Web-based similarity score, then use automated lexico-syntactic pattern extraction, then rank these patterns based on their ability to express semantic similarity. An SVM classifier was trained to classify pairs of words that are synonyms. The output of the SVM was converted into posterior probability. Finally, the semantic similarity between two words was defined based on the posterior probability belonging to the synonym class. Zelikovitz and Hirsh (2000) assessed short text similarity using a combination of labeled training set and unlabeled background knowledge. The approach relies on WHIRL, which is a tool that can search and retrieve text information under specific conditions. Given a test text, WHIRL generates an intermediate table that contains a set of the ordered documents with the highest similarity with the test text, and then the similarity between documents is calculated using TFIDF. Phan et al., (2008) tried to classify short texts by relying on gaining external knowledge to expand the data and thus, building a more generalized classifier. A universal dataset was constructed from Wikipedia and analyzed using the hidden topic analysis model of Latent Dirichlet Allocation (LDA). After topic inference was applied for both the training and the testing data using

Gibbs sampling, the maximum entropy classifier for the classification phase was adopted. Chen et al. (2011) tried to improve the work of (Phan et al., 2008) by using a Multi-Granularity Topics space approach on external data, and they also used an SVM classifier, which performed better than Maximum Entropy. Hu et al. (2009) clustered short text using internal and external semantics. The work was divided into three stages: 1) the Hierarchical Feature Extraction of internal features stage, 2) the External Feature Generation stage and 3) the Feature Selection. In practice, short text classification is useful in web page classification, as suggested by Charalampopoulos and Anagnostopoulos (2011), which used Weka tool (Holmes et al., 1994) for clustering and classification of synthesized data representing web documents. Some of the other studies attempted to use different feature extraction methods instead of trying to expand the feature space with new data. Faguo et al. (2010) proposed an algorithm based on statistics to classify short texts. In this work, they used heuristic weighting for each term in the documents based on the number of documents that do and do not have this term. The features are ordered by their weight score, and the top M words are selected as features. Finally, (Rizk and Awad, 2012) aims to decipher whether a given short sports article was written from an objective or subjective perspective. A genetic algorithm with syntactic features on a homemade corpus of three hundred sports articles demonstrate that special preference to the last sentence in an article provides higher classification accuracy.

3. Proposed framework

The proposed framework consists of a text processing stage and a classifier. Fig. 2 shows the workflow of the methodology. First, it is necessary to extract the text from the comic books. The data used in this study was in PDF format, and because there were no available tools to extract the texts automatically, this step was performed manually. The extracted text was saved in UTF-8 text files. The next step was the text processing, where the raw text files are converted to feature vector representations. To efficiently train and test the methodology on the acquired data, the data were divided into five folds to be used as training and testing pairs. Finally, the KerMinSVM classifier was trained on one training fold to build a model, and this model was used to classify the classes of the testing fold.

3.1. KerMinSVM for non-separable dataset

In this work, we extend the MinSVM formulation that was presented in (Ajeeb et al., 2013) to handle linearly non-separable data.

In addition to the integration of the kernel into MinSVM, we introduce a new term τ_i , which accounts for unequal margins for the majority and minority classes. It minimizes the distance between the majority data samples and the separating hyperplane and maximizes

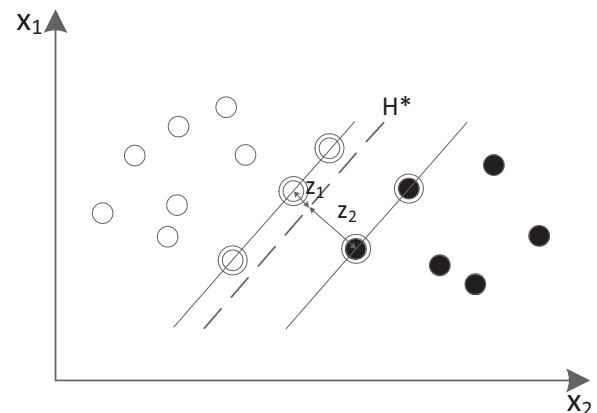


Fig. 1. KerMinSVM hyperplane and margin.

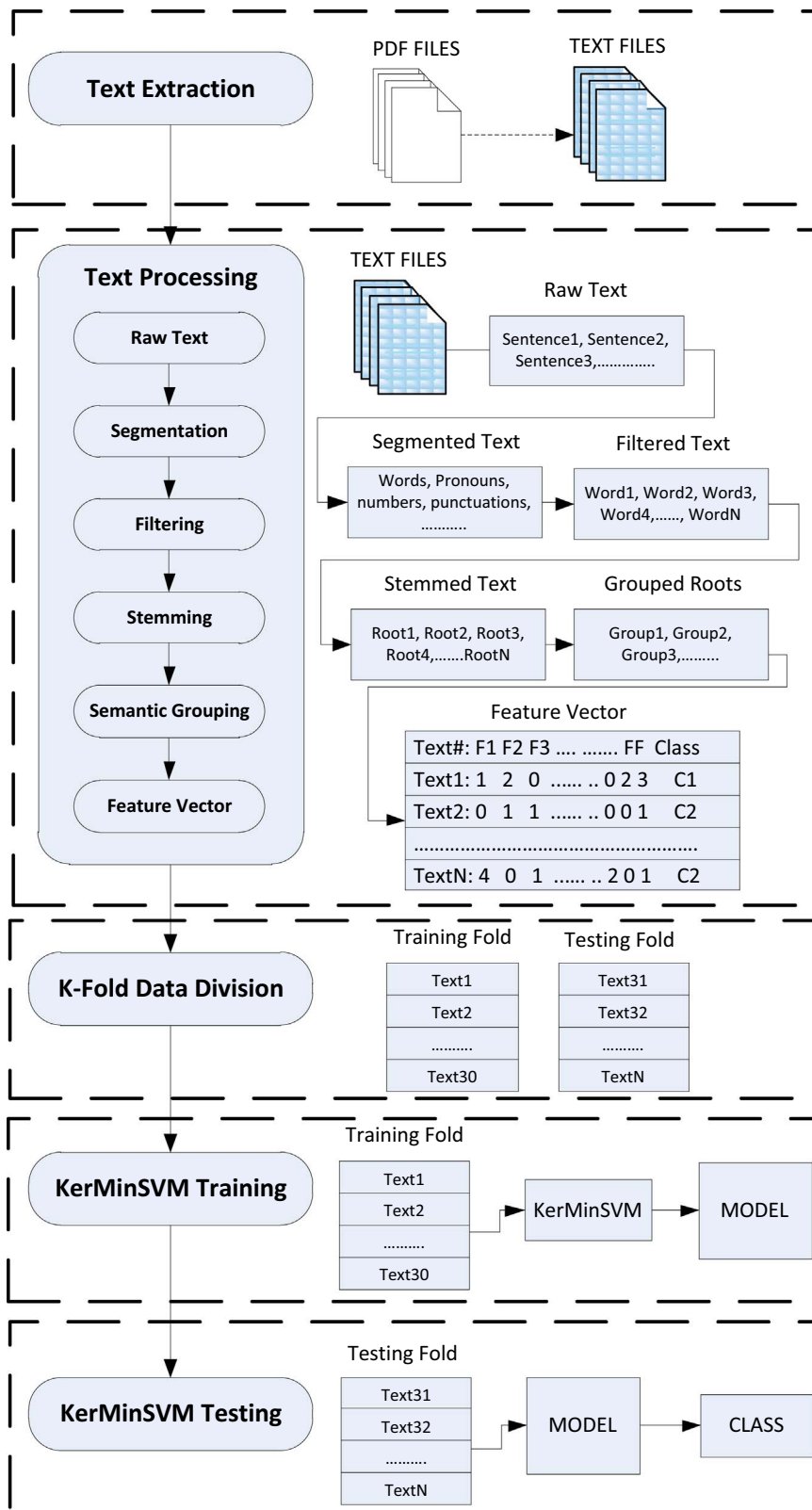


Fig. 2. Workflow for the proposed framework.

