A Semantics Aware Random Forest for Text Classification

Md Zahidul Islam*
islmy008@mymail.unisa.edu.au
University of South Australia

Jixue Liu
Jixue.Liu@unisa.edu.au
University of South Australia

Jiuyong Li
Jiuyong.Li@unisa.edu.au
University of South Australia

Lin Liu
Lin.Liu@unisa.edu.au
University of South Australia

Wei Kang
Wei.Kang@data61.csiro.au
Data61, CSIRO, Australia

ABSTRACT

The Random Forest (RF) classifiers are suitable for dealing with the high dimensional noisy data in text classification. An RF model comprises a set of decision trees each of which is trained using random subsets of features. Given an instance, the prediction by the RF is obtained via majority voting of the predictions of all the trees in the forest. However, different test instances would have different values for the features used in the trees and the trees should contribute differently to the predictions. This diverse contribution of the trees is not considered in traditional RFs. Many approaches have been proposed to model the diverse contributions by selecting a subset of trees for each instance. This paper is among these approaches. It proposes a Semantics Aware Random Forest (SARF) classifier. SARF extracts the features used by trees to generate the predictions and selects a subset of the predictions for which the features are relevant to the predicted classes. We evaluated SARF's classification performance on 30 real-world text datasets and assessed its competitiveness with state-of-the-art ensemble selection methods. The results demonstrate the superior performance of the proposed approach in textual information retrieval and initiate a new direction of research to utilise interpretability of classifiers.

CCS CONCEPTS

• **Computing methodologies** → **Ensemble methods**; • **Information systems** → *Sentiment analysis*; *Clustering and classification.*

KEYWORDS

Random Forest, Ensemble Selection, Semantic Explanations

ACM Reference Format:
Md Zahidul Islam, Jixue Liu, Jiuyong Li, Lin Liu, and Wei Kang. 2019. A Semantics Aware Random Forest for Text Classification. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357891

*Corresponding author.

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357891

1 INTRODUCTION

Recently Fernández-Delgado et al. [15] empirically evaluated 179 classifiers (including 63 ensemble classifiers) on 121 datasets and concluded that Random Forest (RF) [4] offers the best performance. RF is also a high performer in text classification [31, 32, 37, 40]. RF mitigates the inherent challenges involved in textual data such as *high dimensionality*, *sparsity* and *noisy* feature space [20, 40]. In this article, we will improve the performance of RFs for text classification.

For the vast number of features in text data, a large number of trees are required in RFs which increases the chance of overfitting [14]. To construct a decision tree in an RF, a random subset of features are selected to use in the training. The randomly selected feature subsets may be sparse and noisy in some trees [1, 40] and this leads to unreliable predictions. Traditional RFs produce a prediction by combining predictions of all trees and some of the predictions are the unreliable ones. The inclusion of unreliable predictions affects the performance of the RF.

Ensemble selection methods have been used to improve the performance of RFs by selecting and using only a subset of trees based on some selection criteria or competency measures [3, 8, 14]. Static ensemble selection methods apply a selected subset of trees to all future test instances, this could decrease diversity and lose informative features for the predictions of individual test instances.

Dynamic ensemble selection (DES) methods can be used to select a subset of trees from an RF for each test instance. In these methods, the competency measures are derived by the performance of a classifier on the instances similar to the test instance [8, 29, 38, 39]. However, the similar instances are selected from the validation set consisting of a limited number of instances, hence the measure is biased towards the selection of validation set. Additionally, due to the noise and high dimensionality of text datasets, the conventional distance measures may not reflect the true similarity between instances. The limitation of distance measures for texts is evident from the experimental study presented in [37] where a series of comparative studies are performed among various classifiers individually and in ensemble settings for sentiment analysis and it is found that k Nearest Neighbour classifiers achieve the worst performance. As a result, the competency measures, evaluated on the set of similar instances of a test instance, may fail to work well with text datasets.

RFs cannot distinguish between contradictory predictions (different predicted classes for the same input) from the trees while combining them to generate the combined prediction. Both traditional RFs that uses all trees and RFs with selected trees use majority voting to combine the predictions. RFs use trees in a "block-box"

manner. Given two competent trees each predicting a different class label there is no way to tell which of the two is more reliable. Some methods proposed to use weighted voting where the weights are calculated from validation set performance [14, 16, 29, 43]. These methods are still dependent on the samples selected from the training set on which the performance of a tree is measured.

The "black-box" phenomenon in RF classifiers is illustrated in a naïve text classification problem shown in Fig. 1 where T_a and T_b are two trees in an RF. Here, the decision trees in the RF are built from the dataset shown on the left. For the instance `x="i dont like noisy cars"` to be predicted, T_a's prediction is `-1` (negative sentiment), and T_b's prediction is 1 (positive sentiment). In the combined prediction, the RF will weigh the two predictions equally i.e. the total votes towards both the classes will increase, even though the two predictions may not be equally reliable.

sentiment	text
1	cars are great
1	cars with red colors are wonderful
1	i like this car
-1	i dont like blue cars
-1	noisy cars are not good

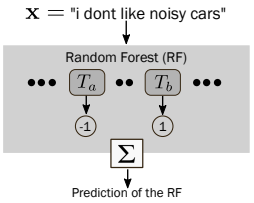

Figure 1: An example of "black-box" text classification in RF.

The reliability of a prediction can be evaluated from the features used by a decision tree to make the prediction. For example, a deeper analysis of the trees in Fig. 2 reveals that T_a predicts the `-1` class due to the features {`noisy`, `dont`} and T_b predicts the 1 class due to {`like`, `blue`}. Comparing these features with the training set (Fig. 1), we see that the features used by T_a are more relevant to the `-1` class since they appear only in the samples corresponding to the `-1` class. On the other hand, the feature `like` appears in both classes but `blue` appears in the samples corresponding to the `-1` class only. Hence, the prediction of 1 considering {`like`, `blue`} is not reliable.

Motivated by the above observations, we propose a novel Semantics Aware Random Forest (SARF) algorithm for text classification. SARF creates diverse decision trees using the RF algorithm. In the prediction phase, given a new instance, each decision tree is assessed on their prediction reliability for the instance. SARF defines

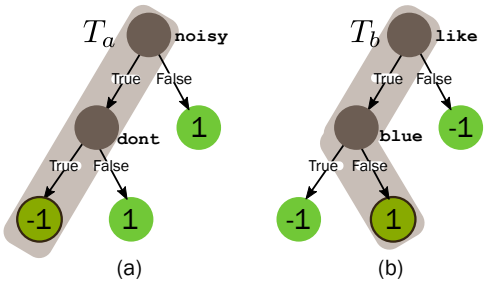

Figure 2: Unfolding the "black-box" classifications of `x = "i dont like noisy cars"`.

the competency in the explanatory space using the semantic explanations behind the predictions to select the reliable predictions for each instance. By semantics we refer to the explanations for a prediction i.e. the features used by a tree to make a prediction. SARF is also able to discriminate the supports of two trees on two classes. For example, SARF determines that the prediction of the tree T_b in Fig. 2 is not reliable as the features considered for the prediction are more relevant to the other class than the predicted class. Therefore, T_b is not used in producing the combined prediction of SARF. To our best knowledge, none of the existing DES methods considers the semantic explanations. Thus, SARF is expected to be the first RF framework incorporating a dynamic selection strategy leveraging the semantical information.

We note that, SARF is different from neighbors-based LazyNN_RF method presented in [31]. The main difference between LazyNN_RF and SARF is that for each test instance SARF selects a subset of predictions where as the LazyNN_RF selects a subset of training samples and trains an RF using the selected samples.

Since RF is a predominant and high performing classifier in use for many domains, our work contributes to the advancement of this important technique in text classification. The main contributions of this paper are:

- We introduce the concept of measuring the competency of a decision tree from explanatory features.
- We introduce SARF to comprehend semantic explanations for dynamically selecting reliable predictions in an RF and develop a method to evaluate the predictions using the explanations.
- We empirically evaluate SARF on 30 real-world text datasets and show that SARF effectively improves the classification performance of RFs.

The rest of the paper is organized as follows. The research problem is defined in Section 2. SARF is presented in Section 3 addressing the research problem. The experimental analysis of the proposed method is presented in Section 4. The related works are summarized in Section 5. Finally, the concluding remarks and directions for future improvements are presented in Section 6.

2 PROBLEM DEFINITION

Let a document d_i be represented using the bag-of-words model, i.e. d_i is a bag of words of $\{w_1, w_2, \ldots, w_m\}$. Let $\mathbf{X}^k = \{d_1, d_2, \ldots, d_n\}$ be the set of documents belonging to a class k. Also, let $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^K\}$ denote the document corpus and $\mathbf{Y} = \{1, 2, \ldots, K\}$ denote the posssible classes of documents in $\mathbf{X}$.

