Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

An effective ensemble deep learning framework for text classification

Ammar Mohammed [a,*], Rania Kora [a]

[a] Department of Computer Science, Faculty of Graduate Studies for Statistical Research, Cairo University, Egypt

ARTICLE INFO

Article history:
Received 15 July 2021
Revised 5 October 2021
Accepted 1 November 2021
Available online 10 November 2021

Keywords:
Deep learning
Ensemble learning
Text classification

ABSTRACT

Over the last decade Deep learning-based models surpasses classical machine learning models in a variety of text classification tasks. The primary challenge with text classification is determining the most appropriate deep learning classifier. Numerous research initiatives incorporated ensemble learning to boost the performance, minimize errors and avoid overfitting. However, the performance of the ensemble-methods is limited by the baseline classifiers and the fusion method. The current study makes the following contributions: First, it proposes a new meta-learning ensemble method that fuses baseline deep learning models using 2-tiers of meta-classifiers. Second, it conducts several experiments on six public benchmark datasets to evaluate the performance of the proposed ensemble. For each benchmark dataset, committees of different deep baseline classifiers are trained, and their best performance is compared with the performance of the proposed ensemble. Furthermore, the paper extends the results by comparing the performance of the proposed ensemble method to other state-of-the-art ensemble methods. The findings indicate that the proposed ensemble method significantly improve the classification accuracy of the baseline deep models. Furthermore, the proposed method outperforms the state-of-art ensemble methods. Finally, using the probability distributions for each class label of the deep baseline models improves the performance of the proposed ensemble method.

1. Introduction

According to a huge number of research in the field of machine learning, two approaches dominate the field right now: ensemble learning (Polikar, 2012; Sagi and Rokach, 2018; Rokach, 2019) and deep learning (Deng and Yu, 2014). In the former approach, the term "ensemble" refers to methods that weigh and integrate multiple base-learners in order to obtain a classifier that outperforms them all. The ensemble's central idea is to maximize the predictive accuracy by combining the strengths of multiple baseline classifiers. The concept of creating a predictive model that combines multiple models has been studied for a long time. The most popular ensemble methods includes bagging, Adaboost, Random forest and stacking (Sagi and Rokach, 2018; Rokach, 2019). Experimental research conducted by the machine learning community over the last few years has demonstrated that combining the outputs of various classifiers improves the performance of individual classifiers (Dietterich, 2000). For its impact on machine learning applications, ensemble has been applied in a wide range of applications including text classification (Tsai et al., 2011; Abellán and Mantas, 2014; Catal et al., 2015; Da Silva et al., 2014; Aburomman and Reaz, 2016).

Deep neural networks (DNN) (LeCun et al., 2015), on the other hand, have emerged as a powerful and dominant player in the field of machine learning. Throughout recent years, DNN has significantly improved the state-of-the-art in a variety of domains including natural language processing, text classifications, speech recognition, visual object recognition and object detection (Deng and Yu, 2014). The main idea of deep learning techniques is to learn and extract automatically complex features instead of manually crafted features as in classical machine learning (Alpaydin, 2020; Bengio, 2009). Deep learning methods employ a wide range of network architectures for problem solving, such as feeding forward neural networks (Bebis and Georgiopoulos, 1994), convolutional neural networks (Collobert and Weston, 2008), recurrent neural networks (Zaremba et al., 2014), and many others (Ain et al., 2017).

For the sake of enhancing predictive performance, there have already been several attempts in recent years to incorporate the ensemble and DNN methods. The most simple and straightforward

[*] Corresponding author.
E-mail address: ammar@cu.edu.eg (A. Mohammed).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2021.11.001
1319-1578/© 2021 The Authors. Published by Elsevier B.V. on behalf of King Saud University.

way to develop ensemble deep learning is to introduce deep learning directly to the conventional ensemble learning methods. The majority of these attempts revolve around creating a weighted average of deep learning models. This combination revealed that introducing the ensemble to deep learning models outperformed the performance of the individual DNN in the classification tasks (Ankit and Saleena, 2018; Heikal et al., 2018; Livieris et al., 2020; Vázquez-Romero and Gallardo-Antolín, 2020; Haghighi and Omranpour, 2021; Al-Omari et al., 2019).

In both machine and deep learning ensemble methods, the performance of the ensemble models is limited by how the existing ensemble methods work and the type of classification algorithms used as baseline models in the ensemble. As a result, numerous studies have proposed a variety of ways to improve ensemble methods (Araque et al., 2017; Srivastava et al., 2014), which in turn can be used to enhance the classification tasks. Generally, ensemble methods vary in terms of how to split the training data and the ways to combine several learners to produce the final prediction model. In Cruz et al. (2015), Džeroski and Ženko (2004), it was demonstrated that the meta-learning ensemble using the stacking method (Haghighi and Omranpour, 2021) is able to mimic all ensemble learning methods. Moreover, the authors in Kuncheva (2005), Brown et al. (2005) indicated that there should be some degree of heterogeneity of the classifiers to maximize the success of the ensemble learning methods. Ensemble methods can benefit from this heterogeneity of the classifiers by decreasing the variance-error without raising the bias-error (Tumer and Ghosh, 1996; Ali and Pazzani, 1996). The use of a diverse set of classifiers results in uncorrelated errors, which increases prediction performance. As a result, to achieve the intended predictive performance, the classifiers involved should be adequately diverse. Generally, creating diverse classifiers is based on the fact that a classifier is affected by the type of classification algorithm and the training data used to create it. As a result, by varying the training samples and/or classification algorithms, it is possible to generate classifiers with decisions that differ from one another. Another efficient way to increase the diversity in neural networks is to employ various network topologies or tune different hyperparameters on the same network.

To enhance the predictive accuracy of deep learning models in text classification, the primary objectives of this research are fourfold. First, we propose a new meta-learning ensemble algorithm to boost classification performance. The proposed algorithm depends on fusing the predictions of several groups of deep learning models using 2-tiers of shallow meta-classifiers. In the proposed method, we train a group of deep base-learners on a different partition of the training data to create the so-called committees. Then, in the first tier, a collection of shallow meta-classifiers is trained on the committees' predictions. The second tier produces the results by combining the predictions of the previous group of classifiers with a top-level shallow meta-classifier. In the proposed method, we increase the diversity in the ensemble using variations of the train data, the diversity of trained baseline deep classifiers, and the variation within the fusion of baseline deep learning models. Second, we conduct a wide range of experiments on six public benchmark datasets to study the performance of the proposed ensemble method for text classification tasks. For each benchmark dataset, committees of different deep baseline classifiers are trained on several partitions of the trained data, and their best performance is compared with the proposed ensemble method. Third, we study the impact of the text prediction types of the baseline deep models on the fusion of the proposed ensemble. In particular, we evaluate the proposed ensemble using the types of predictions; the class label predictions and probability distribution of the class label. Finally, we conduct further comparative experiments on the datasets using the most common ensemble techniques; weighted vote and stacking ensemble methods.

To that end, the paper's main contributions can be summarized as follows:

- We propose a new ensemble learning algorithm that combines baseline deep learning models using a combination of 2-tiers of shallow meta-learners to enhance the classification performance.
- We train several deep learning models based on different network architectures using public benchmark datasets for the task of text classification.
- We conduct several experiments to study the performance of the proposed ensemble method against individual deep learning models.
- We extend the experiments by comparing the performance of various commonly used ensemble methods to that of the proposed ensemble method.
- We investigate the effect of the prediction types within individual deep learning models on the proposed ensemble method.