the distance between the minority data samples and the separating hyperplane as show in Fig. 1. The KerMinSVM formulation becomes:

$$\min_w \frac{1}{2} \|w\|^2 + C^+ \sum_{\{i|y_i=+1\}} \xi_i^+ + C^- \sum_{\{i|y_i=-1\}} \xi_i^- + D^+ \sum_{\{i|y_i=+1\}} \tau_i^+ - D^- \sum_{\{i|y_i=-1\}} \tau_i^-$$

Subject to:

$$w^T \phi(x_i) + b \geq \tau_i^+ - \xi_i^+ \text{ for } x_i : y_i = +1$$

$$w^T \phi(x_i) + b \leq -\tau_i^- - \xi_i^- \text{ for } x_i : y_i = -1$$

$$\tau_i^+, \xi_i^+, \tau_i^-, \xi_i^- \geq 0 \text{ for } \forall x_i$$

where the subscript “+” represents the majority class and the “-” represents the minority class, and C^+, C^-, D^+, D^- are tuning parameters.

The Lagrangian of the problem becomes:

$$\begin{aligned} \mathcal{L}_p = & \frac{1}{2} \|w\|^2 + C^+ \sum_{\{i|y_i=+1\}}^{N_1} \xi_i^+ + C^- \sum_{\{i|y_i=-1\}}^{N_2} \xi_i^- \\ & + D^+ \sum_{\{i|y_i=+1\}}^{N_1} \tau_i^+ - D^- \sum_{\{i|y_i=-1\}}^{N_2} \tau_i^- \\ & - \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i [w^T \phi(x_i) + b - \tau_i^+ + \xi_i^+] \\ & + \sum_{\{i|y_i=-1\}}^{N_2} \mu_i [w^T \phi(x_i) + b + \tau_i^- + \xi_i^-] \\ & - \sum_{\{i|y_i=+1\}}^{N_1} \alpha_i \xi_i^+ - \sum_{\{i|y_i=-1\}}^{N_2} \beta_i \xi_i^- \\ & - \sum_{\{i|y_i=+1\}}^{N_1} \gamma_i \tau_i^+ - \sum_{\{i|y_i=-1\}}^{N_2} \delta_i \tau_i^- \end{aligned}$$

where: $\lambda_i, \mu_i, \alpha_i, \beta_i, \gamma_i, \delta_i$ are the Lagrange multipliers. By finding the KKT conditions and substituting in the Lagrangian, we get the dual problem.

$$\begin{aligned} \max_{\lambda, \mu} & \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=-1\}}^{N_2} \lambda_i \mu_j \phi(x_i) \phi(x_j) \\ & - \frac{1}{2} \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=+1\}}^{N_1} \lambda_i \lambda_j \phi(x_i) \phi(x_j) \\ & - \frac{1}{2} \sum_{\{i|y_i=-1\}}^{N_2} \sum_{\{j|y_j=-1\}}^{N_2} \mu_i \mu_j \phi(x_i) \phi(x_j) \end{aligned}$$

Subject to:

$$\sum_{\{i|y_i=+1\}}^{N_1} \lambda_i - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i = 0$$

$$0 \leq \lambda_i \leq C^+$$

$$\mu_i \geq D^-$$

Using $K(x_i, x_j) = \phi(x_i) \phi(x_j)$ KerMinSVM can be represented as:

$$\begin{aligned} \max_{\lambda, \mu} & \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=-1\}}^{N_2} \lambda_i \mu_j K(x_i, x_j) \\ & - \frac{1}{2} \sum_{\{i|y_i=+1\}}^{N_1} \sum_{\{j|y_j=+1\}}^{N_1} \lambda_i \lambda_j K(x_i, x_j) \\ & - \frac{1}{2} \sum_{\{i|y_i=-1\}}^{N_2} \sum_{\{j|y_j=-1\}}^{N_2} \mu_i \mu_j K(x_i, x_j) \end{aligned}$$

Subject to:

$$\sum_{\{i|y_i=+1\}}^{N_1} \lambda_i - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i = 0$$

$$0 \leq \lambda_i \leq C^+$$

$$\mu_i \geq D^-$$

After solving this problem for λ_i, μ_i we can find the separating hyperplane where:

$$w = \sum_{\{i|y_i=+1\}}^{N_1} \lambda_i \phi(x_i) - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i \phi(x_i)$$

And the classifier's formula becomes:

$$f(x) = \text{sign} \left(\sum_{\{i|y_i=+1\}}^{N_1} \lambda_i K(x, x_i) - \sum_{\{i|y_i=-1\}}^{N_2} \mu_i K(x, x_i) + b \right)$$

3.2. Short arabic text methodology

Classifying short text by applying traditional techniques yields a sparse feature vector and thus results in poor performance for the classifiers. In this study we propose WRFR, a word root based feature reduction approach to reduce the sparsity of the feature vector by applying multiple preprocessing steps on the text before converting it into a feature vector. The proposed approach is designed for the constructed dataset specifically, and it is not extensively tested on other Arabic datasets. This approach can be described in five stages, as shown in Fig. 3.

1) Segmentation

The raw text is segmented using the *Stanford Word Segmenter* (Monroe et al., 2014), which separates the connected prepositions and pronouns from the original word, converts the HAMZA to ALEF in words that starts with HAMZA, and separates any punctuation.

2) Filtering Stage

Filtering the text consists of removing stop words, connected and separated pronouns, non-Arabic words, numerals, and punctuation. These parts of the text simply increase the size of the feature vector without helping to distinguish the texts.

3) Stemming Stage

Arabic language is a root based language, meaning that almost every word is either a root of itself or is derived from a three-letter or a four-letter root. Words that are derived from the same root have similar meanings, and thus can be grouped by their root. Because of this, they can be considered to be one feature, thus reducing the length of the feature vector. At this stage, stemming is applied on the words from the output of the previous stage using the Khoja Stemmer (Khoja and Garside, 1999).