In an RF classifier, a set of decision trees is trained from $(\mathbf{X}, \mathbf{Y})$. Let $\mathcal{F} = \{T_1, T_2, \ldots, T_T\}$ denote the set of trees in the RF. For an instance $\mathbf{x}$, we intend to combine the predictions of the trees with reliable predictions.

Let $\mathbf{\Phi} = \{\varphi_1, \varphi_2, \ldots, \varphi_T\}$ denote the predictions of the decision trees in an RF. Our goal is to find a subset $\mathbf{\Phi}_x \subseteq \mathbf{\Phi}$ so that $\mathbf{\Phi}_x$ includes the reliable predictions only.

In this paper, we evaluate the reliability of a prediction φ_t by the words used by tree T_t (i.e. the semantics for φ_t) to make the corresponding prediction. We define a binary function $\tau_t : T_t, \varphi_t \rightarrow \{0, 1\}$, where 1 (or 0) indicates that φ_t is a reliable (or unreliable) prediction.

We expect y^* to come from the reliable predictions i.e. $\mathbf{\Phi}_x$. Hence, the combined prediction y^* of $\boldsymbol{\mathcal{F}}$ can be represented as:

$$y^* = \arg \max_{k \in \{1,2,\ldots,K\}} \{S_k | S_k = \sum_{t=1}^{T} I[\varphi_t = k] \times \tau_t(T_t, \varphi_t)\} \quad (1)$$

where S_k denotes the total support received by the class k from $\boldsymbol{\mathcal{F}}$ and $I[.]$ is an indicator function returning 1 when T_t predicts k.

That is, our intended RF classifier considers two factors in decision making. The first factor reflects the supports received by the class k, i.e. S_k and the second factor reflects the reliability of the predictions supporting k, i.e. τ_t. Important notations used throughout this paper are listed in Table 1.

To incorporate semantic explanations in RFs, we essentially search for the answers to the following sub-questions:

(1) Given a decision tree $T_t \in \boldsymbol{\mathcal{F}}$ and its prediction φ_t, how to extract the explanations to derive τ_t considering the explanations?
(2) Given a set of predictions $\mathbf{\Phi}$ generated by $\boldsymbol{\mathcal{F}}$, how to formulate the reliability function τ_t?

In the next section, we will present our answers to the above questions, and our approach to construct the SARF classifier.

3 SEMANTICS AWARE RANDOM FOREST (SARF)

The proposed SARF classifier has the following three main stages (Fig. 3):

- Extraction of the semantics (i.e., explanatory features) for each prediction by an individual tree (Sub-Problem 1).
- Identification of the reliable predictions using the explanatory words (Sub-Problem 2).
- Integration of the reliable predictions (Eq. 1).

3.1 Explanatory Feature Extraction

Before combining the predictions, SARF evaluates the prediction from each tree. To evaluate a prediction $\varphi_t \in \mathbf{\Phi}, 1 \leq t \leq T$, SARF inspects the features, called explanatory features, used to make the prediction φ_t. In bag-of-words [24] representation, these features are some of the words from the training corpus. They provide useful information regarding the reliability of φ_t [17, 28, 34].

Table 1: Important notations used in the paper.

Sym.	Description
d_i	a document having $\{w_1, w_2, \ldots, w_m\}$ bag of words.
$\mathbf{X}^k$	the set of documents corresponding to class k.
$\mathbf{X}$	a collection of documents having K distinct classes.
$\mathbf{Y}$	a set of K class labels.
T_t	a decision tree classifier.
$\boldsymbol{\mathcal{F}}$	a random forest (RF) with $\{T_1, T_2, \ldots, T_T\}$ trees.
φ_t	the prediction of the decision tree T_t.
$\mathbf{\Phi}$	the set of predictions of the decision trees in $\boldsymbol{\mathcal{F}}$.
$\mathbf{\Phi}_x$	the set of reliable predictions for an instance $\mathbf{x}$.
S_k	the support of the RF $\boldsymbol{\mathcal{F}}$ on class k.
y^*	the ensemble prediction of the RF $\boldsymbol{\mathcal{F}}$.

The explanatory features $\mathbf{e}_t = \{\omega_1, \omega_2, \ldots, \omega_l\}$ are not readily available in RF classifiers [2]. In the explanatory feature extraction phase, we want to extract $\mathbf{e}_t$ for a prediction φ_t. In an RF with T trees there are T sets of $\mathbf{e}_t$ (in text classifications, the value of T is typically between 200 to 500). The number of words from the root to a decision node in a tree can be of different sizes for different input instances depending on various parameters such as the maximum depth of the tree, the minimum number of samples considered to split a node etc. In addition, they can be very small (two or three out of thousands of features) in some trees and larger in the others for the same instance. Therefore, defining an appropriate measure to compare the features on multiple decision paths is difficult.

In this work, we use a framework, called LIME, introduced by Ribeiro et al. [28] to extract $\mathbf{e}_t$. Given a trained tree T_t and instance $\mathbf{x}$, LIME provides the explanatory features for the prediction φ_t. LIME determines $\mathbf{e}_t$ using the coefficients of a linear model fitted to a new dataset created by taking random samples near to $\mathbf{x}$ as predictors and the output of T_t on those samples as the target variable. The $\mathbf{e}_t$ is selected using Lasso. LIME converts the original feature space to an explainable space using a transformation function so that the returned features are meaningful. However, in this paper we consider the bag-of-words model representation, thus the features are self-explanatory i.e., the feature transformation is not required.

Ribeiro et al. [28] have showed that the fetures extracted by LIME are more than 90% similar to the features in the decision paths. We determine the size of $\mathbf{e}_t$ empirically through cross validation. Fixing the same size for all $\mathbf{e}_t$ solves the problem of comparing variable sized explanatory features. Moreover, LIME also makes our framework extensible to other ensemble methods involving different classifiers such as Naïve Bayes, Artificial Neural Network, Support Vector Machines etc.

We use the example introduced in Fig. 2 to show the results of explanatory feature extraction. For the two decision trees T_a and T_b, we extract the following features:

$$\mathbf{e}_a = \{\texttt{noisy}, \texttt{dont}\}$$
$$\mathbf{e}_b = \{\texttt{like}, \texttt{blue}\}$$

In the next step, we evaluate the words in $\mathbf{e}_a$ and $\mathbf{e}_b$ to determine the reliability of the predictions $\{-1, 1\}$.

3.2 Reliability Measurement

As discussed in [17, 28] and our example (Fig. 2), $\mathbf{e}_t$ can be evaluated to determine the reliability of φ_t. Our intuition is that $\mathbf{e}_t$ will include words closest to the training samples corresponding to the predicted class. For example, if $\varphi_t = i, 1 \leq t \leq T$ and $1 \leq i \leq K$ then $\mathbf{e}_t$ should include words relevant to the samples having class label i. In the case when $\mathbf{e}_t$ includes more words from the documents corresponding to the predicted class than the documents corresponding to the other classes, $\tau_t(T_t, \varphi_t)$ should return 1. Consequently, if $\mathbf{e}_t$ includes words more relevant to the documents of class $j, 1 \leq j \leq K$ and $i \neq j$ then φ_t is not reliable i.e. $\tau_t(T_t, \varphi_t)$ should return 0. Likewise, $\tau_t(T_t, \varphi_t)$ returns 0 when $\mathbf{x}$ does not have any words from the training set.

To define the reliability function τ_t, we need to measure the relevance of $\mathbf{e}_t$ with the words in documents corresponding to each

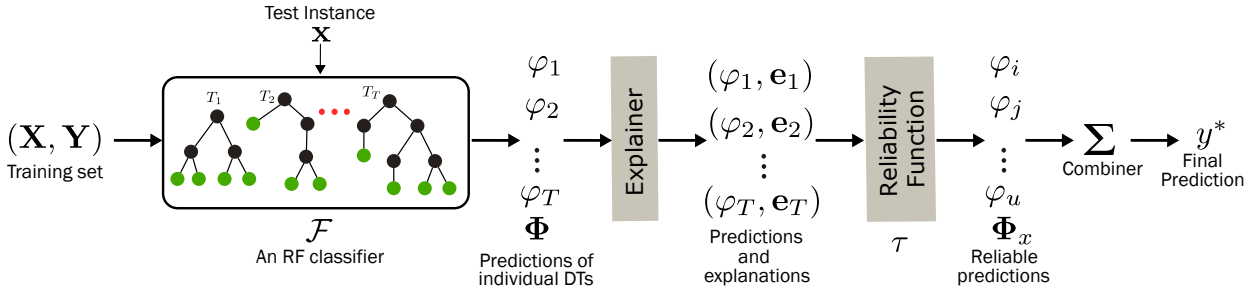

Figure 3: An overview of text classification using SARF.

target class. We define a relevance function $\rho : \mathbf{e}_t, k \rightarrow \mathbb{R}^+$ to assign a relevance score to a set of explanatory words $\mathbf{e}_t, 1 \leq t \leq T$ and target class $k, 1 \leq k \leq K$. Recall that $\mathbf{X}^k$ are the samples corresponding to class k. For an instance $\mathbf{x}$ and a prediction φ_t, there are K such relevance scores $\{\rho(\mathbf{e}_t, 1), \rho(\mathbf{e}_t, 2), \ldots, \rho(\mathbf{e}_t, K)\}$. According to our intuition, we expect $\rho(\mathbf{e}_t, k)$ to have the maximum value if $\varphi_t = k$. We now define τ_t as follows:

$$\tau_t(T_t, \varphi_t) = \begin{cases} 1, & \text{if } \varphi_t = \arg\max_k \{\rho(\mathbf{e}_t, k)\} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Hence, τ_t considers the explanatory features $\mathbf{e}_t$, extracted from T_t, and the prediction φ_t to decide the reliability of φ_t.