The remainder of the paper is structured as follows. Section 2 briefly introduces the common methods of ensemble learning. Section 3 brings out some literature reviews of using ensemble learning in text classification using both machine and deep learning methods. Section 4 proposes the new ensemble learning scheme. Section 5 describes the environmental setup of the experiments, including the benchmark datasets. Section 6 analyses and discusses the experimental results. Finally, Section 7 draws conclusions and the findings of the paper.

2. Background on ensemble methods

Generally, ensemble methods may be classified as dependent or independent methods based on how each baseline classifier interacts with the others. In the dependent method, the result of one classifier has an impact on the formation of the next classifier. Boosting algorithms (Mayr et al., 2014) are the most well-known of examples of dependent methods. The independent method, on the other hand, builds each classifier separately from subsets of the dataset and then combines the results in some way. According to Chandra and Yao (2006), Liu and Yao (1999), classifiers should ideally be independent or negatively correlated to make effective ensemble. Bühlmann (2012), Random forest (ensemble of decision Tress) (Breiman, 2001) and Stacking (Haghighi and Omranpour, 2021) are the widely examples, among many others, of independent methods. The general framework of any ensemble learning in the independent method is to use an aggregation function G to combine a set k of baseline classifiers, $c_1, c_2, \ldots, c_k$, towards predicting a single output. Given a dataset of size n and features of dimension $m, D = \{(x_i, y_i)\}, 1 \leqslant i \leqslant n, x_i \in R^m$, the predication of the output based on this method is given by Eq. (1).

$$y_i = \phi(x_i) = G(c_1, c_2, \ldots, c_k) \tag{1}$$

where $y_i \in Z$ denotes classification. Building an ensemble model, given this general framework, entails deciding how to train baseline classifiers and a suitable process for aggregating the outputs of baseline classifiers. For their successful improvement on predictive accuracy and easily parallelized in training, several independent ensemble methods have been proposed over the last few years (Sagi and Rokach, 2018; Rokach, 2019; Polikar, 2012). Each ensemble method requires a proper fusion of several learners in order to generate the final prediction model. Generally, fusion techniques can be divided up further into averaging and meta-learning techniques. In this section, we give an overview of some of the most popular ensemble algorithms. In particular, three of the most effective and frequently used ensemble approaches will be discussed: averaging, bagging, and meta-learning using stacking.

2.1. Averaging ensemble

The simplest method for combining the predictions of multiple models is averaging method (Livieris et al., 2020). It is a widely used ensemble technique in which each model is trained separately, and the averaging technique linearly integrates all predictions of models by averaging them to produce the final prediction. This technique is simple to apply without needing extra training on huge numbers of individual predictions. Usually, voting is the standard way for averaging the prediction of the baseline classifiers. The final prediction results are usually determined by a majority vote on the predictions of many classifiers, which is referred as hard voting. The term hard voting can be defined mathematically by the Eq. (2), which specifies the statistical mode of the classifiers' predictions.

$$y_i = mode\{c_1, c_2, \ldots, c_k\} \tag{2}$$

While hard voting is easy to implement and produces better results than baseline classifiers, it, however, does not take into consideration the likelihood of the minor predicated classes. For example, if we have three classifiers with prediction probabilities of (0.49, 0.48, and 63), hard voting will yield (0,0,1) as corresponding predictions of the probabilities. In this case, the final hard vote prediction of the three classifiers votes becomes 0. However, when the averaging probabilities of classifiers is considered, the weighted average becomes 0.526, which predicates 1. Consequently, Soft voting considers the probabilities value of each classifier rather than the prediction labels of each classifier. Soft voting prediction can be formalized using the Eq. (3).

$$y = \underset{i}{argmax} \frac{1}{n} \sum_{j=1}^{n} w_{ij} \tag{3}$$

where w_{ij} is the probability of the i^{th} class label of the j^{th} classifier. A modified version of voting is to weight each classifier proportional to its accuracy performance on a validation set (Opitz et al., 1996).

2.2. Bagging ensemble

Bühlmann (2012) is one of the most commonly used techniques to improve the prediction results of individual models. Its fundamental key idea lies in creating more diverse predictive models through adjusting a stochastic distribution of the training datasets. In particular, on various bootstrap samples of the original training set, it applies the same learning algorithm, and the final result is achieved by averaging method. The bagging technique is extremely useful when dealing with large and high-dimensional datasets where identifying a single model that can show great performance is impossible.

2.3. Meta-learning ensemble

Meta-learning (Vilalta and Drissi, 2002; Prodromidis et al., 2000) is a method of learning from other classifiers. Unlike conventional learners, meta-training classifiers have two or more learning steps rather than just one. It starts by training baseline classifiers and then trains the meta-classifier, which fuses the predictions of baseline classifiers. During the prediction phase, the baseline classifiers output their classifications, and the meta-classifier performs the final classification. Stacking is a widely known method of meta-learning that employs a two-stage classification structure, namely the baseline classifiers and the meta-classifiers. The motivation behind this approach stems from the restriction of the simple average ensemble in which each model, regardless of how well it performed, is treated equally in the ensemble prediction. Stacking, on the other hand, creates a higher-level model for combining the predictions of individual models. In particular, the models that make up the ensemble are all individually trained with the same training set, usually called Level-0 training set. The collected predictions of all individual models are then used to create a Level-1 training set. It is worth noting that, to avoid overfitting the meta-learner, the data samples used to train the baseline classifiers must be excluded when training the meta-learner. As a result, the dataset should be partitioned into two distinct parts. The first part is used to build the base-level classifiers, while the second part is used to build the meta-dataset. In Chawla et al. (2004), it was recommended that randomly partitioning the datasets into disjoint partitions would resolve the memory constraint of conventional ensemble techniques while also forming an ensemble of accurate and diverse classifiers each constructed from a disjoint split.

It should be noted that the authors in Seewald (2002) conducted empirical proof that stacking performs worse in the multi-class datasets than on binary class datasets. The justification given was that as the number of classes in the dataset grows, the dimensionality of the meta-level data grows correspondingly. There are two further negatives of increasing dimensionality. First, it increases the meta-classifier's training time. Second, it increases the memory used during the training process.

3. Literature review

This section highlights related work on ensemble learning used in text classification by means of both machine and deep learning as baseline classifiers.

3.1. Ensemble classification in machine learning

Numerous research efforts have been proposed to use state-of-the-art ensemble learning methods to effectively generalize machine learning techniques in several domains, including text classification. They have shown that the existing ensemble learning methods have outperformed the baseline classifiers. For example, the research efforts of Fersini et al. (2014), Fersini et al. (2016), Perikos and Hatzilygeroudis (2016) applied ensemble using bagging on a group of baseline classifiers for sentiment classification. Similarly, the work in Chalothom and Ellman (2015) proposed two ensemble methods by majority voting and stacking based on Naive Bayes and SVM for sentiment analysis classification. Similarly, the work of Prusa et al. (2015) trained different baseline classifiers and used two popular ensemble techniques, bagging and boosting, for tweet sentiment analysis. In particular, they applied KNN, SVM, and logistic regression to the sentiment140 corpus (Go et al., 2009). In contrast, other researchers evaluated and compared the performance of their classification problems using different ensemble methods. For instance, the authors in Onan et al. (2016) trained several baseline classifiers like SVM, Naive Bayes, logistic regression, and linear discriminant analysis. They introduced an ensemble classification framework for text sentiment classification that employs a weighted voting scheme to identify the best classification algorithms used as base learners.