4) Semantic Grouping Stage

This methodology takes the idea of grouping similar words one step further. Stemming helps to group words with the same root, but some words with similar meanings do not share the same root. We were not aware of an available offline dataset that contains groups of similar Arabic roots, so a semantic method was used to group the roots with similar meanings in the following manner:

- Each root from the dataset is used as a query word for: <http://dictionary.sensagent.com/> "root"/ar-ar/ (sensagent.com), which returns a webpage containing the synonyms of that root, if available.
- The synonyms are extracted from the webpage source and stored in a table containing each root with its synonyms.
- The roots in the table are compared together. If a root shares a synonym with another root, the roots are considered to have a similar meaning and grouped together. If one root shares synonyms with a root that is already in a group, the new root is added to the existing group. This process is applied for only one iteration. This means that the resulting groups are not aggregated together again.

Fig. 4 illustrates the semantic grouping stage. By the end of this stage, roots that share a similar meaning are grouped together and can be considered to be one feature.

The same offline dataset used in (Jomaa et al., 2015, 2016) was adopted in this study. The dataset contains all 1) the roots of the Arabic

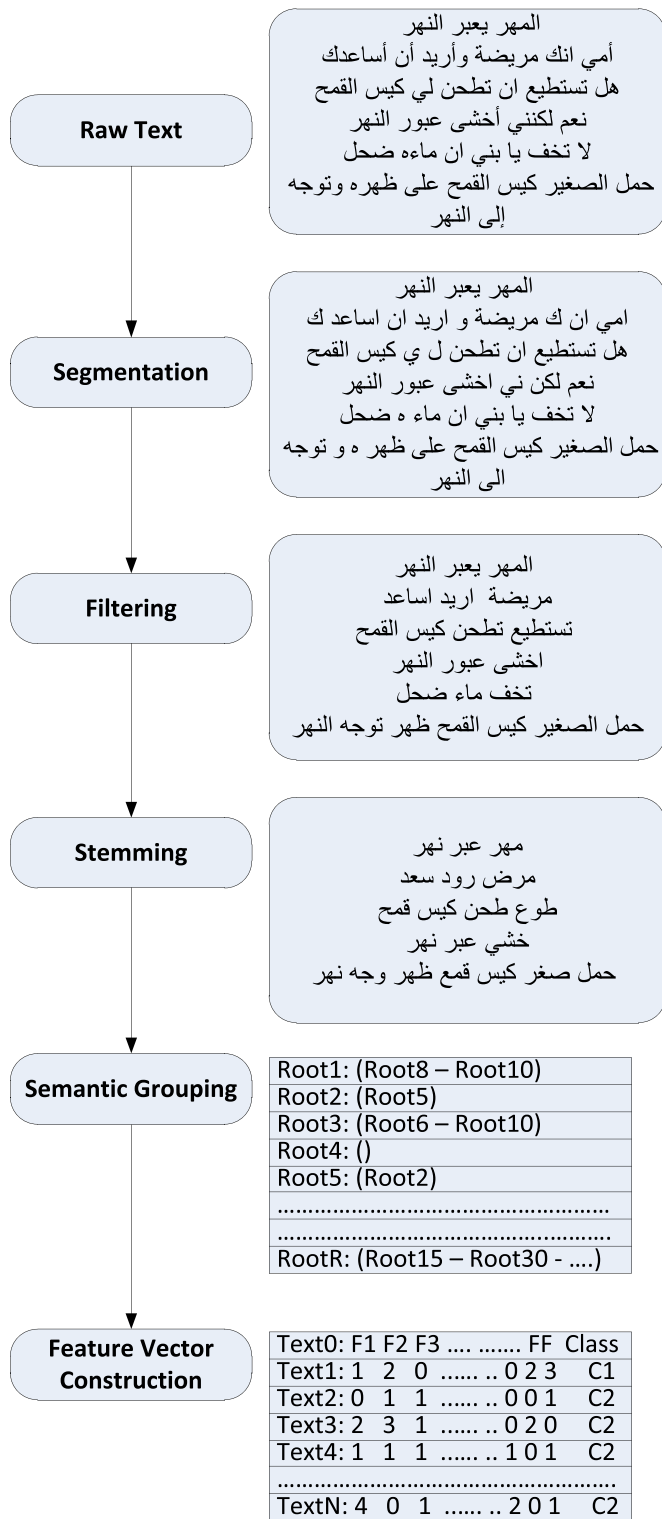


Fig. 3. Text processing workflow.

language obtained from “Mukhtar Al-Sihah” Wikipedia page¹ 2) a table containing all the roots with their synonyms, if available, and 3) a table containing each root, the other roots that share a similar meaning and their corresponding English translations from Google Translate, as shown in Table 1.

¹ Arabic Website: (https://ar.wikisource.org/wiki/%D9%85%D8%AE%D8%AA%D8%A7%D8%B1_%D8%A7%D9%84%D8%B5%D8%AD%D8%A7%D8%AD/%D9%81%D9%87%D8%B1%D8%B3)

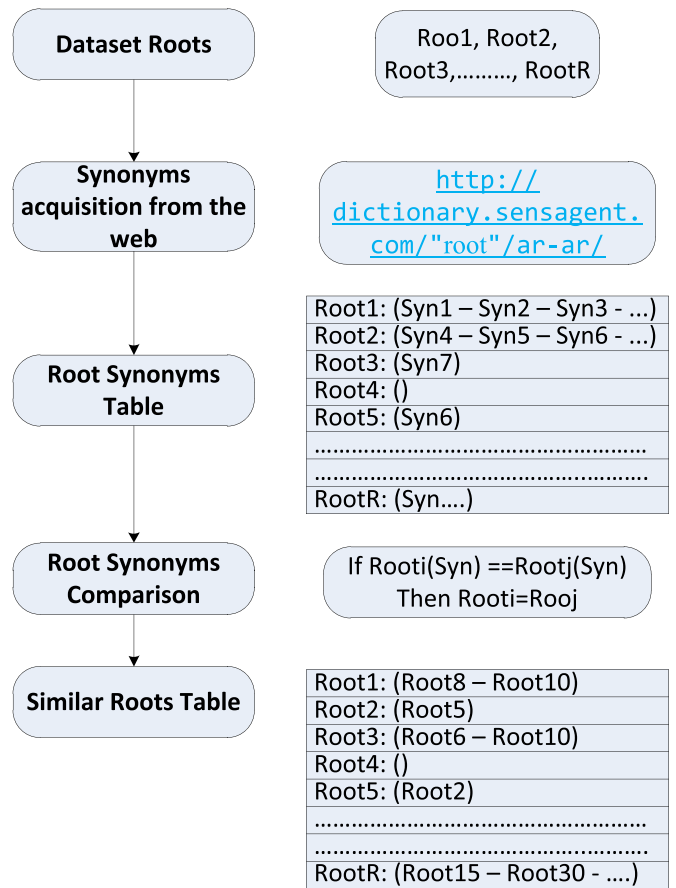


Fig. 4. Semantic Grouping Illustration.