In the first step, we build the RF using the bag of words with binary weights and extract the explanatory features. In this step, we measure the relevance of each word $\omega \in \mathbf{e}_t$ to the words in $\mathbf{X}^k$ and add them to get the relevance score of $\mathbf{e}_t$ and k. The individual relevance is measured using the well-known term frequency-inverse document frequency (TFIDF) metric [33].

In general, the TFIDF weight represents the relevance of a word in a document to a corpus. Mathematically TFIDF weight of a word w in a document d_i of class k can be defined as follows [33]:

$$\text{TFIDF}_{w,d_i,k} = (1 + \log(\text{TF}_{w,d_i,k})) \times \log\left(1 + \frac{|\mathbf{X}^k|}{|\mathbf{X}^k_w|}\right) \tag{3}$$

where, $\text{TF}_{w,d_i,k}$ is the frequency of w in $d_i \in \mathbf{X}^k$ and $|\mathbf{X}^k|$ and $|\mathbf{X}^k_w|$ are the number of documents of class k and the number of documents of class k having w. As given below, we sum up the individual $\text{TFIDF}_{w,d_i,k}$ weights across all documents in $\mathbf{X}^k$ to establish the relevancy of w to k, as proposed in [21].

$$\text{TFIDF}_{w,k} = \sum_{i=1}^{|\mathbf{X}^k|} \text{TFIDF}_{w,d_i,k} \tag{4}$$

We now define our relevance function ρ as follows, which assigns a relevance score for each set of explanatory words $\mathbf{e}_t, 1 \leq t \leq T$ and each target class $k, 1 \leq k \leq K$.

$$\rho(\mathbf{e}_t, k) = \sum_{i=1}^{|\mathbf{e}_t|} \text{TFIDF}_{\omega_i,k} \tag{5}$$

Note that Eq. 5 will return 0 for all $k, 1 \leq k \leq K$ if there is no common word in the training features and $\mathbf{x}$. We set τ_t to 0 in those cases.

We continue our example in Fig. 2. Using the relevance measure ρ, we calculate the relevance scores of the explanatory words $\mathbf{e}_a$ and $\mathbf{e}_b$ for our example presented in Fig. 2 as follows:

$$\begin{aligned} \rho(\mathbf{e}_a, -1) &= 1.005 \\ \rho(\mathbf{e}_a, 1) &= 0 \\ \rho(\mathbf{e}_b, -1) &= 1.068 \\ \rho(\mathbf{e}_b, 1) &= 0.652 \end{aligned}$$

Subsequently, using τ_t we get $\tau_a(T_a, \varphi_a) = 1$ and $\tau_b(T_b, \varphi_b) = 0$.

Applying τ to $\mathbf{\Phi}$ and $\mathcal{F}$ we get the set of reliable predictions $\mathbf{\Phi}_x$. We aggregate the predictions in $\mathbf{\Phi}_x$ to produce y^* in the next step.

3.3 Prediction Combination

In order to aggregate the supports received by each target class $k, 1 \leq k \leq K$, we apply a combination method for abstract level predictions namely the majority voting (MV) [19]. Given T' predictions in $\mathbf{\Phi}_x, T' \leq T$, the MV combination method counts the number of votes for each class which can be expressed as follows:

$$S_k = \sum_{t=1}^{T'} I[\varphi_t = k] \tag{6}$$

where $I[.]$ is an indicator function returning 1 when T_t predicts k.

In case of ties, we calculate the average relevance scores for the $\mathbf{e}_t$ corresponding to the same class after selection. Hence, we define the modified vote counts as:

$$S'_k = \frac{1}{|S_k|} \sum_{t=1}^{T'} (\rho(\mathbf{e}_t, k) | \varphi_t = k) \tag{7}$$

After aggregating the support for each target class, MV selects the class with the maximum vote as the ensemble prediction y^* [19].

We argue that exploring the semantic explanations through this novel ensemble classifier reduces the risk of blindly predicting the class supported by the majority of the trees. In SARF, only the reliable predictions are combined using the simple yet powerful MV combination method and reliabilities are determined using the semantic explanations behind the predictions.

3.4 The SARF Algorithm

We summarize the proposed method in Algorithm 1. The algorithm starts with a set of decision trees $\mathcal{F}$ built from a text corpus $(\mathbf{X}, \mathbf{Y})$ according to the RF algorithm. Given a new instance $\mathbf{x}$, it applies the trees $\mathcal{F}$ to get the set of predictions $\mathbf{\Phi}$. Next, SARF finds the reliable predictions $\mathbf{\Phi}_x$ by applying the function $\tau_t(T_t, \varphi_t)$ on the

predictions Φ. The function $\tau_t(T_t, \varphi_t)$ involves extraction of the explanatory features $\mathbf{e}_t$ and calculation of relevance scores $\rho(\mathbf{e}_t, k)$. The support for each class S_k is computed from the reliable predictions Φ_x. If multiple classes receive the maximum support then the average reliability scores of the predictions supporting the classes are used to break the ties. Finally, the class receiving maximum support is returned as the prediction of the RF $\mathcal{F}$.

Algorithm 1 Semantics Aware Random Forest (SARF)

Input: A set of decision trees $\mathcal{F} = \{T_1, T_2, \ldots, T_T\}$ obtained using the RF algorithm, the training corpus $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^K\}$ having samples from the set of target classes $\mathbf{Y} = \{1, 2, \ldots, K\}$ and a test instance $\mathbf{x}$.
Output: Ensemble prediction y^* for $\mathbf{x}$.

for $i \in \{1, 2, \ldots, T\}$ **do**
  $\varphi_i \leftarrow T_i(\mathbf{x})$
  $\Phi \leftarrow \Phi \cup \varphi_i$
//Find the reliable predicions Φ_x from Φ.
for $i \in \{1, 2, \ldots, T\}$ **do**
  $\tau_i(T_i, \varphi_i) \leftarrow$ using Eq. 2 and Eq. 5
  if $\tau_i(T_i, \varphi_i) \neq 0$ **then**
    $\Phi_x \leftarrow \Phi_x \cup \varphi_i$
$S_k \leftarrow$ using Eq. 6
$\mathbf{O} \leftarrow \{\forall_{1 \leq k \leq K} \arg\max_k \{S_k\}\}$
if $|\mathbf{O}| = 1$ **then**
  $y^* \leftarrow \arg\max_k \{S_k\}$
else
  $S'_k \leftarrow$ for each $k \in \mathbf{O}$ using Eq. 7
  $y^* \leftarrow \arg\max_k \{S'_k\}$

4 EXPERIMENTS

To evaluate SARF, we apply the algorithm to 30 real-world text datasets. We discuss the settings used to develop the empirical study in Section 4.1. The details of the datasets used in the evaluation are provided in Section 4.1.1. The implementation of the RF and five state-of-the-arts DES methods included in this study are described in 4.1.2. The metrics and statistical tests for comparing the performance of the studied methods are specified in Section 4.1.3. The result of our empirical study is presented in Section 4.2. In the first experiment, we show that the studied DES methods do not improve the performance of the RF in Section 4.2.1. Then, we present an analysis to show the effectiveness of SARF in Section 4.2.2.

4.1 Experimental Setup

4.1.1 Datasets. Table 2 provides a summary of the 30 real-world text datasets used in our experiments (after some preprocessing as described later). The datasets are collected from various sources and include diverse types of texts such as user reviews, comments, tweets, newsgroups etc. They are frequently used in text classification research e.g. [22–24]. The datasets also vary in terms of size, writing style, and class distribution as shown in Table 2. In Table 2, a unique identifier (ID), a brief description of the contents (Content), the number of instances (#Ins.), the average number of words (#Aw.) per instance, the number of unigram features (#Feat.), the distribution of class (#Dc.) and the source of the dataset (Source) are included for each dataset.