They compared their ensemble scheme with different ensemble methods, including AdaBoost, bagging, random subspace, and majority voting. Their imperial results on nine small datasets outperformed those obtained using conventional ensemble learning methods. Similarly, the authors in Wang et al. (2014) compared the performance of Bagging, Boosting, and Random Subspace for sentiment classification using five base-learners: decision tree, KNN, SVM, NB, and Maximum Entropy. Their imperial results on sentiment analysis using ten small datasets demonstrated that Random Subspace outperformed other ensemble learning methods. Other similar research is found in the literature. Table 1 summarizes some of those works that introduced ensemble learning methods for text classification tasks. It briefly summarizes the baseline classifiers, the ensemble technique, the fusion method, and the dataset used in the experiments.

Despite the success of ensemble methods in improving the accuracy of baseline machine learning models in text classification tasks, previous research efforts have been constrained by the size of the classification dataset and the predictive accuracy of the baseline classifiers.

3.2. Deep learning and ensemble classification

Deep learning is a promising alternative to traditional machine learning methods. It has shown excellent performance for larger datasets in different tasks of natural language processing, including text classification. For example, the work in Elnagar et al. (2020) applied several deep learning models for Arabic text categorization. They applied CNN, GRU, LSTM, Hierarchical attention network on two different datasets SANAD and NADIA (Einea et al., 2019). The authors proposed similar work in Abdi et al. (2019) in which they applied LSTM model to classify users' opinions expressed in reviews. The work of Alayba et al. (2017) applied CNN on Arabic corpus, which covers three health-related topics. Also, the authors in Alayba et al. (2018) proposed CNN_LSTM model using word-level for Arabic sentiment analysis. They used three different corpora of tweets with sizes of 1,732, 1,975, and 2,479, respectively, covering health services (Alayba et al., 2017), political topics (Abdulla et al., 2013), and general topics (Nabil et al., 2015). Another work was proposed in Mohammed and Kora (2019), in which the authors applied three deep learning models, namely CNN, LSTM, RCNN, to classify tweets on the Arabic tweet corpus (Kora and Mohammed, 2019). The results showed that the LSTM model achieved the highest performance. Likewise, the authors in Abdul-Mageed and Ungar (2017), Samy et al. (2018) proposed a GRU deep learning model to classify tweets. They used SemEval-2017 (Rosenthal et al., 2017), SemEval-2018 (Mohammad et al., 2018; Wang et al., 2012) datasets to construct and evaluate their model.

Because of its dominance in classification, deep learning classification methods are used in many research in a variety of languages and dialects for text classification. For example, the work in Smetanin and Komarov (2021) used a bidirectional encoder with transformers for the Russian language. Other researchers, for example, Kim (2014), Vizcarra et al. (2018), Mahmood et al. (2020), applied CNN models to English, Spanish, and Roman languages. The work in Huang et al. (2016), Hassan et al. (2016), Baly et al. (2017) applied LSTM models to Chinese, Bengali, and Arab Egyptian, and Emirati dialects.

The recent progress in deep learning has further introduced existing ensemble learning methods on deep learning classifiers. The ensemble methods have shown an effective role in improving results in many domains against the baseline models of deep learning. For instance, the researchers in Akhtyamova et al. (2017) suggested CNN based on the voting ensemble method for predicting drug safety on the user comments from healthcare discussion forums. Also, the work of Akhtar et al. (2018) proposed a multi-task ensemble framework for sentiment analysis, emotion analysis, and intensity prediction on three benchmark datasets. They applied an ensemble using voting and stacking on three deep learning models, namely LSTM, CNN, and GRU. Similarly, the work of Heikal et al. (2018) applied voting ensemble on the output of CNN and LSTM models using Arabic benchmark dataset (ASTD) (Nabil et al., 2015). Likewise, the efforts in Minaee et al. (2019) proposed voting ensemble by averaging the prediction of CNN and LSTM models on both IMDB reviews (Oghina et al., 2012) and SST2 (Müller et al., 2020) datasets. The work in Al-Omari et al. (2019) proposed a similar piece of work in which they applied a voting ensemble using BiLSTM model for the fake news and propa-

Table 1
Summary of the previous work using ensemble techniques in Machine Learning approach.

Studies	Baseline classifiers	Ensemble method	Classification task	Dataset
Prusa et al. (2015)	KNN, SVM, LR	Bagging, Boosting	Sentimental Analysis	Sentiment140
Wang et al. (2014)	SVM, KNN, DT, ME, NB	Bagging, Boosting	Sentimental Analysis	10 Movie reviews Datasets
Fersini et al. (2014)	ME, SVM, NB	Voting, Bagging	Sentimental Analysis	3 Product Reviews datasets
Chalothom and Ellman (2015)	NB, SVM	Voting, Stacking	Sentimental Analysis	SemEval 2013
Onan et al. (2016)	BLR,NB,LDA,LR, SVM	Stacking, AdaBoost, Bagging	Sentimental Analysis	9 Services Reviews Datasets
Fersini et al. (2016)	NB, DT, SVM	Voting	Sentimental Analysis	Movie Reviews,SemEval 2013
Perikos and Hatzilygeroudis (2016)	NB, ME	Voting	Sentimental Analysis	Twitter Dataset
Dedhia and Ramteke (2017)	NB, SVM and ME	Boosting	Sentimental Analysis	Twitter Dataset
Saleena (2018)	SVM, RF,NB,LR	Voting	Sentimental Analysis	4 Twitter Datasets
Oussous et al. (2018)	MNB, SVM,ME	Voting, Stacking	Sentimental Analysis	Moroccan Tweets Dataset
Pasupulety et al. (2019)	SVM,RF	Stacking	Sentimental Analysis	Indian Tweets Dataset
Alrehili and Albalawi (2019)	NB, SVM	Voting, Bagging, Boosting	Sentimental Analysis	Customer Reviews Datasets
Seker and Ocak (2019)	RF, LR, Linear R	Bagging	Text Classification	Services Reviews Datasets
Erdoğan and Namli (2019)	SVM	Voting, Stacking	Text Classification	Services Reviews Datasets
Cai et al. (2020)	SVM, LR	Voting	Text Classification	Services Reviews Dataset

Table 2
Summary of the previous work using ensemble techniques in Deep Learning approach.

Studies	Baseline classifiers	Ensemble method	Classification task	Dataset
Akhtyamova et al. (2017)	CNNs	Voting	Sentimental Analysis	Health Reviews Dataset
Araque et al. (2017)	LSTM,CNN,GRU	Voting, Stacking	Sentimental Analysis	Movie Reviews Datasets
Akhtar et al. (2018)	LSTM,CNN,GRU	Voting, Stacking	Sentimental Analysis	Twitter Datasets
Heikal et al. (2018)	CNN, LSTM	Voting	Sentimental Analysis	ASTD
Al-Omari et al. (2019)	Bi_LSTM	Voting	Text Classification	Fake News Dataset
Minaee et al. (2019)	CNN, LSTM	Voting	Sentimental Analysis	IMDB,SST2
Livieris et al. (2020)	LSTM,Bi_LSTM	Voting, Bagging, Stacking	Text Classification	Services Reviews Datasets
Haralabopoulos et al. (2020)	LSTM, GRU, CNN, RCNN,DNN	Voting, Stacking	Sentimental Analysis	Semeval2018, Toxic Comment
Mohammadi and Shaverizade (2021)	CNN,LSTM,GRU,Bi_LSTM	Stacking	Sentimental Analysis	SemEval2016

ganda detection corpus (NLP4IF 2019) (Aggarwal and Sadana, 2019). The researchers in Haralabopoulos et al. (2020) introduced a multi-label ensemble model using a voting and stacking ensemble method on both Semeval2018-Task (Mohammad et al., 2018) and Toxic comments (van Aken et al., 2018) datasets. The baseline deep learning models used in their proposed work include LSTM, GRU, RNN, RCNN, and a feed-forward deep neural network.