Table 1
Synonyms of Arabic words and English translation.

ثمن	فُدِّر - أَجْرَة - قَيْم - بَيْعَر - حَمِيَن - تَقْرِيَبِي - يَقْدَر - ثَمَن - فَيْمَه - رِسوم - كَنْز - فَيْمَة - كَلْفَة - نَحْمِيَن - ثَمَن - تَكْلَفَة - قُدِّر - ثَمَن - فَيْمَة - نَسْبَة - حَسَب
Price	amount – fare – valuable – price – guess – approximate – estimated – price – value – fees – treasure - value – cost – guess – price – cost – amount – price – values – rate – number
فرز	فَضَلَ - تَضْيِيف - تَبْوِيِب - صَب - فَرَع - تَرْتِيِب - فَرَع - أَطْلَق
sort	separation – classification – tabulation – casting – empty – arrangement – release
فرد	نَفْس - شَخْص - ذَكَر - مَثَل - أَنَس - صَنَف - شَبَه - رُوح - دُور - رَجَل - بَشَر - أَحَد - أَنَا - نَرَب - وَحْش
individual	same- person – male – like – anas – class – semi – soul – role – man – humans – one – me – soils – monster
فُسد	مَضَر - حَمْض
mess	corruption - sabotage – mangle – harmful - acid

5) Feature Vector Building Stage

At this stage, all the data are processed and rooted and the roots are semantically grouped, so the feature vector is built for each document in the dataset and the whole dataset can be presented in a tabular format.

At this level, it should be noted that using this approach for grouping has a shortcoming. The efficiency of the grouping process depends on how relevant the retrieved synonyms are. Since the process is completely automated without humane observation, this might cause some roots to be grouped together because of one common synonym,

Table 2
Datasets used for testing.

Data	# of points	IR	# of Features
Paw	600	5	2
Subclass	600	5	2
Clover	600	5	2
Ecoli	336	8.6	7
Cleveland	177	12.62	13
Abalone	731	16.4	8
Zoo	101	19.2	16
Poker	244	23.5	10
Derma	171	1.85	34
Hepatitis	76	1.3	19
Breast cancer	748	3.2	4
Haber	306	2.78	3
Led	443	10.97	7
Wisconsin	683	1.86	9
Thyroid	215	5.14	5
Vehicle	846	2.9	18

even if they are not semantically similar. For example, the root (ثمن - Price) has the synonym (رسوم - fees), which also means (paintings), which will cause the roots (Price - ثمن, Paint - رسم) to be grouped together.

4. Experimental results

4.1. KerMinSVM benchmark testing

This section evaluates the performance of KerMinSVM and compares it with the performances of the original SVM implementation, an SVM with different cost functions (CSVM), an SVM after applying SMOTE (SMOTE-SVM) and finally with an SVM after applying RUS on the data (RUS-SVM). Non-SVM approaches such as KNN and Tree-Fitting were also evaluated to highlight a distinction between different types of classification methods. For these tests, we chose 16 datasets with different Imbalance Ratios (IR), ranging between (1.3–23.5), as listed in Table 2. These datasets can be found in the Keel data repository (Alcalá et al., 2010).

For testing, a five-fold cross-validation was performed on each dataset. To measure the performance of the classifiers, two metrics were used: the normal accuracy measure to evaluate the accuracy of the classifier for each class, and the F-measure to compare the overall performance of the classifier. In other words, the F-measure was used to evaluate the trade-off between improving the accuracy of the minority class and accuracy loss of the majority one. Here the classifier with the highest F-measure was considered to be most accurate.

In this work, the minority class samples are considered the positive samples and the majority class samples are considered the negative ones.

- TP: Positive samples that are correctly classified.
- FP: Negative samples that are incorrectly classified.
- TN: Negative samples that are correctly classified.

Table 3
Average results for ALL datasets for all classifiers.

Data	Averaged					
	Sensitivity	Specificity	Accuracy	Precision	Recall	F-measure
KerMinSVM	0.69	0.92	0.85	0.70	0.69	0.64
SVM	0.68	0.83	0.89	0.67	0.68	0.66
RUS-SVM	0.65	0.85	0.84	0.61	0.65	0.61
SMOTE-SVM	0.70	0.89	0.88	0.64	0.70	0.65
CSVM	0.68	0.89	0.88	0.62	0.68	0.62
DT	0.65	0.87	0.88	0.64	0.65	0.61

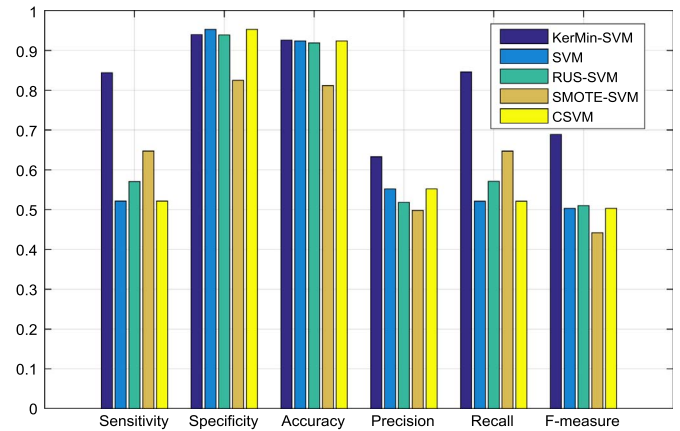


Fig. 5. Averaged Results for some classifier over some datasets.

FN: Positive samples that are incorrectly classified.

$$\text{Minority class accuracy or Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Majority class accuracy or Specificity} = \frac{TN}{TN + FP}$$

$$\text{Overall Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F - Measure} = 2 * \left(\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \right)$$

The simulations were executed using MATLAB R2013b and CVX 2.1 toolbox on a machine that has 2 Intel(R) Xeon(R) 2620@2.00 GHz CPUs and 24 GB of RAM running windows 7 64 bit.

Table 3 contains the averaged values over all the datasets.

In Table 2 and Fig. 5, we noticed that the KerMinSVM classifier outperformed all other techniques that are used to enhance the performance of the SVM classifier. In Figs. 6 and 7 we can see that KerMinSVM improved the sensitivity (7–430%) and the F-measure (6–235%) without scarifying too much of the specificity, whereas other techniques do not guarantee the improvement of the SVM performance. We can see that data-resampling techniques (SMOTE - RUS) do not always improve the performance of the SVM classifier. Here, we can see that in some tests that these techniques degrade the SVM performance by changing the distribution of the data and possibly leading to more outliers. The SVM with different cost functions does improve the sensitivity of the classifier, but this improvement comes at the cost of decreasing the specificity, which leads to a lower F-measure and lower overall accuracy. Therefore, the best results for CSVM are

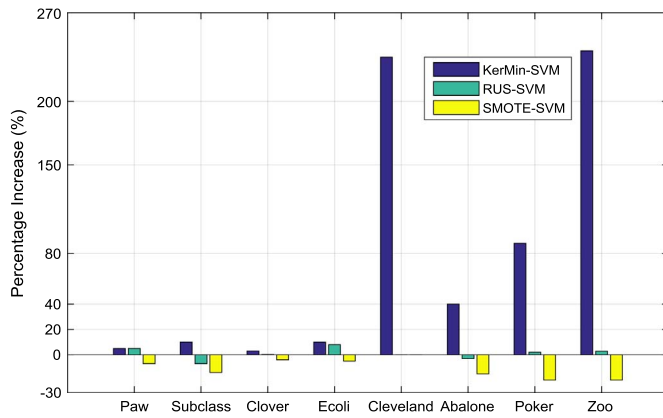


Fig. 6. F-measure improvement of some of the classifiers over the original SVM.