Dataset preparation. Most of the datasets in our experiments have been used for sentiment classification tasks except - NGAM, NGBH, NGPM, NGGM, R8CM, R8CT, R8MI, R8TM, TWKG, and SPHM. The NGAM, NGBH, NGPM, and NGGM are subsets of the popular 20Newsgroup[1] dataset. The R8CM, R8CT, R8MI, and R8TM are subsets of the Reuters[2] dataset. The description of the subsets creation process is described below. The SPHM dataset consists of SMS messages classified into spam and not spam (ham) messages.

For simplicity, we design our experiments for binary classification. For datasets involving more than two classes, such as 20Newsgroup and Reuters, we take the samples having two class labels. For example, each of NGAM, NGBH, NGPM, and NGGM consists of documents of two topics out of the 20 topics in the complete dataset. Similarly, each R8CM, R8CT, R8MI, and R8TM consists of articles having two class labels. The TWKG dataset includes tweets related to Ebola, Malaria and Meningitis. We took all samples from the Ebola and Malaria classes to make it a two class dataset. For the sentiment analysis datasets involving more than two classes, we only take the instances with positive and negative labels and ignore the other classes.

For the sentiment datasets where numerical ratings are provided, we categorize the numerical values to positive and negative labels. The SSMS, SSDG, SSRW, SSBB, SSYT, and TWSS datasets provide mean positive and negative scores for each instance. The TWVD dataset provides a positive or negative score for each instance. We compared the provided values to categorize the instances.

As a preprocessing step, we removed the punctuations and stop words. Additionally, for the Tweet datasets, we removed the hashtags, emojis and URLs. We avoid more complex preprocessing such as stemming or lemmatization as they do not improve the performance of text classification significantly [6]. Reducing the feature space increases the chance of overfitting and less diverse trees in the RF. In the cases where stemming or lemmatization improves performance, we expect SARF to also improve in those cases.

4.1.2 The RF classifier and baseline for comparison. For the experiments, an RF is created with 200 fully grown trees according to Breiman's RF algorithm [4] implemented in the Python Scikit-learn machine learning library (http://scikit-learn.org/). The default parameter settings provided by the implementation in the Scikit-learn library are used. We choose 200 trees due to the lower computational cost and the finding that 300 and 500 trees do not improve statistically significant performance over 200 trees [23].

We use unigram features with binary weighting to train the RF as it showed better results in [24]. For some datasets with a large number of unigrams, for example, RVMV, NGGM, R8TM etc. we consider the unigrams appearing in at least six instances.

We experimented with a different number of words (to be returned by LIME) for $\mathbf{e}_t$ and choose to use 6, 8 or 10 depending on the number of average words per instance in the dataset.

Baseline DES methods. TWe compare SARF with traditional RF and five well-known algorithms reported to achieve the best performance in DES literature [5, 8, 9, 16]. Each of the DES methods is applied to an RF with 200 decision trees. A brief overview of the compared methods are given below:

[1] http://qwone.com/~jason/20Newsgroups/
[2] http://www.daviddlewis.com/resources/testcollections/reuters21578/

Table 2: A summary of the datasets used in the experiments.

ID	Content	#Ins.	#Aw.	#Feat.	#Dc.	Source
RVLP	Reviews of laptops	176	109	9540	88/88	http://cs.coloradocollege.edu/~mwhitehead
RVLW	Reviews of lawyers	176	30	1635	85/91	http://cs.coloradocollege.edu/~mwhitehead
RVTV	Reviews of TV shows	302	20	1799	145/157	http://cs.coloradocollege.edu/~mwhitehead
RVMU	Reviews of music albums	578	79	6632	289/289	http://cs.coloradocollege.edu/~mwhitehead
RVRD	Reviews of radio shows	708	24	3555	388/320	http://cs.coloradocollege.edu/~mwhitehead
RVDR	Reviews of drugs	741	42	3609	384/357	http://cs.coloradocollege.edu/~mwhitehead
RVDC	Reviews of doctors	1204	28	4768	633/571	http://cs.coloradocollege.edu/~mwhitehead
RVMV	Reviews of movies	1999	328	13345	999/1000	http://www.cs.cornell.edu/people/pabo
SSMS	Comments on Myspace	599	16	2276	88/511	http://sentistrength.wlv.ac.uk
SSDG	Comments on Digg	631	19	4272	486/145	http://sentistrength.wlv.ac.uk
SSRW	Comments on Runners World	642	38	4739	203/439	http://sentistrength.wlv.ac.uk
SSBB	Comments on BBC news	711	38	6591	628/83	http://sentistrength.wlv.ac.uk
SSYT	Comments on YouTube	1510	14	5849	533/977	http://sentistrength.wlv.ac.uk
TW14	Tweets on various entities	299	10	1194	145/154	http://www.sentiment140.com
TWSN	Tweets on Apple, Google, Microsoft and Twitter	905	11	2540	482/423	http://www.sananalytics.com
TWKG	Tweets related to Ebola, Malaria and Meningitis	1432	13	3340	809/623	https://www.kaggle.com/kandahugues
TWHC	Tweets regarding health care reform in the USA during March, 2010	1814	13	4351	1318/496	https://bitbucket.org/speriosu
TWSS	Public tweets on random topics	1825	11	5487	769/1056	http://sentistrength.wlv.ac.uk
TWVD	Tweets including common syntactical and grammatical language features	2612	10	6699	833/1779	https://github.com/cjhutto
TWTS	Tweets on generic topics	2722	13	7587	918/1804	http://www.mpi-inf.mpg.de/~smukherjee
NGGM	Political newsgroup documents related to guns and middle east	1833	206	3443	929/904	http://ana.cachopo.org
NGPM	Newsgroup documents related to PC and Mac hardware	1910	96	3371	942/968	http://ana.cachopo.org
NGBH	Sports documents regarding baseball and hockey	1947	127	4900	981/966	http://ana.cachopo.org
NGAM	Newsgroup documents regarding automobile and motorcycles	1955	104	4716	986/969	http://ana.cachopo.org
R8MI	Reuters documents labelled money-fx and interest	448	104	2431	199/249	http://ana.cachopo.org
R8CM	Reuters documents labelled crude and money-fx	478	120	1915	229/249	http://ana.cachopo.org
R8TM	Reuters documents labelled trade and money-fx	549	131	3477	249/300	http://ana.cachopo.org
R8CT	Reuters documents labelled crude and trade	629	137	4065	300/329	http://ana.cachopo.org
SPHM	SMS spam messages from the Grumbletext website	3689	14	7170	739/2950	http://www.comp.nus.edu.sg/~rpnlpir
KGMV	Short discussion on movies	7227	14	7767	3564/3663	https://www.kaggle.com/pirateshadow

- **RF**: The traditional RF [4] combining the predictions of all the decision trees.
- **KNORA**: The K-nearest-oracles (KNORA) [18] selects a subset of the decision trees by evaluating them on K neighbours from a validation set. We compare two variations of KNORA.
  KNORAE. The trees with less than 100% accuracy on the K neighbours are eliminated from the RF. The selected trees are combined using the MV method.
  KNORAU. The trees correctly classifying at least one neighbour are selected. The prediction of each decision tree is weighted by the number of correctly classifying neighbours. The ensemble result is obtained by the weighted MV method.
- **DESP**: The DES-Performance (DESP) [38] method selects the decision trees achieving performances more than random guessing i.e., 0.5 on the K nearest neighbours. The MV is used as the combination method.
- **METADES**: The METADES framework [8, 10] determines a competency score for each tree using a meta-classifier (a multinomial naíve Bayes in our experiments). The decision trees with competency scores higher than a pre-defined threshold are combined using a weighted MV method where weights are assigned according to the competency scores.
- **DESMI**: The DES method for multi-class imbalanced datasets (DESMI) [16] selects the set of classifiers contributing to all the target classes and combined them using a weighted MV method. The weights are calculated emphasizing the competency of the trees on the minority class.

For the methods requiring K neighbours, we set the value of K to 7 as it achieved the best results in previous studies [8, 10]. For the **METADES** method, 50% of the training set and five different types of features (*local*, *global* accuracies, and *consensus* on the K neighbours, the degree of *confidence* of the decision tree for the test instance, and the *output profiles*) are used to train the meta-classifier. The competency threshold is set to 0.5. If no tree is selected by any of the above DES methods, all the trees are used for classification i.e., same as the RF. We use the implementations of the DES methods provided by DESlib [11] library.

4.1.3 Evaluation metrics. The performances of the methods are tested using 5-fold cross-validation experiments. We use the same 5-fold splits of each dataset for all investigated methods. Following [7], we use the micro-averaged (MicroF1) and macro-averaged F1 scores (MacroF1) as our performance metrics.

Furthermore, to observe whether significant differences exist among the performances of RF, the baseline methods, and SARF we perform the Wilcoxon signed-rank test, following [12, 16]. For the Wilcoxon test, we used the MacroF1 score which is not dominated by the majority class. To compare two methods $M1$ and $M2$, Wilcoxon signed-rank test ranks the absolute differences of their MacroF1 scores on all the datasets. Two sums of the ranks, namely, R^+ where $M1$ outperforms $M2$ and R^- where $M2$ outperforms $M1$ are computed. Finally, a z statistic is calculated using R^+ and R^-. We set the level of significance $\alpha = 0.05$ i.e., at 95% confidence level. In our experimental settings with 30 datasets, the null-hypothesis can be rejected with $\alpha = 0.05$ if $z < -1.96$.