Several proposed works in the literature compared the performance of several ensemble learning models based on deep learning. For example, the work in Araque et al. (2017) applied several ensemble methods, including voting and meta-learning methods, for sentiment analysis classification on seven public datasets from the microblogging and movie reviews domains. The experimental results indicated that their proposed model outperformed the baseline deep learning models in terms of F1-Score. Similar studies are available in the literature. Table 2 summarizes some works that introduced ensemble learning methods in the deep learning approach for text classification tasks. It briefly summarizes the baseline classifiers, the ensemble technique, the fusion method of classifiers, and the datasets used in their experiments.

4. Proposed ensemble scheme

The main idea of the proposed ensemble method is to combine a group of diverse base-classifiers into different tiers. The first tier (Tier-0) trains a group of deep base-learners on a different partition of the training dataset to form the so-called committees. In the next tier (Tier-1), a group of shallow meta-classifiers is trained on the predictions of the committees of Tier-0. In the final tier (Tier-2), the predictions of the previous group of classifiers are combined with a top-level shallow meta-classifier to produce the results.

The term committee is used here to represent a group of different classifiers in the proposed method. The committee size refers to the number of different classifiers involved in the committee. The proposed ensemble method employs the concept of meta-classifiers or meta-learners to combine, in two levels, inter-committee classifiers and intra-committees predictions. Inter-committee fusion means combining the baseline classifiers inside the committee using a shallow meta-classifier. Intra-committees mean to combine the predictions of the committees using a top-level meta-classifier. Meta-classifiers or Tier-1 models are created through the use of learning algorithms on each committee. These models try to predict how the outputs of baseline classifiers in a committee (Tier-0 models) should be combined to generate meta-data. In addition, a top meta-learner or Tier-2 model is created by combining the outputs of the Tier-1 models with a top-level learning algorithm.

4.1. The proposed architecture

The over all learning architecture of the proposed ensemble is depicted in Fig. 1. The architecture makes use of three distinct tiers of classifiers in which (1) each committee is trained separately using unique training data and deep classifiers. (2) the outputs of the baseline-learners in each committee are combined using a meta-classifier, and (3) the outputs of all committees are combined using a top-level meta-classifier.

In essence, the proposed architecture is similar to the general architecture of a multi-layer perceptron, with Tier-0 representing the input layer, Tier-1 representing the hidden layer, and Tier-2 representing the output layer. The meta-classifiers in the hidden Tier (Tier-1) act as activation functions, taking input from Tier and producing output to Tier-2.

4.2. Formal description

Algorithm 1. Proposed Training Algorithm

```
procedure SAMPLING Input: Data
    $SampleData_i^{(0)} = Train_i^{(0)} \bigcup (Test_i^{(0)} = (X_i^{(0)}, Y_i^{(0)})), 1 \leqslant i \leqslant n$
Procedure LEARNING BASELINE CLASSIFIERS $Tier - 0$
    $DL = \{DL_1, DL_2, \ldots, DL_k\}$ a set of k Deep learning Algorithms
    for $eachTrain_i^{(0)}, 1 \leqslant i \leqslant n$
        for $eachDL_j \in DL, 1 \leqslant j \leqslant k$
            $M_{ij} \leftarrow fit(DL_j, Train_i^{(0)}), 1 \leqslant j \leqslant k$
        $G_i \leftarrow \{M_{i1}, M_{i2}, \ldots, M_{ik}\}, 1 \leqslant i \leqslant n$
    for $eachM_{ij} \in G_i, 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant k$
        $y_{ij}^{(1)} \leftarrow predictionofM_{ij}(X_i^{(0)}), 1 \leqslant j \leqslant k$
        $Data_i^{(1)} \leftarrow stack([y_{i1}^{(1)}, y_{i2,}^{(1)} \ldots, y_{ik}^{(1)}, Y_i^{(0)}]), 1 \leqslant i \leqslant n$
procedure META-LEARNING TIER$-1$
    $SplitData_i^{(1)} = Train_i^{(1)} \bigcup (Test_i^{(1)} = (X_i^{(1)}, Y_i^{(1)})), 1 \leqslant i \leqslant n$
    $F = \{f_1, f_2, \ldots, f_n\}$a set of $\mathbf{n}$ shallow classifiers
    for $eachTrain_i^{(1)}, 1 \leqslant i \leqslant n$
        $clf_j \leftarrow fit(f_i, Train_i^{(1)}), 1 \leqslant j \leqslant n$
    $clf \leftarrow \{clf_1, clf_2, \ldots, clf_n\}$
procedure TOP LEVEL META-LEARNING TIER$-2$
    for $eachTest_i^{(1)} = (X_i^{(1)}, Y_i^{(1)}), 1 \leqslant i \leqslant n$
        for $eachclf_j \in clf$
            $y_{ij}^{(2)} \leftarrow predictionofclf_j(X_i^{(1)})$
        $Data_i^{(2)} \leftarrow stack[y_{i1}^{(2)}, y_{i2}^{(2)}, \ldots, y_{in}^{(2)}, Y_i^{(1)}], 1 \leqslant i \leqslant n$
    $FinalMetaData = stack([Data_1^{(2)}, Data_2^{(2)}, \ldots, Data_n^{(2)}]^T)$
    $Top \leftarrow$ a shallow classifier
    $Model \leftarrow fit(Top, FinalMetaData)$
```

Algorithm 1 shows the formal procedure to train the proposed ensemble. Given a dataset $Data^{(0)}$, the procedure randomly generates n samples of equal size from the dataset. Each sample is further partitioned into training and test $Data_i^{(0)}$=(Train$_i^{(0)}$,Test$_i^{(0)}$). The $Tier - 0$ classifiers are generated by applying K learning algorithms to each training data Train$_i^{(0)}$, resulting in K classifiers in each committee. As a result, we have $n$committees $G_i, 1 \leqslant i \leqslant n$ each containing Kdiverse baseline classifiers $G_i = M_{i1}, M_{i2}, \ldots, M_{ik}$. For each Test$_i^{(0)} = (X^{(0)}, Y^{(0)})$, of the n samples, is used to create meta-data of the next Tier. Following the development of Tier-0 committees, each Test$_i^{(0)}$ are used to create Tier-1 instances. Tier-1 consists of n meta-datasets $Data_i^{(1)}, 1 \leqslant i \leqslant n$ generated from the predictions of committees on the test data of Tier-0.