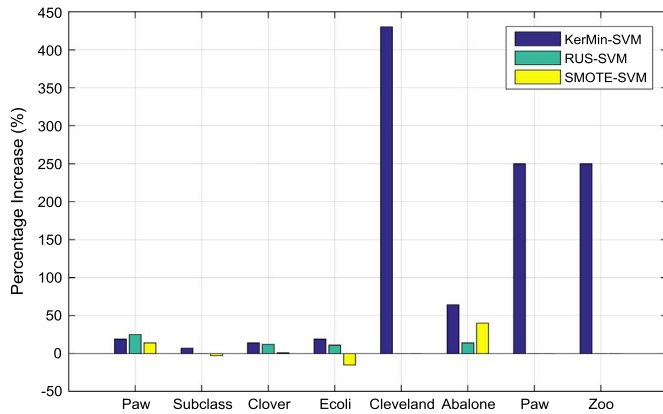


Fig. 7. Sensitivity improvement of some of the classifiers over the original SVM.

when we have the same cost function for both the minority and majority data samples.

On the other hand, SVM modification did not always achieve highest sensitivity amongst other classification techniques, namely three nearest neighbor. This could be because the major class has many more entries than the minor class. Moreover, this affects the specificity, where the true negative classification rate is the lowest on average.

4.2. Statistical analysis

Statistical analysis is used to test the significance of the difference in the accuracy between classifiers. Given two classifiers, the statistical test compares whether the classifiers have the same expected error rate.

The K-fold cross-validated paired *t*-test uses K-fold cross-validation to get K training/testing set pairs. The classifiers are trained on the training set $train_i$ and tested on the testing set $test_i$. The error rates of the classifiers are p_i^1, p_i^2 where: $i = 1, 2, \dots, K$.

If the classifiers have the same error rate they should have the same mean, i.e., the difference in their mean should be equal to 0. The difference in error rates on fold i is $p_i = p_i^1 - p_i^2$ for K cross-validation tests, and we get a distribution of p_i containing K points. Assuming both p_i^1 and p_i^2 are normally distributed, then their difference p_i is also normally distributed.

The null hypothesis H_0 is that this distribution has a normal zero mean.

$$H_0 : \mu = 0 \text{ vs. } H_1 : \mu \neq 0$$

$$\text{Let: } \bar{p} = \frac{\sum_{i=1}^K p_i}{K}, S^2 = \frac{\sum_{i=1}^K (p_i - \bar{p})^2}{K-1}$$

Under the null hypothesis where $\mu = 0$, we have a statistic that is t-

Table 4
Statistical analysis score.

Ecoli	Majority class score	Minority class score	Overall score
KerMinSVM vs. SVM	0.408	-3.162	-1.372
	Accepted	rejected	accepted
KerMinSVM vs. SMOTE-SVM	1.87	-4.81	-0.492
	accepted	rejected	accepted
KerMinSVM vs. RUS-SVM	0.34	-2.436	-1.372
	accepted	rejected	accepted
Cleveland			
Classifiers Pairs	Majority class score	Minority class score	Overall score
KerMinSVM vs. SVM	3.316	-6.32	0.25
	rejected	rejected	accepted
KerMinSVM vs. SMOTE-SVM	2.82	-5.79	0.166
	rejected	rejected	accepted
KerMinSVM vs. RUS-SVM	3.316	-6.32	0.25
	rejected	rejected	accepted
Abalone			
Classifiers Pairs	Majority class score	Minority class score	Overall Score
KerMinSVM vs. SVM	0	-4.22	-1.168
	accepted	rejected	accepted
KerMinSVM vs. SMOTE-SVM	0.269	-3.764	-2.358
	accepted	rejected	accepted
MinSVM vs. RUS-SVM	-0.971	-2.236	-1.544
	accepted	rejected	accepted
Zoo			
Classifiers Pairs	Majority class score	Minority class score	Overall Score
KerMinSVM vs. SVM	2.236	-2.633	1.176
	rejected	rejected	accepted
KerMinSVM vs. SMOTE-SVM	2.236	-2.449	0.667
	rejected	rejected	accepted
KerMinSVM vs. RUS-SVM	2.13	-2.449	0.4082
	accepted	rejected	accepted
Poker			
Classifiers Pairs	Majority class score	Minority class score	Overall Score
KerMinSVM vs. SVM	2.18	-2.449	1.469
	Rejected	rejected	accepted
KerMinSVM vs. SMOTE-SVM	2.018	-2.449	-1.49
	Accepted	rejected	accepted
KerMinSVM vs. RUS-SVM	1.772	-2.449	1.088
	Accepted	rejected	accepted

distributed with $K - 1$ degrees of freedom

$$\frac{\sqrt{K} \cdot \bar{p}}{S} \sim t_{K-1}$$

The test rejects the hypothesis at a significant level α if this value is outside the interval $(-t_{\alpha/2, K-1}, t_{\alpha/2, K-1})$ for $\alpha = 0.1$ the confidence level is at 90% and the interval is $(-2.132, 2.132)$.

The analysis was applied to test the performance on five datasets. The test was performed on the error rates for the majority and minority class and the overall error rate. This will show the effect of the KerMinSVM classifier on the majority class and the overall accuracy.

As shown in Table 4, all the tests reject the hypothesis for the minority class and indicate a significant difference in the error rates in favor of the KerMinSVM. For the majority class, the hypothesis is rejected on two datasets and accepted on three, meaning that there were differences in the error rates on two of the datasets in favor of the other classifiers and no difference on the other three datasets. As for the overall error rate, the hypothesis was accepted for all data, meaning there was no difference for the overall error rate. In conclusion, the KerMinSVM classifier has better accuracy on the minority class without sacrificing the overall accuracy, even when it had less accuracy on the majority class.

To evaluate the complexity of the classifiers, the runtime for each classifier on each dataset is measured and then averaged over all the

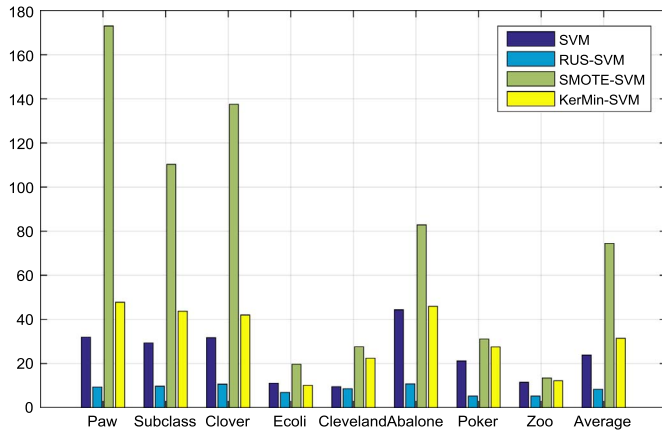


Fig. 8. Time consumption for each classifier.