4.2 Result and Discussion

We designed our experiments with two objectives. Firstly, we observed *the effect of the existing DES methods on RF*. Secondly, we

observed *the effect of the SARF method on RF*. The MicroF1 and MacroF1 scores of the compared methods for all 30 datasets are reported in Table 3.

4.2.1 Existing DES methods to improve RF. In this experiment, we analyse the impact of existing DES methods on RF. The objective of this experiment is to verify whether existing DES methods can improve the performance of RF for text classification. We consider the DES methods introduced in Section 4.1.2.

From Table 3, we observe that there is not much difference in the performance of RF, KNORAU, KNORAE, DESP, METADES, and DESMI. On the SSBB and R8CM datasets, all methods achieve the same performance. RF and METADES show the same performance on 8 datasets. METADES achieve the best performance on 12 datasets, KNORAU on 4 datasets, and both KNORAE and DESP achieve the best score on 3 datasets.

The results in Table 3 show that there are not significant difference among the performance of RF and existing DES methods, we confirmed it using standard Wilcoxon signed rank tests using the MacroF1 scores achieved by the methods. The statistical results are shown in Table 4, where R^+ corresponds to the sum of ranks for an existing DES method and R^- for the RF. In Table 4, we observe that there are no statistically significant differences the preformance of RF and compared method and in all cases R^- are smaller (i.e. better) than R^+. Hence, we can conclude that applying KNORAU, KNORAE, DESP, METADES and DESMI DES methods do not improve the performance of the RF classifier.

4.2.2 SARF to improve RF. In this experiment, we compare SARF with the RF and existing DES methods. The objective of this experiment is to verify whether existing SARF can improve the performance of RF for text classification.

From Table 3, we observe that SARF performs considerably better than the traditional RF and the compared DES methods in terms of both MicroF1 and MacroF1 scores. SARF achieves the best results on 20 out of 30 text datasets.

For some datasets, the performance improvements by SARF are quite high. More specifically, for RVLW, RVTV, RVRD, SSDG, and TW14 datasets the proposed method improves performance by more than 10%. In the RVLW dataset, among the compared DES methods, the best MicroF1 score 0.8063 is achieved by DESP and METADES methods and the best MacroF1 score 0.8035 is achieved by METADES whereas SARF achieves MicroF1 score 0.9487 and MacroF1 score 0.9483 improving more than 14%.

Similar to the previous experiment, we have performed the Wilcoxon tests to compare the statistical significance of the proposed method (SARF) with the DES methods included in our experiments (KNORAU, KNORAE, DESP, METADES, and DESMI). The results of the tests are shown in Table 5 where R^+ and R^- correspond to the sum of the ranks of SARF and one of the compared methods respectively. Observing Table 5 we find that SARF is significantly better than RF, KNORAU, KNORAE, DESP, METADES, and DESMI since the obtained p-values are lower than $\alpha(0.05)$.

In general, the RF classifier itself presents very high classification accuracy, especially when the dataset is large [15]. For many datasets, e.g. RVLW, RVTV, SSBB, NGGM etc., the existing DES methods fail to improve the performance of RF. However, SARF improves the performance on most of the datasets and this indicates that SARF is able to improve the performance of RF using semantics based reliability measure.

Further analysis of the results elaborates that the main contributing factor behind the superior performance of SARF is the incorporation of semantics. For illustration, four instances from each of the RVLW and TW14 datasets are presented in Table 6. Both the datasets are labelled for sentiment analysis where the positive and negative sentiments are represented as 1 and −1 respectively. For each instance, the true class (True), prediction (Pred.) of a decision tree in the RF, the corresponding semantic explanations and reliability indicator (Reliability) are shown. An inspection of the features (i.e., words) in the first instance shows that the word {`screw`} is possibly highly correlated with the negative sentiment which is the predicted class. Hence, our reliability indicator marks it as a reliable prediction. However, in the second instance, the words {`cares, skills, ethical, like, advanced`} are imposing a positive sentiment but the decision tree predicted the negative sentiment and the reliability indicator correctly identified it as an unreliable prediction. SARF is able to correctly identify the unreliable predictions using the corresponding semantic explanations. Removing such unreliable predictions resulted in increased performance.

In summary, the experimental results demonstrate the superiority of our method over the other compared methods. SARF performs better for the datasets where the input texts are more formal e.g. the review datasets. On the other hand, for the texts having an informal writing style, e.g., tweets, SARF does not show better performance. SARF achieved a significantly poor result in the TWHR dataset. In such informal datasets, same words are written in many forms such as 'great', 'gr8', 'greaaaaat' etc. Therefore, the context of relevance is lost and it is difficult to develop a reliability function using word matching.

5 RELATED WORK

There are three directions of information retrieval research providing the theoretical foundation for our proposed method. Firstly, we owe the selection of RF to the research showing promising results of ensemble methods, especially RFs in text classification. Secondly, our idea of selecting a reliable subset of predictions is influenced by the research in ensemble selection. Finally, we credit the concept of evaluating individual predictions based on explanations to the research in interpretable machine learning. In this section, we give a brief overview of several developments in these areas.

Ensemble classifiers showed a significant increase in accuracy over single classifiers in several studies including [13, 15, 27, 30]. The popular methods for combining individual predictions include majority voting (MV), weighted majority voting, arithmetic operators such as sum, average, weighted average, product, maximum, minimum, and meta-classifiers such as stacking, the mixture of experts etc. [19, 27]. Recent applications of ensemble methods for text classification can be found in [23, 32, 37].

The use of RF classifiers in text classification, aimed at reducing the overfitting issues, are presented in [7, 31, 32]. In [32], a new sample weighting method based on out-of-bag error is presented for RF construction. The authors proposed a stacking based combination method for RFs in [7]. A dynamic RF construction method (for each test instance) based on training set selection is proposed

Table 3: The micro and macro average, and standard deviations of F1 scores for the compared methods and SARF. The best result in each dataset is highlighted in bold-face.