Each generated $Data_i^{(1)}$ in Tier-1 data have $K + 1$ features: K features whose values are the predictions of the classifiers M_{ij} of the $1 \leqslant j \leqslant K$ in the committee $G^{(i)}$ for $Test_i$, and one additional feature represents the target class $Y^{(0)}$ of Test$_i^{(0)}$. Each $Data_i^{(1)}$ is partitioned into two disjoints subsets (Train$_i^{(1)}$,Test$_i^{(1)}$). Once the Tier-1 meta-data have been generated, n shallow meta-classifiers f_i are used to generate the Tier-1 models clf_i. These shallow meta-classifiers are trained on each Train$_i^{(1)}$ of the generated meta-data $Data_i^{(1)}$. Following the creation of Tier-1 models, Test$_i^{(1)} = (X^{(a)}, Y^{(1)})$ are used to create Top level meta-data or Tier-2 meta-data. The top level meta-data are generated in two steps. The first step generates $Data_i^{(1)}$ of $n + 1$ features whose values are the predictions of Tier-1 models on $X^{(1)}$ and target class $Y^{(1)}$. The second step superim-

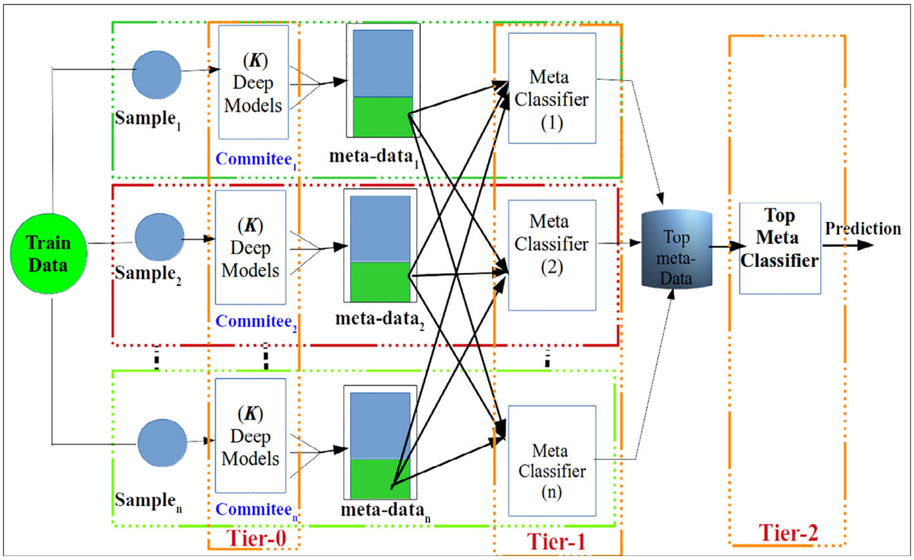

Fig. 1. The general procedure of the proposed ensemble.

poses all $Data_i^{(1)}$ to form the final meta data. A top level meta classifier is finally used to learn the final meta-Model of Tier-2.

4.3. Classification using proposed ensemble

The classification process used by the trained ensemble model on unseen data is illustrated in Fig. 2. When a new unseen sample, x, is fed to the proposed ensemble method, it is first distributed to all committees in the Tier-0. Next, the committees produce a two-dimensional tensor with K columns and n rows of predictions that are the input to Tier-1 models. Those models predict a vector of length n predictions. This vector will be fed to the Tier-2 top-level meta-model, which will generate the final prediction.

5. Experimental setup

This section details the conditions that will be used to evaluate the proposed ensemble scheme. In particular, this section demonstrates the benchmark datasets' setup environment, as well as the hyper-parameters of the baseline deep learning models, which are used to test our proposed ensemble approach.

5.1. Benchmark datasets

In order to evaluate the proposed ensemble method, we selected several datasets for conducting the experiments based on two different languages, Arabic and English:corpus on Arabic tweets (Kora and Mohammed, 2019), AJGT (Alomari et al., 2017), IMDB review (Oghina et al., 2012), SemEval 2017 Task 4 dataset (Júnior et al., 2017), COVID19 fake news detection (Patwa et al., 2020) and ArSarcasm Dataset (Farha and Magdy, 2021). Table 3 summarizes the details of each benchmark dataset.

5.2. Generating baseline deep learning models

To evaluate our proposed ensemble approach, we first need to build a group of deep classification models constituting the

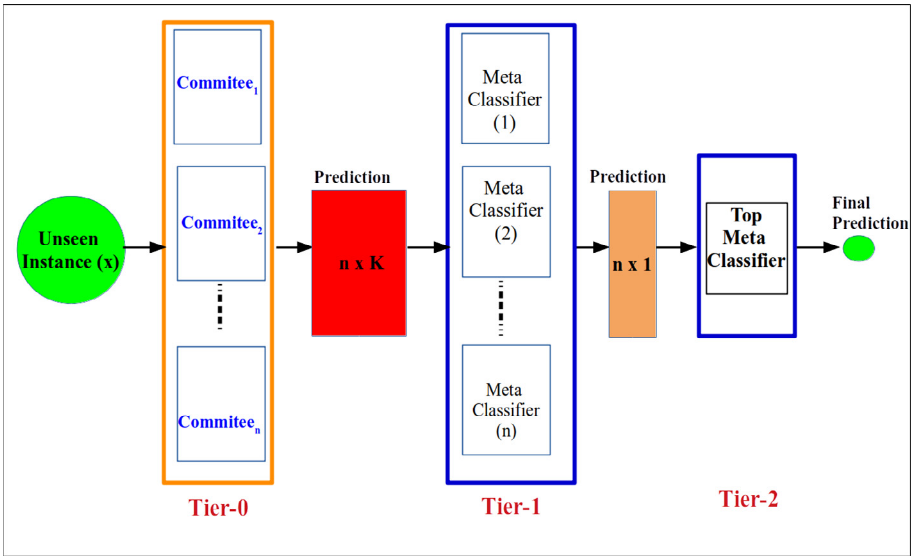

Fig. 2. Classification of unseen data.

Table 3
Details of benchmark datasets.

Benchmarks	Description	Classes	Data types	Polarity	Total
Arabic Corpus (Kora and Mohammed, 2019)	Egyptian dialects, MSA	2	Tweets	POS/NEG	40 k
AJGT (Alomari et al., 2017)	Jordanian dialects, MSA	2	Tweets	POS/NEG	1800
IMDB review (Maas et al., 2011)	English language	2	Review	POS/NEG	50 k
SemEval (Júnior et al., 2017)	English language	3	Tweets	POS/NEG/NEU	20634
COVID19-Fake (Patwa et al., 2020)	English language	2	Posts	Fake/Real	10700
ArSarcasm (Farha and Magdy, 2021)	five Arabic dialects	2	Tweets	Sar/Unsar	10567

baseline classifiers of the proposed ensemble approach. We construct a group of baseline deep models based on several architectures of networks for each benchmark dataset. Each group of baseline models is trained on different sizes of the benchmark data. The first baseline classifier used in our evaluation is the Long Short-term Memory (LSTM) (Mohammed and Kora, 2019). The Gated recurrent unit (GRU) (Cho et al., 2014) is another type of RNN that is used as a baseline classifier in our evaluation. In comparison to LSTM and traditional RNN, GRU is considered a simplified and more efficient LSTM model in terms of computational power (Habimana et al., 2020). Another common type of classifier is the Convolutional Neural Network Model (CNN) (Minaee et al., 2019). In recent years, the CNN model has proven to be more effective at a wide range of computer vision and natural language processing tasks. The fourth architecture used to train baseline models in the evaluation of the proposed framework is the loosely coupled network of both LSTM and CNN architecture, also known as the LSTM-CNN, which integrates both LSTM and CNN (Girshick et al., 2014) into a sequential form. Likewise, GRU-CNN is another loosely coupled network used in the proposed framework. Last but not least, a Bidirectional LSTM-CNN (BLSTM-CNN) (Minaee et al., 2019) is also used as a baseline classifier in the evaluation.

Each of the previous baseline architecture is trained on different hyper-parameters. Table 4 shows the hyper-parameters used during the training of the baseline models.

Before training the baseline deep classifiers, the textual data must be preprocessed into a suitable format for the network. Thus, one-hot encoding or word-embedding (Mikolov et al., 2013), is used as the initial layer before training the network.

5.3. The combiner shallow meta-classifiers

To combine the baseline deep models trained inside the committees, we use several shallow meta-classifiers as top surface meta-learners. In particular, we used a group of top successful algorithms for shallow learning that include Gradient Boosting (GB), Support Vector Machines (SVM), Naive Bayes (NB), Logistic Regression (LG), and Random Forest (RF). In general, any shallow classifier can be used in Tier-1 to combine the predictions of committees.