Table 5
Mean and Standard Deviation of the Comic Data.

	Min	Max	Mean	Standard deviation
Characters	91	2030	834	484
Words	25	460	195	113

Table 6
Number of samples in each testing case.

	# of Majority		# of Minority		# of Minority
	Samples		Samples 1		
	113		10		50
	# of Training samples		# of Testing samples		
Case	Major	Minor	Major	Minor	Class Ratio
Case 1	90	12	23	3	7.6
Case 2	98	4	25	1	25
Case 3	8	4	2	1	2

data. Fig. 8 shows that the SMOTE-SVM has the highest runtime because it needs to perform oversampling on the data, which is time consuming and leads to a larger number of data samples, and thus longer processing time. The RUS has the lowest processing time because it randomly removes majority data samples, which leads to a

Table 7
Test results.

Data	Standard vs. Islamic with positive and violent themes					
Classifier	Sensitivity	Specificity	Accuracy	Recall	Precision	F-measure
KerMinSVM	0.73	0.96	0.93	0.72	0.73	0.71
SVM	0.80	0.51	0.55	0.34	0.8	0.38
RUS-SVM	0.60	0.91	0.87	0.78	0.6	0.54
SMOTE-SVM	0.80	0.68	0.69	0.26	0.8	0.38
Data	Standard and positive Islamic vs. Islamic containing violent content					
Classifier	Sensitivity	Specificity	Accuracy	Recall	Precision	F-measure
KerMinSVM	0.80	0.99	0.98	0.70	0.80	0.73
SVM	0.40	0.99	0.97	0.40	0.40	0.33
RUS-SVM	0.40	0.99	0.95	0.80	0.40	0.53
SMOTE-SVM	0.60	0.98	0.96	0.60	0.45	0.48
Data	Positive Islamic comics vs. Islamic comics with violent content					
Classifier	Sensitivity	Specificity	Accuracy	Recall	Precision	F-measure
KerMinSVM	0.80	1	0.93	0.8	0.80	0.80
SVM	0.40	0.8	0.66	0.26	0.40	0.30
RUS-SVM	0.80	0.9	0.85	0.7	0.80	0.75
SMOTE-SVM	0.20	0.8	0.60	0.0666	0.20	0.10

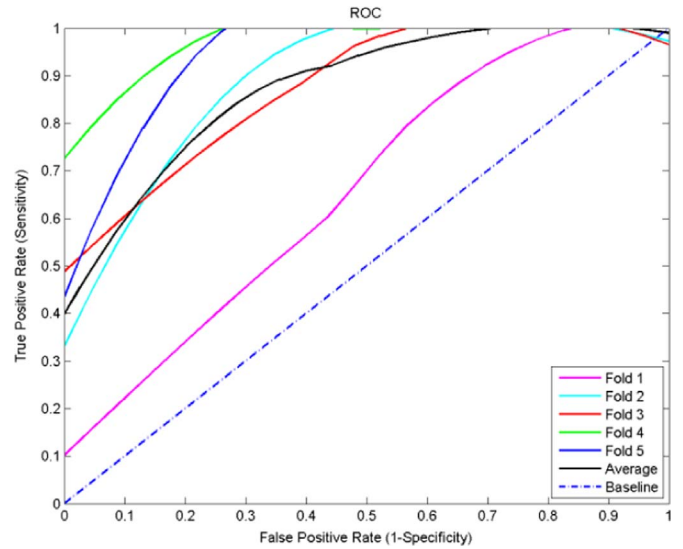


Fig. 9. ROC Curves for KerMinSVM for the first testing case.

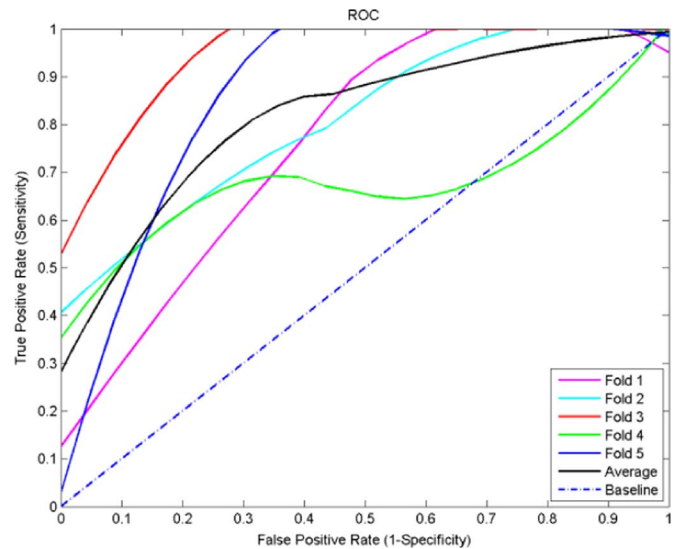


Fig. 10. ROC Curves for SVM for the first testing case.

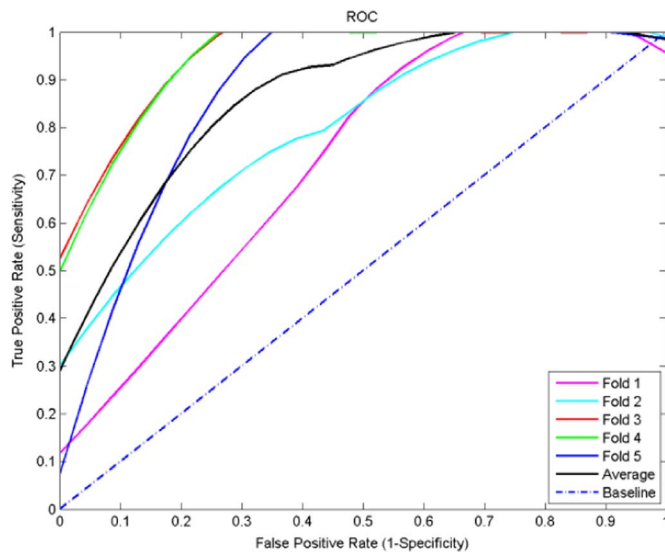


Fig. 11. ROC Curves for RUS-SVM for the first testing case.