ID		RF	KNORAU	KNORAE	DESP	METADES	DESMI	SARF
RVLP	MicroF1	0.8072 ± 0.05	0.8124 ± 0.01	0.7725 ± 0.01	0.8124 ± 0.01	0.8072 ± 0.01	0.7386 ± 0.10	**0.8464** ± 0.07
	MacroF1	0.8049 ± 0.05	0.8101 ± 0.01	0.7662 ± 0.01	0.8101 ± 0.01	0.8049 ± 0.01	0.7329 ± 0.10	**0.8446** ± 0.07
RVLW	MicroF1	0.8063 ± 0.06	0.7892 ± 0.01	0.7497 ± 0.01	0.8063 ± 0.01	0.8063 ± 0.01	0.7556 ± 0.06	**0.9487** ± 0.04
	MacroF1	0.8035 ± 0.06	0.7864 ± 0.01	0.7465 ± 0.01	0.8034 ± 0.01	0.8035 ± 0.01	0.7505 ± 0.06	**0.9483** ± 0.03
RVTV	MicroF1	0.7420 ± 0.09	0.7387 ± 0.01	0.7319 ± 0.01	0.7387 ± 0.01	0.7420 ± 0.01	0.7420 ± 0.08	**0.8509** ± 0.03
	MacroF1	0.7392 ± 0.09	0.7361 ± 0.01	0.7287 ± 0.01	0.7361 ± 0.01	0.7392 ± 0.01	0.7385 ± 0.08	**0.8501** ± 0.03
RVMU	MicroF1	0.6747 ± 0.02	0.6782 ± 0.01	0.6643 ± 0.01	0.6747 ± 0.01	0.6747 ± 0.01	0.6730 ± 0.03	**0.7613** ± 0.03
	MacroF1	0.6738 ± 0.02	0.6769 ± 0.01	0.6629 ± 0.01	0.6738 ± 0.01	0.6738 ± 0.01	0.6720 ± 0.03	**0.7563** ± 0.03
RVRD	MicroF1	0.7429 ± 0.03	0.7387 ± 0.004	0.7232 ± 0.004	0.7387 ± 0.004	0.7429 ± 0.004	0.7246 ± 0.04	**0.8432** ± 0.04
	MacroF1	0.7307 ± 0.04	0.7275 ± 0.004	0.7134 ± 0.004	0.7272 ± 0.004	0.7307 ± 0.004	0.7161 ± 0.04	**0.8401** ± 0.03
RVDR	MicroF1	0.7087 ± 0.07	0.7074 ± 0.01	0.7020 ± 0.01	0.7074 ± 0.01	0.7087 ± 0.01	0.7154 ± 0.05	**0.7600** ± 0.05
	MacroF1	0.7071 ± 0.07	0.7059 ± 0.01	0.7003 ± 0.01	0.7057 ± 0.01	0.7071 ± 0.01	0.7127 ± 0.05	**0.7504** ± 0.05
RVDC	MicroF1	0.8688 ± 0.02	0.8663 ± 0.01	0.8680 ± 0.01	0.8680 ± 0.01	0.8688 ± 0.01	**0.8705** ± 0.03	0.8522 ± 0.03
	MacroF1	0.8683 ± 0.02	0.8659 ± 0.01	0.8676 ± 0.01	0.8675 ± 0.01	0.8683 ± 0.01	**0.8700** ± 0.03	0.8520 ± 0.03
RVMV	MicroF1	0.8284 ± 0.02	0.8264 ± 0.01	0.8054 ± 0.01	0.8269 ± 0.01	0.8284 ± 0.01	0.8129 ± 0.02	**0.8414** ± 0.02
	MacroF1	0.8283 ± 0.02	0.8263 ± 0.01	0.8053 ± 0.01	0.8268 ± 0.01	0.8283 ± 0.01	0.8128 ± 0.02	**0.8404** ± 0.02
SSMS	MicroF1	0.8564 ± 0.01	0.8564 ± 0.03	0.8531 ± 0.02	0.8564 ± 0.03	0.8564 ± 0.02	0.8565 ± 0.01	**0.8715** ± 0.01
	MacroF1	0.4812 ± 0.05	0.4812 ± 0.03	0.4874 ± 0.02	0.4812 ± 0.03	0.4812 ± 0.02	0.4996 ± 0.05	**0.5747** ± 0.04
SSDG	MicroF1	0.7718 ± 0.01	0.7702 ± 0.06	0.7655 ± 0.05	0.7718 ± 0.07	0.7718 ± 0.07	0.7766 ± 0.02	**0.8098** ± 0.01
	MacroF1	0.4612 ± 0.03	0.4606 ± 0.06	0.4583 ± 0.05	0.4612 ± 0.06	0.4612 ± 0.07	0.4911 ± 0.04	**0.5918** ± 0.02
SSRW	MicroF1	0.7056 ± 0.01	0.7056 ± 0.06	0.7055 ± 0.04	0.7056 ± 0.06	0.7056 ± 0.05	0.7056 ± 0.02	**0.7338** ± 0.04
	MacroF1	0.4901 ± 0.03	0.4901 ± 0.06	0.5288 ± 0.04	0.4905 ± 0.06	0.4901 ± 0.05	0.5166 ± 0.05	**0.5498** ± 0.08
SSBB	MicroF1	0.8833 ± 0.003	0.8833 ± 0.06	0.8833 ± 0.04	0.8833 ± 0.06	0.8833 ± 0.06	0.8833 ± 0.003	**0.8875** ± 0.01
	MacroF1	0.4690 ± 0.001	0.4690 ± 0.06	0.4690 ± 0.04	0.4690 ± 0.06	0.4690 ± 0.06	0.4690 ± 0.001	**0.5022** ± 0.04
SSYT	MicroF1	0.7477 ± 0.03	0.7477 ± 0.02	0.7563 ± 0.01	0.7484 ± 0.02	0.7477 ± 0.02	0.7563 ± 0.03	**0.8152** ± 0.01
	MacroF1	0.6605 ± 0.06	0.6601 ± 0.02	0.6801 ± 0.01	0.6610 ± 0.02	0.6605 ± 0.02	0.6851 ± 0.04	**0.7617** ± 0.02
TW14	MicroF1	0.7893 ± 0.06	0.7893 ± 0.02	0.7792 ± 0.03	0.7926 ± 0.03	0.7893 ± 0.02	0.7726 ± 0.05	**0.9030** ± 0.03
	MacroF1	0.7799 ± 0.06	0.7799 ± 0.02	0.7706 ± 0.03	0.7830 ± 0.03	0.7799 ± 0.02	0.7626 ± 0.05	**0.9018** ± 0.02
TWSN	MicroF1	0.7624 ± 0.04	0.7624 ± 0.04	0.7579 ± 0.03	0.7602 ± 0.04	0.7624 ± 0.03	0.7723 ± 0.04	**0.8221** ± 0.03
	MacroF1	0.7557 ± 0.04	0.7560 ± 0.04	0.7519 ± 0.03	0.7534 ± 0.04	0.7557 ± 0.04	0.7675 ± 0.04	**0.8219** ± 0.03
TWKG	MicroF1	0.9483 ± 0.02	0.9490 ± 0.09	0.9476 ± 0.08	0.9483 ± 0.09	0.9483 ± 0.09	0.9434 ± 0.02	**0.9616** ± 0.02
	MacroF1	0.9471 ± 0.02	0.9478 ± 0.09	0.9463 ± 0.08	0.9471 ± 0.09	0.9471 ± 0.09	0.9422 ± 0.02	**0.9610** ± 0.02
TWHC	MicroF1	0.7811 ± 0.01	0.7828 ± 0.003	0.7889 ± 0.003	0.7811 ± 0.003	0.7817 ± 0.003	**0.7922** ± 0.01	0.7337 ± 0.001
	MacroF1	0.6317 ± 0.02	0.6376 ± 0.001	0.6612 ± 0.001	0.6309 ± 0.001	0.6322 ± 0.001	**0.6700** ± 0.02	0.4481 ± 0.005
TWSS	MicroF1	0.6679 ± 0.02	0.6679 ± 0.01	0.6767 ± 0.01	0.6696 ± 0.01	0.6679 ± 0.01	0.6751 ± 0.03	**0.7551** ± 0.02
	MacroF1	0.6022 ± 0.03	0.6034 ± 0.03	0.6238 ± 0.03	0.6043 ± 0.03	0.6022 ± 0.03	0.6249 ± 0.04	**0.7154** ± 0.02
TWVD	MicroF1	0.8101 ± 0.02	0.8086 ± 0.01	0.8086 ± 0.02	0.8097 ± 0.01	0.8101 ± 0.01	**0.8136** ± 0.01	0.8017 ± 0.01
	MacroF1	0.7401 ± 0.03	0.7375 ± 0.05	0.7392 ± 0.07	0.7395 ± 0.05	0.7401 ± 0.05	**0.7480** ± 0.02	0.7151 ± 0.02
TWTS	MicroF1	0.6969 ± 0.02	0.6962 ± 0.01	0.7009 ± 0.02	0.6954 ± 0.01	0.6969 ± 0.01	**0.7028** ± 0.02	0.7017 ± 0.004
	MacroF1	0.5472 ± 0.02	0.5461 ± 0.02	0.5711 ± 0.04	0.5445 ± 0.02	0.5478 ± 0.03	**0.5798** ± 0.03	0.5137 ± 0.01
NGGM	MicroF1	**0.9924** ± 0.01	0.9907 ± 0.03	0.9891 ± 0.03	**0.9924** ± 0.03	**0.9924** ± 0.03	0.9896 ± 0.01	0.9858 ± 0.004
	MacroF1	**0.9924** ± 0.01	0.9907 ± 0.06	0.9891 ± 0.06	**0.9924** ± 0.06	**0.9924** ± 0.06	0.9896 ± 0.01	0.9858 ± 0.004
NGPM	MicroF1	0.9445 ± 0.01	**0.9455** ± 0.02	0.9403 ± 0.02	0.9450 ± 0.02	0.9445 ± 0.02	0.9403 ± 0.01	0.9440 ± 0.01
	MacroF1	0.9444 ± 0.01	**0.9455** ± 0.02	0.9402 ± 0.02	0.9449 ± 0.02	0.9444 ± 0.02	0.9402 ± 0.01	0.9439 ± 0.01
NGBH	MicroF1	0.9748 ± 0.01	0.9753 ± 0.004	0.9759 ± 0.002	0.9748 ± 0.01	0.9748 ± 0.01	0.9748 ± 0.01	**0.9795** ± 0.01
	MacroF1	0.9748 ± 0.01	0.9753 ± 0.01	0.9759 ± 0.004	0.9748 ± 0.01	0.9748 ± 0.01	0.9748 ± 0.01	**0.9795** ± 0.01
NGAM	MicroF1	0.9611 ± 0.01	0.9611 ± 0.02	0.9611 ± 0.01	0.9606 ± 0.01	0.9611 ± 0.01	0.9591 ± 0.01	**0.9637** ± 0.01
	MacroF1	0.9611 ± 0.01	0.9611 ± 0.02	0.9611 ± 0.02	0.9606 ± 0.03	0.9611 ± 0.03	0.9591 ± 0.01	**0.9637** ± 0.01
R8MI	MicroF1	0.8728 ± 0.01	0.8751 ± 0.02	0.8727 ± 0.01	0.8728 ± 0.02	0.8728 ± 0.02	0.8683 ± 0.02	**0.9018** ± 0.01
	MacroF1	0.8704 ± 0.01	0.8726 ± 0.02	0.8698 ± 0.01	0.8704 ± 0.02	0.8704 ± 0.02	0.8664 ± 0.02	**0.9002** ± 0.01
R8CM	MicroF1	**0.9983** ± 0.004	**0.9983** ± 0.04	**0.9983** ± 0.04	**0.9983** ± 0.04	**0.9983** ± 0.04	**0.9983** ± 0.004	0.9966 ± 0.01
	MacroF1	**0.9982** ± 0.004	**0.9982** ± 0.04	**0.9982** ± 0.04	**0.9982** ± 0.04	**0.9982** ± 0.04	**0.9982** ± 0.004	0.9965 ± 0.01
R8TM	MicroF1	0.9836 ± 0.01	0.9836 ± 0.06	0.9872 ± 0.05	0.9836 ± 0.06	0.9836 ± 0.06	0.9763 ± 0.01	**0.9873** ± 0.01
	MacroF1	0.9835 ± 0.01	0.9835 ± 0.07	**0.9871** ± 0.05	0.9835 ± 0.06	0.9835 ± 0.06	0.9761 ± 0.01	**0.9871** ± 0.01
R8CT	MicroF1	0.9905 ± 0.01	0.9905 ± 0.02	0.9920 ± 0.02	0.9905 ± 0.02	0.9905 ± 0.02	**0.9936** ± 0.01	0.9921 ± 0.01
	MacroF1	0.9904 ± 0.01	0.9904 ± 0.03	0.9920 ± 0.03	0.9904 ± 0.03	0.9904 ± 0.03	**0.9936** ± 0.01	0.9920 ± 0.01
SPHM	MicroF1	0.9718 ± 0.01	0.9718 ± 0.02	0.9707 ± 0.02	0.9718 ± 0.02	0.9718 ± 0.02	**0.9732** ± 0.01	0.9691 ± 0.01
	MacroF1	0.9536 ± 0.01	0.9536 ± 0.03	0.9518 ± 0.03	0.9536 ± 0.02	0.9536 ± 0.02	**0.9559** ± 0.01	0.9492 ± 0.01
KGMV	MicroF1	0.7013 ± 0.02	0.7010 ± 0.02	0.6985 ± 0.02	0.7019 ± 0.02	0.7013 ± 0.02	0.6968 ± 0.02	**0.7386** ± 0.01
	MacroF1	0.7010 ± 0.02	0.7008 ± 0.03	0.6983 ± 0.03	0.7017 ± 0.03	0.7010 ± 0.03	0.6966 ± 0.02	**0.7382** ± 0.01
Avg.	MicroF1	0.8330 ± 0.02	0.8323 ± 0.02	0.8275 ± 0.02	0.8329 ± 0.02	0.8331 ± 0.02	0.8284 ± 0.03	**0.8653** ± 0.02
	MacroF1	0.7697 ± 0.03	0.7692 ± 0.03	0.7681 ± 0.03	0.7696 ± 0.03	0.7698 ± 0.03	0.7704 ± 0.03	**0.8059** ± 0.02