6. Experimental results and comparative analysis

The deep learning experiments for the baseline classifiers were implemented using Keras with TensorFlow as the back-end. To implement the meta-classifiers, we used the scikit-learn python library (Pedregosa et al., 2011), which includes a variety of machine learning algorithms.

To evaluate the impact of the proposed deep learning ensemble approach on the predictions, we conducted several experiments on the previous benchmark datasets and compared the ensemble's performance to the best individual baseline models. Additionally, we evaluate the proposed ensemble using hard and soft predictions generated from baseline models. We finally summarize all experimental results and compare the proposed method with other popular ensemble methods; stacking and voting.

We divided each benchmark dataset into training and validation test sets with a ratio of 80% and 20%, according to the Pareto principle (Harvey and Sotardi, 2018). To train baseline classifiers in the committees, it is necessary to partition the training data using a data partitioning method. Technically, ensemble methods include the following partitioning techniques (Dong and Han, 2005): random-size sampling, disjunct, random selection of data with replacement, and fold partitioning. In our experiments, we adopted the disjunct partitioning method, which randomly divides the training set into k equal-size partitions. A different partition of the benchmark dataset is used to train each committee. There are 5, 8, 8, 8, 8, 8, 8, and 8 partitions for the Arabic, AJGT, IMDB, SemEval, COVID19-Fake, and Arsaracsm corpora, respectively. We chose 5 partitions in the Arabic corpus rather than 8, because the authors of the original deep learning models on this dataset, Mohammed and Kora (2019), used five disjunct partitions to achieve the best results, and we would like to compare their results to those of the proposed approach. A committee of baseline classifiers is

Table 4
Summary of hyper-parameters of the baseline models.

Parameters	LSTM	GRU	CNN	LSTM-CNN	GRU-CNN	BLSTM-CNN
LSTM layer	1 or 2	–	–	1	–	–
LSTM size	256	–	–	128	–	–
GRU layer	–	1 or 2	–	–	1	–
GRU size	–	256	–	–	128	–
No. of filters	–	–	32	32	32	32
Filters size	–	–	16	16	16	16
BLSTM layer	–	–	–	–	–	1
BLSTM units	–	–	–	–	–	128
Vocab size	–	–	10000	10000	10000	10000
Batch size	(400,200,10)	(400,500,20)	(400,10)	500	500	500
No. of epochs	11	11	11	11	11	11
Learning rate	0.001	0.001	0.001	0.001	0.001	0.001
Dropout rate	0.7	0.7	0.7	0.7	0.7	0.7
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam

trained on each split. The size of the committees varies by dataset. In Arabic, AJGT, IMDB, SemEval, Fakenews covid19, Arsarcasm corpora, the sizes are 6, 3, 4, 4, 4, 4,4 respectively. A total of 182 baseline models have been trained in the conducted experiments.

Using 5-fold cross-validation on the predictions of baseline classifiers, Logistic regression is shown as the most frequent best combiner to fuse the committees in the Tier-1. However, for the final prediction, different shallow top-meta classifiers show better performance. In the following, we provide an empirical analysis of the proposed ensemble method and compare the performance with the best baseline models.

6.1. Results on the arabic corpus

According to the original experimental results on the Arabic corpus, Mohammed and Kora (2019), the best-achieved accuracy was 81.53%. Here, we train several individual models as baseline classifiers. Table 5 summarizes the accuracy of the trained models. The value in bold-font indicates the best-achieved accuracy per each data split. The last line of Table 5 averages the accuracy of each model on all splits. The average accuracy of all splits represents robust cross-validation of the baseline models. The evaluation of baseline models indicates that the average accuracy of LSTM outperforms all other baseline deep models with an average accuracy score of 89.54%.

We apply the proposed ensemble method to five committees, each with size six baseline models, and combine the baseline models' predictions (soft and hard) using several meta-classifiers in two categories of experiments. The first category considers the results of the hard predictions of the baseline models, whereas the second category considers the soft predictions. Table 6 summarizes the accuracy of the proposed ensemble using a different fusion of top meta-learners. In both categories of the experiments, the results of the proposed ensemble method significantly outperformed the best-performed model of the baseline models in all conducted shallow meta-learners. Moreover, the fusion of baseline models on soft predication further increased the accuracy. The results indicate that the ensemble with SVM as a meta-learner surpasses the results of other meta-classifiers with an accuracy of 92.6% and 93.2% in hard and soft predication. Compared to the best individual baseline deep model, the accuracy of the proposed ensemble was increased by 3.4% and 4.1% using hard and soft predictions, respectively.

6.2. Results on AJGT dataset

Essentially, the original experimental results on the AJGT dataset revealed that the best-achieved accuracy was 71.4% (Alomari et al., 2017). However, we performed an 8-fold validation split into three baseline classifiers: LSTM, GRU, and CNN. The experimental results reveal that the average accuracy of LSTM outperforms all other baseline deep models with an average accuracy score of 76.53% as shown in Table 7.

Table 8 summarizes the accuracy of the proposed ensemble using different fusion top meta-learners. The results indicate that the proposed ensemble, with the NB classifier, achieved the highest performance in hard prediction with an accuracy score of 80.6%. On the other hand, the ensemble method with the RF algorithm achieved the best performance in soft prediction by 88.8%. In comparison with the best baseline models, the accuracy of the proposed ensemble was increased by 5.3% and 16.03% using hard and soft predictions, respectively.

6.3. Results on IMDB dataset

In Oghina et al. (2012), the original experimental results on the IMDB showed that the best-obtained accuracy was 73.77%. In our experiments, we performed 8-fold splits on four baseline models. The four models are constructed using two different architectures of both GRU and LSTM. The experimental results indicate that the average accuracy of the *GRU*1 model outperforms all other baseline deep models with an average accuracy score of 78.41% as shown in Table 9. On the other hand, Table 10 summarizes the accuracy of the proposed ensemble method, and the results indicate that the

Table 7
The accuracy results of baseline deep classifiers in AJGT dataset.

Split	LSTM	GRU	CNN
1	80.7	**81.4**	74.59
2	75.7	**80.4**	80
3	**80**	78.9	56.4
4	62.9	**76.4**	66.6
5	**82.9**	59.6	54.19
6	**80.4**	**80.4**	54.6
7	67.9	**80**	60.8
8	**80**	60	61.19
Total AVG	**76.53**	74.63	63.79

Table 5
The accuracy results of baseline deep classifiers in arabic corpus dataset.

Split	LSTM	GRU	CNN	GRU_CNN	LSTM_CNN	BiLSTM
1	89.7	**89.9**	87.64	88.82	86.77	88.11
2	**89.8**	**89.8**	85.04	88.20	87.08	86.76
3	**89.8**	89.2	84.78	87.18	84.31	87.17
4	89.1	**89.5**	85.65	86.67	85.97	86.90
5	89.2	**89.3**	87.40	87.88	86.27	87.21
Total AVG	**89.54**	89.52	86.10	87.75	86.08	87.23

Table 6
The accuracy of the proposed ensemble using different meta-classifiers in arabic corpus.

	Proposed Ensemble				
	GB	SVM	NB	LG	RF
Hard prediction	92	**92.6**	91.6	91.9	91.9
Soft prediction	91.8	**93.2**	92.2	92.3	90

Table 8
The accuracy of the proposed ensemble method in AJGT dataset.

	Proposed Ensemble				
	GB	SVM	NB	LG	RF
Hard Prediction	77.8	68	**80.6**	77.8	78.6
Soft Prediction	88	69.3	80.6	79	**88.8**

Table 9
The accuracy results of baseline deep classifiers in IMDB dataset.