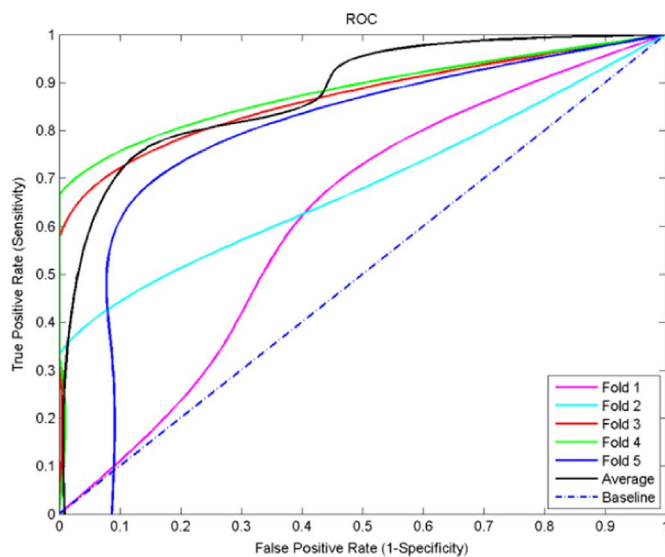


Fig. 12. ROC Curves for SMOTE-SVM for the first testing case.

smaller dataset and hence a shorter processing time. The KerMinSVM classifier has a slightly longer processing time than the standard SVM classifier, which means that there is no large overhead in processing time for the KerMinSVM.

4.3. Short arabic text classification

To test the various themes found in comic books, a dataset of Arabic comics was collected from different comic books and magazines sold in the Middle East (Basem, Fulla, Mahdi, Ahmad). The dataset consists of 128 comics divided into 3 categories: 113 which do not have a religious theme, 10 religious comics comprising positive themes with no strong content, and finally 5 religious themed comics that include strong content unsuitable for children. The text length in these comics ranges between (25–460) words, or (91–2030) characters, as listed in Table 5.

Three different tests are applied to assess the efficiency of the new approach.

1. Normal comics vs. religious comics with positive and strong content.
2. Normal comics and positive religious comics vs. religious comics containing strong content.
3. Positive religious comics vs. religious comics with strong content.

It is clear that the Arabic comic dataset and its classes are imbalanced, so it is necessary to adopt the imbalanced data classification method. To test the data sets, KerMinSVM is compared to the standard SVM, RUS-SVM and SMOTE SVM. The proposed approach is compared to the standard one, which takes the words as features without filtering or stemming and therefore yielded a feature vector of length 6204, while the proposed new approach had a feature vector length of 1163, with almost 5.3 times fewer features. It is noteworthy that the root based grouping reduction of the feature vector length performed well for this comic dataset built for this study, but it has not been tested on generic Arabic datasets. A five-fold cross-validation is applied on these data while maintaining the imbalance ratio for each fold. The accuracy measures are then computed and averaged over the five folds.

The results in Tables 6 and 7 and Figs. 9–12 clearly demonstrate that the proposed approach resulted in better accuracy than the standard approach, with the best results being obtained from KerMinSVM. This shows that KerMinSVM, when combined with the new method for preprocessing, can handle classifying unbalanced short text efficiently. (Table 8)

To further validate KerMinSVM, we tested on the ‘Twitter Data set for Arabic Sentiment Analysis’ (Abdulla et al., 2013). The data set contains tweets on a range of topics including politics and arts written in

Table 8

Test results for the standard approach.

Data	Standard vs. Islamic with positive and violent themes					
Classifier	Sensitivity	Specificity	Accuracy	Recall	Precision	F-measure
KerMinSVM	0.73	0.76	0.75	0.29	0.73	0.40
SVM	0.87	0.21	0.29	0.13	0.87	0.22
RUS-SVM	0.25	1	0.89	1	0.25	0.40
SMOTE-SVM	0.53	0.51	0.52	0.20	0.53	0.29
Data	Standard and positive Islamic vs. Islamic containing violent content					
Classifier	Sensitivity	Specificity	Accuracy	Recall	Precision	F-measure
KerMinSVM	0.40	0.93	0.91	0.09	0.4	0.17
SVM	0.40	0.95	0.93	0.13	0.40	0.19
RUS-SVM	0.50	1	0.96	1	0.5	0.67
SMOTE-SVM	0	1	0.96	0	0	0
Data	Positive Islamic comics vs. Islamic comics with violent content					
Classifier	Sensitivity	Specificity	Accuracy	Recall	Precision	F-measure
KerMinSVM	0.80	0.70	0.73	0.57	0.80	0.63
SVM	0.20	0.80	0.60	0.17	0.20	0.18
RUS-SVM	0.80	0.70	0.75	0.80	0.80	0.77
SMOTE-SVM	0.20	1	0.73	0.20	0.20	0.20

Modern Standard Arabic and the Jordanian dialect. The entries convey positive and negative feelings that are manually annotated by native Arabic speakers. In total, the data set contains 1000 positive and 1000 negative entries. To demonstrate the performance of KerMinSVM on an imbalanced data set, 26 Negative entries and 226 Positive entries were chosen randomly. The same processing techniques were applied to the short Arabic texts provided, namely segmentation, filtering, stemming, and grouping before finally extracting the feature vector. Using five-fold cross validation, KerMinSVM achieved an accuracy rate of 88%, versus 90%, 88% and 87% for SMOTE SVM, SVM and NN, respectively.

5. Conclusion

In this paper, we introduced the KerMinSVM classifier, which is a modification of the original SVM and designed to solve the problem of learning imbalanced datasets. As shown in the experimental section, KerMinSVM outperformed other techniques for learning an imbalanced dataset. KerMinSVM has a higher sensitivity and F-Measure than the normal SVM and the other techniques, and it does not sacrifice the specificity of the data. Moreover, KerMinSVM is computationally efficient, since it does not require significant processing time compared to data-oversampling. We also built and manually annotated a database of Arabic comics on which we performed a data specific feature reduction. Our proposed approach has better performance than the standard approach, especially when KerMinSVM is used as the classifier. The testing results on our dataset show that the proposed approach for short Arabic text classification with MinSVM can handle classifying short Arabic text when the dataset is linearly non-separable and imbalanced.

Acknowledgments

This work is partly supported by the Qatar National Research Foundation (QNRF) and partly by the University Research Board at the American University of Beirut. The authors would like to thank Yara Rizk, Wissam Marrouche and Mohamad Kamareddine for their help in formatting the paper.