in [31]. However, the difference of these works with our approach is that none of them involves an ensemble selection method which is the objective of our work. Though a dynamic selection is proposed in [31], the selection strategy is applied to the training samples whereas we apply the dynamic selection to the trees in the RFs.

Some in-depth reviews of various classifier selection methods are available in [5, 9, 35]. Our proposed method is identical to the dynamic ensemble selection methods where a set of classifiers are selected for each instance. We review the works related to dynamic ensemble selection methods below.

Table 4: Results of Wilcoxon tests for comparing KNORAU, KNORAE, DESP, METADES, and DESMI, with RF.

Comparison	R^+	R^-	Hypothesis	p-value
KNORAU vs. RF	204	261	Not rejected	0.5569
KNORAE vs. RF	181.5	283.5	Not rejected	0.2942
DESP vs. RF	185	280	Not rejected	0.3249
METADES vs. RF	276	189	Not rejected	0.3251
DESMI vs. RF	252.5	212.5	Not rejected	0.6808

Table 5: Results of Wilcoxon tests for comparing SARF with RF, KNORAU, KNORAE, DESP, METADES, and DESMI, respectively. The p-values indicating significant differences at $\alpha = 0.05$ (95% confidence) are marked with *.

Comparison	R^+	R^-	Hypothesis	p-value
SARF vs. RF	379	86	Rejected at 5%	0.0026*
SARF vs. KNORAU	380	85	Rejected at 5%	0.0024*
SARF vs. KNORAE	380	85	Rejected at 5%	0.0024*
SARF vs. DESP	380	85	Rejected at 5%	0.0024*
SARF vs. METADES	379	86	Rejected at 5%	0.0026*
SARF vs. DESMI	381	84	Rejected at 5%	0.0023*

Table 6: Examples of reliability measurement by the SARF method using the semantics.

True	Pred.	Semantic explanations	Reliability
(Dataset: RVLW)			
−1	−1	{screw, anybody, court, telephone, money}	1
1	−1	{deal, cares, skills, ethical, like, advanced}	0
1	−1	{court, fast, efficient, kindest, thankful, rescue}	0
1	1	{vast, knowlege, law, passion, treated}	1
(Dataset: TW14)			
1	1	{html5, demos, lots, great}	1
1	−1	{worth, liked, learning, jquery}	0
−1	−1	{hate, totally, got, church}	1
−1	1	{kinda, apple, man, dislike}	0

Given a test instance, existing dynamic ensemble selection methods evaluate the competency of the classifiers on a validation set or on k nearest neighbours of the given instance in the validation set. The number of selected classifiers depends on a predefined number or competency threshold. Ko et al. [18] proposed two approaches based on accuracies. In one approach, the classifiers with 100% accuracies on the k neighbours are selected and in the other approach, the classifiers which can correctly classify at least one of the neighbours are selected. In [16], each of the k neighbours assigned a weight according to its class label and the number of samples in each class. Each classifier is ranked according to the aggregated weights of the correctly classified neighbours. Finally, a predefined number of top ranked classifiers are selected for each test instance. A competency measure based on the performance of random classification is introduced in [38] where the classifiers having accuracies more than 50% are selected. However, these approaches focus on the performances of the classifiers on a validation set or the part of the validation which may not be appropriate in many situations. The performance of the ensemble depends on the selection of the validation set, the neighbours and the competency threshold. Instead of searching for the optimal values of those parameters, our method aims to leverage the semantic explanations to find a set of reliable predictions in an RF.

Some approaches use a meta-classifier to determine the competency of a classifier where a meta training set $\{\mathbf{X}_{meta}, \mathbf{Y}_{meta}\}$ is built from the outputs of the base classifiers [8, 10, 22, 25]. In [8, 10], the authors extract $\mathbf{X}_{meta}$ by evaluating the base classifiers using multiple competency measures and $\mathbf{Y}_{meta}$ is a set of binary indicators expressing the competency of the classifiers on the validation set. They have used a Naíve Bayes classifier as the meta-classifier to estimate the competency scores of the classifiers. The other approach [22, 25] uses the same features from the original training set as $\mathbf{X}_{meta}$ and converts the predictions of the base classifiers to a binary vector where the correct predictions are 1 and incorrect ones are 0 to obtain $\mathbf{Y}_{meta}$. Thus the classifier selection problem is reformulated as a multi-label classification problem which is solved using multi-label classifiers. The [8, 10] approaches still need to evaluate the base classifiers to obtain the meta-features and poses additional cost of training the meta-classifier. Though the [22, 25] approaches do not require additional feature extraction, the multi-label classifiers are computationally expensive.

The use of ensemble selection methods to improve the performance of RF classifiers are presented in [3, 14, 36, 41, 42]. A margin optimization based selection method is presented in [41] where the margin is defined as the difference between correct and incorrect votes in the RF. The trees are ranked according to the margin metrics and the lowest ranked trees are removed from the RF iteratively. In [14] Elghazel et al. proposed to rank the trees according to a fitness score calculated from accuracy and diversity. An optimal number of trees are selected from the ranked trees where the value of the optimum number is obtained by a hill-climbing search. Zhang and Wang [42] proposed to iteratively add diverse trees in the RF. The diversity is measured using a correlation function. Bernard et al. proposed two iterative selection methods using forward and backward selection strategies [3]. In each iteration, a tree is added (forward selection) or removed (backward selection). The accuracy of the RF is measured after each alteration and the RF achieving the highest gain in accuracy is retained. The iterative methods require to evaluate the performance of the RF after each iteration which is computationally infeasible for large problems. Another accuracy based competency measure is used in [36]. All the above methods belong to the static selection family i.e., the same set of classifiers is used for all future instances. Hence, the methods are not suitable for large, sparse and noisy features present in text classification.