Split	LSTM1	GRU1	LSTM2	GRU2
1	78.5	**79.1**	73.6	56.9
2	**79.3**	78.6	70.2	56.6
3	77.5	**78.7**	67.4	54.3
4	79.2	**80.1**	60.1	54.9
5	**78**	77.3	67	54.6
6	77.6	**78.7**	69.5	55.8
7	77.1	**77.7**	68.4	55.1
8	**78**	77.1	59.7	56.4
Total AVG	78.15	**78.41**	66.98	55.57

Table 10
The accuracy of the proposed Ensemble in IMDB dataset.

	Proposed Ensemble				
	GB	SVM	NB	LG	RF
Hard Prediction	72.7	77.9	78	**78.8**	72.5
Soft Prediction	80.6	78.7	78.8	78.7	**86.6**

ensemble method achieved the best results using LG and RF meta-fusing classifiers in hard and soft prediction, respectively. Compared to the best individual baseline model, the accuracy of the proposed ensemble was increased by 0.49% and 10.44% using hard and soft predictions, respectively.

6.4. Results on SemEval dataset

According to Júnior et al. (2017), the original experimental results on SemEval indicated that the best accuracy obtained was 61.7%. However, in our experiments, we used 8-fold splits on four baseline models generated from two different GRU and LSTM architectures. Table 11 illustrates that the *LSTM*1 model outperforms all other baseline deep models with an average accuracy score of 76.75%. Also, Table 12 indicates that the ensemble with GB achieved the best accuracy in both hard and soft prediction with a score of 79.4% and 86%, respectively. Accordingly, the accuracy of the proposed ensemble was improved by 3.4% and 12.05% in hard and soft predictions, respectively.

6.5. Results on COVID19-Fake dataset

The original experimental results for COVID19-Fake, in Patwa et al. (2020), indicated that the highest level of accuracy obtained was 93.46%. However, our experiments on 8-fold split with four baseline models indicate that the average accuracy of *LSTM*2 outperforms all other baseline deep models with an average accuracy score of 93%. The summary of the full results is shown in the Table 13. In contrast, Table 14 summarizes the accuracy of the proposed ensemble method. The ensemble results indicate that SVM achieved the best accuracy in hard prediction with a score of 95.95%. However, both SVM and LG achieved the best accuracy in soft prediction with a score of 96.31%. Compared to the best individual model, the accuracy of the proposed ensemble was increased by 2.98% and 3.55% using hard and soft predictions, respectively.

6.6. Results on ArSarcasm dataset

Finally, in the ArSarcasm dataset, the experiments were conducted on 8-fold cross-validation splits using four baseline classi-

Table 11
The accuracy results of baseline deep classifiers in SemEval dataset.

Split	LSTM1	GRU1	LSTM2	GRU2
1	**76**	75.4	66.4	62.9
2	**75.7**	**75.7**	66	68.5
3	76.2	**77**	74.4	73.2
4	**77.3**	76.6	68.8	71.5
5	**77**	74.8	74.1	71.3
6	**75.7**	74.8	68.3	67.8
7	**77.9**	77.6	75.9	73.7
8	**78.2**	75.7	67.3	67.5
Total AVG	**76.75**	75.95	70.15	69.55

Table 12
The accuracy of the proposed ensemble in SemEval dataset.

	Proposed Ensemble				
	GB	SVM	NB	LG	RF
Hard Prediction	**79.4**	78.3	78.4	78.4	79.2
Soft Prediction	**86**	79.1	79	79.8	85.6

Table 13
The accuracy results of baseline deep classifiers in COVID19 fake news detection dataset.

Split	LSTM1	GRU1	LSTM2	GRU2
1	92.5	90.5	92.4	**92.7**
2	90.5	89.7	**92.4**	91.8
3	92.1	90.8	**93.9**	93.4
4	91.4	89.7	**93.4**	90.8
5	91.6	90.2	**94**	91.4
6	**93.2**	90.9	93	92.7
7	90.7	87.5	**92.5**	92.4
8	91.2	90	**92.4**	91.9
Total AVG	91.65	89.91	**93**	92.13

Table 14
The accuracy of the proposed ensemble in COVID19-Fake dataset.

	Proposed Ensemble				
	GB	SVM	NB	LG	RF
Hard Prediction	95.17	**95.95**	95.78	95.78	95.08
Soft Prediction	95.78	**96.31**	95.96	**96.31**	95.78

Table 15
The accuracy results of baseline deep classifiers in ArSarcasm dataset.

Split	LSTM1	GRU1	LSTM2	GRU2
1	84.7	**85.3**	83.7	83.7
2	83.2	**85.4**	83.4	83.4
3	84.1	**85.3**	83.4	83.4
4	85.3	**85.6**	84	83.7
5	**85.8**	85.6	84.1	84.1
6	85.4	**85.9**	84.2	84.3
7	84.8	**85**	83.6	83.6
8	84.5	**85.6**	84	84
Total AVG	84.72	**85.46**	83.8	83.77

Table 16
The accuracy of the proposed ensemble in ArSarcasm dataset.

	Proposed Ensemble				
	GB	SVM	NB	LG	RF
Hard Prediction	78.75	84.57	86.64	**87**	78.57
Soft Prediction	89.01	84.57	86.72	86.71	**89.94**

fiers created from two different architectures of GRU and LSTM. The experimental results, illustrated in Table 15, indicate that the average accuracy of the *GRU*1 model outperforms all other baseline deep models with an average accuracy score of 85.46%. By contrast, Table 16 describes the accuracy results of the proposed ensemble method, and the results indicate that the ensemble using the LG classifier achieved the best accuracy in hard prediction with a score of 87%. However, the ensemble using RF classifier achieved the best accuracy in soft prediction with a score of 89.94%. Compared to the best baseline models, the accuracy of the proposed ensemble was increased by 1.8% and 5.24% using hard and soft predictions, respectively.

6.7. Summary and discussion of results

The previously conducted experiments on all benchmark datasets indicate that the proposed ensemble method improves the accuracy of the baseline models without hyper-parameter tuning. Additionally, the results demonstrated that incorporating the probability distributions of the class prediction of the baseline models enhances the ensemble performance compared to incorporating the class label predictions.

Along with the previous experimental results, we compare the proposed ensemble method's performance to the performance of two well-known effective ensemble techniques on the same gener-

Table 17
Summary of accuracy and ranking of all methods.

Benchmarks	Baseline models	Voting	Stacking	Proposed ensemble

Arabic corpus	LSTM = 89.54(3)
	GRU = 89.52
	CNN = 86.10
	GRU_CNN = 87.75
	LSTM_CNN = 86.08
	BiLSTM = 87.23
		85.2(4)	92.9(2)	**93.2**(1)

AJGT	LSTM = 76.53(4)
	GRU = 74.63
	CNN = 63.79
		77(3)	82.6(2)	**88.8**(1)

IMDB review	LSTM1 = 78.15
	GRU1 = 78.41(4)
	LSTM2 = 66.98
	GRU2 = 55.57
		79.8(2)	79.3(3)	**86.6** (1)

SemEval	LSTM1 = 76.75(4)
	GRU1 = 75.95
	LSTM2 = 70.15
	GRU2 = 69.55
		78.1(2)	77.1(3)	**86**(1)

COVID19-Fake	LSTM1 = 91.65
	GRU1 = 89.91
	LSTM2 = 93(4)
	GRU2 = 92.13
		93.27(2)	93.24(3)	**96.31**(1)