References

- Abdulla, N., et al. 2013. Arabic sentiment analysis: Corpus-based and lexicon-based. Proceedings of The IEEE conference on Applied Electrical Engineering and Computing Technologies (AEECT).
- Ajeeb, N., Nayal, A., Awad, M., 2013. "Minority SVM for linearly separable imbalanced datasets, In: Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1–5.
- Akbani, R., Kwek, S., Japkowicz, N., 2004. Applying support vector machines to imbalanced datasets. *Mach. Learn. ECML 2004*, 39–50.
- Alcalá, J., et al., 2010. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *J. Mult.-Value. Log. Soft Comput.* 17. 2-3, 255–287.
- Al-Harbi, S., Almuhareb, A., Al-Thubaity, A., Al-Rajeh, M., Khorsheed, A., 2008. Automatic Arabic text classification.
- Awad, M., Khanna, R., 2015. Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers, Apress
- Batuwita, R., Palade, V., 2010. FSVM-CIL: fuzzy support vector machines for class imbalance learning, in: *IEEE Transactions on Fuzzy Systems*, pp. 558–571
- Bollegala, D., Matsuo, Y., Ishizuka, M., 2007. Measuring semantic similarity between words using web search engines, In: *Proceedings of WWW*, p. 766.
- Charalampopoulos, I., Anagnostopoulos, I., 2011. A comparable study employing weka clustering/classification algorithms for web page classification. *Informatics (PCI)*, 2011 In: *Proceedings of the 15th Panhellenic Conference on IEEE*.
- Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2011. SMOTE: synthetic minority over-sampling technique. *arXiv Prepr. arXiv:1106.1813*.
- Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W., 2003. SMOTEBoost: Improving prediction of the minority class in boosting, in: *Knowledge Discovery in Databases: PKDD*, pp. 107–119
- Chen, M., Jin, X., Shen, D., 2011. "Short text classification improved by learning multi-granularity topics," In: *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Three*, pp. 1776–1781.
- Cieslak, David A., et al., 2012. Hellingr distance decision trees are robust and skew-insensitive. *Data Min. Knowl. Discov.* 24.1, 136–158.
- El-Halees, A., 2007. Arabic text classification using maximum entropy. *Islam. Univ. J. (Ser. Nat. Stud. Eng.)* 15, 157–167.
- El Kourdi, M., Bensaid, A., Rachidi, T., 2004. "Automatic Arabic document categorization based on the Naive Bayes algorithm," In: *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, pp. 51–58.
- Fago, Z., Fan, Z., Bingru, Y., Xingang, Y., 2010. Research on short text classification algorithm based on statistics and rules, in *Electronic Commerce and Security (ISECS)* In: *Proceedings of the Third International Symposium on*, pp. 3–7.
- Gazzah, S., Amara, N., 2008. New oversampling approaches based on polynomial fitting for imbalanced data sets. In: *Proceedings of the Eighth IAPR International Workshop on Document Analysis Systems*, 2008. DAS'08. IEEE.
- Holmes, G., Donkin, A., Witten, I., 1994. Weka: A machine learning workbench. *Intelligent Information Systems*, 1994. In: *Proceedings of the 1994s Australian and New Zealand Conference on IEEE*.
- Hu, X., Sun, N., Zhang, C., Chua, T., 2009. "Exploiting internal and external semantics for the clustering of short texts using world knowledge, in proceedings In: *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 919–928.
- Imam, T., Ting, K.M., Kamruzzaman, J., 2006. z-svm: an svm for improved classification of imbalanced data. *AI 2006: Adv. Artif. Intell.*, 264–273.
- Jomaa*, H., Kamareddine*, M., Nayal*, A., Rizk*, Y., Awad, M., 2016. Affective Relationship Between Color and Text in Arabic Comic books, In: *Proceedings of the 12th International Conference on Signal-Image Technology & Internet-Based Systems*, Nov 28–Dec 2, 2016
- Jomaa, H.S., Awad, M., Ghaibeh, L., 2015. Panel tracking for the extraction and the classification of speech balloons. *Image Anal. Process.*, 394–405.
- Khoja, S., Garside, R., 1999. Stemming arabic text. Lancaster, UK, Computing Department, Lancaster University.
- Khraisat, L., 2006. Arabic text classification using N-gram frequency statistics a comparative study, In: *Proceeding of the Conference on Data Mining| DMN*, p. 79.
- Kubat, M., Matwin, S., others, 1997. "Addressing the curse of imbalanced training sets: one-sided selection, In: *Proceeding of the Machine Learning-international Workshop Then Conference*, pp. 179–186.
- Liu, Wei, et al., 2010. A robust decision tree algorithm for imbalanced data sets. *SDM 10*.
- Moh'd, A., Mesleh, A., 2007. Chi square feature extraction based SVMs Arabic language text categorization system. *J. Comput. Sci.* 3, 430–435.
- Monroe, Will, Green, Spence, Manning, Christopher D., 2014. Word segmentation of informal arabic with domain adaptation. *ACL (2)*.
- Napierala, K., Stefanowski, J., Wilk, S., 2010. Learning from imbalanced data in presence of noisy and borderline examples, in: *Rough Sets and Current Trends in Computing*, pp. 158–167
- Padmaja, T.M., Dhulipalla, N., Bapi, R.S., Radha Krishna, P., 2007. "Unbalanced data classification using extreme outlier elimination and sampling techniques for fraud detection," In: *Proceeding of the International Conference on Advanced Computing and Communications, (ADCOM 2007)*, pp. 511–516.
- Phan, X., Nguyen, L., Horiguchi, S., 2008. "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," In: *Proceedings of the 17th international conference on World Wide Web*, pp. 91–100.
- Rizk, Y., Awad, M., 2012. Syntactic Genetic Algorithm for a Subjectivity Analysis of Sports Articles', *International Conference on Cybernetic Intelligent Systems, At Limerick, Ireland*.
- Sahami, M., Heilman, T.D., 2006. "A web-based kernel function for measuring the similarity of short text snippets," In: *Proceedings of the 15th international conference on World Wide Web*, pp. 377–386.
- Sawaf, H., Zaplo, J., Ney, H., 2001. *Statistical Classification Methods for Arabic News Articles*.
- Seiffert, C., Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A., 2010. RUSBoost: A hybrid approach to alleviating class imbalance, in: *Systems, Man and Cybernetics, Part A: IEEE Transactions on Systems and Humans*, pp. 185–197
- Sun, Yanmin, et al., 2007. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognit.* 40.12, 3358–3378.
- Tang, Y., Zhang, Y., Chawla, N.V., Krasser, S., 2009. SVMs modeling for highly imbalanced classification. *Syst. Man Cybern. Part B: Cybern.* IEEE Trans. 39, 281–288.
- Tax, D.M.J., Duin, R.P.W., 2004. Support vector data description. *Mach. Learn.* 54, 45–66.
- Thabtah, F., Hadi, W., Al-shammare, G., 2008. VSMs with K-Nearest neighbour to categorise Arabic text data.
- Veropoulos, K., Campbell, C., Cristianini, N., 1999. "Controlling the sensitivity of support vector machines," In: *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 55–60.
- Vo, N., Won, Y., 2007. Classification of unbalanced medical data with weighted regularized least squares. *Frontiers in the Convergence of Bioscience and Information Technologies, (FBIT 2007)*. IEEE.
- Wang, B.X., Japkowicz, N., 2010. Boosting support vector machines for imbalanced data sets. *Knowl. Inf. Syst.* 25, 1–20.
- Yih, M., Meek, M., 2007. "Improving similarity measures for short segments of text," In: *Proceedings of the National Conference on Artificial Intelligence*, p. 1489.
- Zelikovitz, S., Hirsh, H., 2000. "Improving short text classification using unlabeled background knowledge to assess document similarity," In: *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 1183–1190.