The research in interpretable machine learning has shown that the semantics behind the predictions can be identified using the features and the predictions of the classifiers [26, 28]. The semantics reveal useful indication towards the reliability of the predictions and successfully utilized in ensemble classifiers in [17] where the consistency among the classifiers is compared using semantics to determine the most consistent decision. In this work, we incorporate semantics to determine the reliability of individual predictions. Thus, our method avoids the evaluation of the trees on a validation set, expensive computations involved in the re-evaluation of RFs to find the optimal set of trees or training meta-classifiers, and retains the diversity offered by the trees.

6 CONCLUSION

In this work, we present a new method to improve the performance of the prominent RF classifiers for text classification using a new dynamic ensemble selection (DES) method where the competency of the trees are determined by evaluating the features used by the trees to make predictions. Contrary to popular DES approaches that apply only the competent trees to an instance, we apply all the decision trees in an RF and combine the reliable predictions for generating the prediction of the RF. Moreover, the traditional approaches evaluate the trees on a small validation set to determine competency whereas the proposed Semantics Aware Random Forest method uses the semantic explanations of the predictions to determine their reliabilities. Our method computes relevance scores between a prediction and all target classes using TFIDF weights of the words in the semantic explanations. The reliability is determined by comparing the class receiving the maximum relevance score with the predicted class. The experimental results on 30 text datasets demonstrated that (i) existing DES methods fail to improve the performance of RF for text classification, but (ii) the proposed method achieves statistically significant improvement over the traditional RF as well as existing DES methods. In future, we aim to extend the concept of reliability measure to other ensemble algorithms in order to increase their performance.

ACKNOWLEDGMENTS

We acknowledge the University of South Australia, and D2DCRC, Cooperative Research Centres Programme for funding this research.

REFERENCES

[1] Dhammika Amaratunga, Javier Cabrera, and Yung-Seop Lee. 2008. Enriched random forests. *Bioinformatics* 24, 18 (2008), 2010–2014.
[2] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to Explain Individual Classification Decisions. *JMLR* 11 (2010).
[3] Simon Bernard, Laurent Heutte, and Sébastien Adam. 2008. On the selection of decision trees in random forests. *International Joint Conference on Neural Networks* (2008), 302–307.
[4] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.
[5] Alceu S Britto Jr, Robert Sabourin, and Luiz ES Oliveira. 2014. Dynamic selection of classifiers a comprehensive review. *Pattern Recognition* 47, 11 (2014).
[6] Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018. On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis. In *Proceedings of the 2018 EMNLP Workshop*. 40–46.
[7] Raphael Campos, Sérgio Canuto, Thiago Salles, Clebson CA de Sá, and Marcos André Gonçalves. 2017. Stacking bagged and boosted forests for effective automated classification. In *Proceedings of the 40th ACM SIGIR*. 105–114.
[8] Rafael MO Cruz, Robert Sabourin, and George DC Cavalcanti. 2017. META-DES. Oracle: Meta-learning and feature selection for dynamic ensemble selection. *Information fusion* 38 (2017), 84–103.
[9] Rafael MO Cruz, Robert Sabourin, and George DC Cavalcanti. 2018. Dynamic classifier selection: Recent advances and perspectives. *Information Fusion* 41 (2018), 195–216.
[10] Rafael MO Cruz, Robert Sabourin, George DC Cavalcanti, and Tsang Ing Ren. 2015. META-DES: a dynamic ensemble selection framework using meta-learning. *Pattern recognition* 48, 5 (2015), 1925–1935.
[11] Rafael M. O. Cruz, Luiz G. Hafemann, Robert Sabourin, and George D. C. Cavalcanti. 2018. DESlib: A Dynamic ensemble selection library in Python. *arXiv preprint arXiv:1802.04967* (2018).
[12] Janez Demšar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research* 7 (2006), 1–30.
[13] Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*. Springer, 1–15.
[14] Haytham Elghazel, Alex Aussem, and Florence Perraud. 2011. Trading-off diversity and accuracy for optimal ensemble tree selection in random forests. In *Ensembles in Machine Learning Applications*. Springer, 169–179.
[15] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. 2014. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research* 15, 1 (2014), 3133–3181.
[16] Salvador García, Zhong-Liang Zhang, Abdulrahman Altalhi, Saleh Alshomrani, and Francisco Herrera. 2018. Dynamic ensemble selection for multi-class imbalanced datasets. *Information Sciences* 445 (2018), 22–37.
[17] Md Zahidul Islam, Jixue Liu, Lin Liu, Jiuyong Li, and Wei Kang. 2019. Semantic Explanations in Ensemble Learning. In *Proceedings of the PAKDD 2019*. 29–41.
[18] Albert HR Ko, Robert Sabourin, and Alceu Souza Britto Jr. 2008. From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition* 41 (2008).
[19] L. I Kuncheva. 2014. *Combining Pattern Classifiers: Methods and Algorithms* (second edition ed.). John Wiley & Sons, Inc.
[20] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent Convolutional Neural Networks for Text Classification.. In *AAAI*, Vol. 333. 2267–2273.
[21] Christopher D Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT press.
[22] Anil Narassiguin, Haytham Elghazel, and Alex Aussem. 2017. Dynamic Ensemble Selection with Probabilistic Classifier Chains. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 169–186.
[23] Aytuğ Onan, Serdar Korukoğlu, and Hasan Bulut. 2016. Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications* 57 (2016), 232–247.
[24] Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.* 2, 1-2 (2008), 1–135.
[25] Fábio Pinto, Carlos Soares, and João Mendes-Moreira. 2016. CHADE: Metalearning with Classifier Chains for Dynamic Combination of Classifiers. In *Proceedings of the ECML PKDD*. 410–425.
[26] Gregory Plumb, Denali Molitor, and Ameet S Talwalkar. 2018. Model Agnostic Supervised Local Explanations. In *Advances in Neural Information Processing Systems*. 2520–2529.
[27] Robi Polikar. 2006. Ensemble Based Systems in Decision Making. *IEEE Circuits and Systems Magazine* 6, 3 (2006), 21–45.
[28] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *the 22nd ACM SIGKDD*. 1135–1144.
[29] Marko Robnik-Šikonja. 2004. Improving random forests. In *European conference on machine learning*. Springer, 359–370.
[30] Lior Rokach. 2009. Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Computational Statistics & Data Analysis* 53, 12 (2009), 4046–4072.
[31] Thiago Salles, Marcos Gonçalves, Victor Rodrigues, and Leonardo Rocha. 2018. Improving random forests by neighborhood projection for effective text classification. *Information Systems* 77 (2018), 1–21.
[32] Thiago Salles, Marcos Gonçalves, Victor Rodrigues, and Leonardo Rocha. 2015. BROOF: Exploiting Out-of-Bag Errors, Boosting and Random Forests for Effective Automated Classification. In *Proceedings of the 38th ACM SIGIR*. 353–362.
[33] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24, 5 (1988).
[34] Robert E Schapire and Yoram Singer. 2000. BoosTexter: A boosting-based system for text categorization. *Machine learning* 39, 2–3 (2000), 135–168.
[35] Grigorios Tsoumakas, Ioannis Partalas, and Ioannis Vlahavas. 2009. An ensemble pruning primer. In *Applications of supervised and unsupervised ensemble methods*.
[36] Alexey Tsymbal, Mykola Pechenizkiy, and Pádraig Cunningham. 2006. Dynamic integration with random forests. In *Proceedings of the ECML*. Springer, 801–808.
[37] Gang Wang, Jianshan Sun, Jian Ma, Kaiquan Xu, and Jibao Gu. 2014. Sentiment classification: The contribution of ensemble learning. *Decision support systems* 57 (2014), 77–93.
[38] Tomasz Woloszynski, Marek Kurzynski, Pawel Podsiadlo, and Gwidon W Stachowiak. 2012. A measure of competence based on random classification for dynamic ensemble selection. *Information Fusion* 13, 3 (2012), 207–213.
[39] Kevin Woods, W. Philip Kegelmeyer, and Kevin Bowyer. 1997. Combination of multiple classifiers using local accuracy estimates. *IEEE transactions on pattern analysis and machine intelligence* 19, 4 (1997), 405–410.
[40] Baoxun Xu, Xiufeng Guo, Yunming Ye, and Jiefeng Cheng. 2012. An Improved Random Forest Classifier for Text Categorization. *JCP* 7, 12 (2012), 2913–2920.
[41] Fan Yang, Wei-hang Lu, Lin-kai Luo, and Tao Li. 2012. Margin optimization based pruning for random forest. *Neurocomputing* 94 (2012), 54–63.
[42] Heping Zhang and Minghui Wang. 2009. Search for the smallest random forest. *Statistics and its Interface* 2, 3 (2009), 381.
[43] Zhong-Liang Zhang, Yu-Yu Chen, Jing Li, and Xing-Gang Luo. 2019. A distance-based weighting framework for boosting the performance of dynamic ensemble selection. *Information Processing & Management* 56, 4 (2019), 1300–1316.