ArSarcasm	LSTM1 = 84.72
	GRU1 = 85.46(2)
	LSTM2 = 83.8
	GRU2 = 83.77
		84.05(4)	84.15 (3)	**89.94** (1)

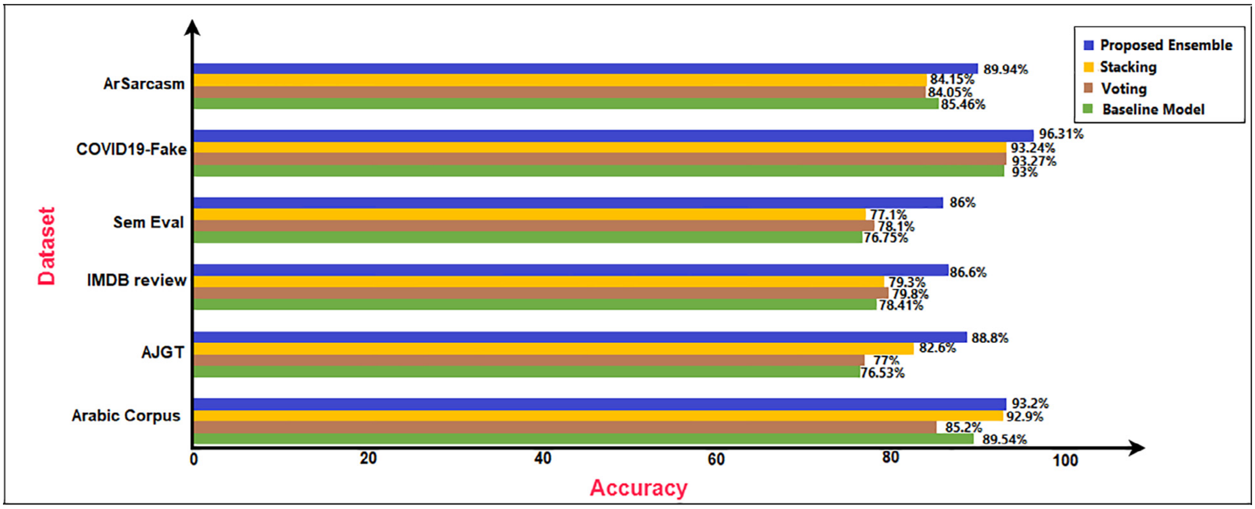

Fig. 3. The performance against both state-of-the-art ensemble methods and the best Baseline model.

ated baseline models: weighted average by voting and Stacking ensemble. The summary of the accuracy of the benchmarks data is shown in the Table 17. The table also summarizes the average accuracy results of baseline models. The baseline model with underlined font indicates the best accuracy among all baseline models. Moreover, the table includes the rank of each method in round brackets alongside the accuracy score. Fig. 3 illustrates the full performance on all benchmark datasets. The results reveal that the proposed ensemble method is ranked first in all datasets. In contrast, the voting ensemble is ranked second in IMDB, SemVal, and COVID19-Fake, third in AJGT, and fourth in Arabic Corpus and ArSarcasm. Similarly, the experimental results indicate that ensemble by stacking is ranked second in both the Arabic Corpus and the AJGT, third in IMDB, SemEval, COVID19-Fake and ArSarcasm.

7. Conclusion

The experimental research conducted by the machine learning community over the last few years has demonstrated that combining the outputs of various classifiers can reduce generalization errors and deal with the high variance of individual classifiers. Hence, the ensemble is an elegant solution for dealing with the high variance of individual classifiers while minimizing general errors. The idea of combining different models into an ensemble to create a predictive model has been explored for a long time. The key idea behind every ensemble strategy is based on the principle of weighing various models and combining their predictions to achieve better performance. In this paper, we introduced a new ensemble meta-learning strategy that uses two levels of meta-learners to fuse committees of baseline classifiers. The key idea of the proposed ensemble depends on increasing the diversity of classifiers for better performance. To test the efficiency of the proposed ensemble approaches, we ran many experiments on six public benchmark datasets for text classification. Groups of baseline classifiers are trained on each benchmark dataset, and their best model is compared to the proposed ensemble methods. In particular, we have trained 182 deep models and conducted a comparative study on five different shallow meta-classifiers to ensemble those models. Also, we have evaluated the proposed ensemble method's accuracy against that of other deep learning ensemble approaches widely used in the literature on the same trained baseline models. The results revealed that the suggested ensemble approach significantly enhanced the performance of the baseline classifiers on all benchmark datasets and outperformed stacking and weighted vote ensemble methods. Moreover, instead of using the predicted class label, the probability distributions for each class label of the baseline classifiers improve the proposed ensemble method's meta-learners' performance.

Declaration of Competing Interest

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

References

Polikar, R., 2012. Ensemble learning. In: Ensemble Machine Learning, Springer, pp. 1–34.

Sagi, O., Rokach, L., 2018. Ensemble learning: a survey. WIREs Data Min. Knowl. Discovery 8, (4) e1249.

Rokach, L., 2019. Ensemble learning: pattern classification using ensemble methods. World Sci. 85.

Deng, L., Yu, D., 2014. Deep learning: methods and applications. Found. Trends Signal Process. 7 (3–4), 197–387.

Dietterich, T.G., 2000. Ensemble methods in machine learning. In: International Workshop on Multiple Classifier Systems, Springer, pp. 1–15.

Tsai, C.-F., Lin, Y.-C., Yen, D.C., Chen, Y.-M., 2011. Predicting stock returns by classifier ensembles. Appl. Soft Comput. 11 (2), 2452–2459.

Abellán, J., Mantas, C.J., 2014. Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring. Expert Syst. Appl. 41 (8), 3825–3830.

Catal, C., Tufekci, S., Pirmit, E., Kocabag, G., 2015. On the use of ensemble of classifiers for accelerometer-based activity recognition. Appl. Soft Comput. 37, 1018–1022.

Da Silva, N.F., Hruschka, E.R., Hruschka Jr, E.R., 2014. Tweet sentiment analysis with classifier ensembles. Decis. Support Syst. 66, 170–179.

Aburomman, A.A., Reaz, M.B.I., 2016. A novel svm-knn-pso ensemble method for intrusion detection system. Appl. Soft Comput. 38, 360–372.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553) 436–444. [Online]. Available: URL:10.1038/nature14539.

Alpaydin, E., 2020. Introduction to Machine Learning. MIT press.

Bengio, Y., 2009. Learning Deep Architectures for AI. Now Publishers Inc.

Bebis, G., Georgiopoulos, M., 1994. Feed-forward neural networks. IEEE Potent. 13 (4), 27–31.

Collobert, R., Weston, J., 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning, pp. 160–167.

Zaremba, W., Sutskever, I., Vinyals, O., 2014. Recurrent neural network regularization, arXiv preprint arXiv:1409.2329.

Ain, Q.T., Ali, M., Riaz, A., Noureen, A., Kamran, M., Hayat, B., Rehman, A., 2017. Sentiment analysis using deep learning techniques: a review. Int. J. Adv. Comput. Sci. Appl. 8 (6), 424.

Ankit, Saleena, N., 2018. An ensemble classification system for twitter sentiment analysis. Procedia Comput. Sci. 132, 937–946. International Conference on Computational Intelligence and Data Science.

Heikal, M., Torki, M., El-Makky, N., 2018. Sentiment analysis of arabic tweets using deep learning. Procedia Comput. Sci. 142, 114–122.

Livieris, I.E., Iliadis, L., Pintelas, P., 2020. On ensemble techniques of weight-constrained neural networks. Evol. Syst., 1–13

Vázquez-Romero, A., Gallardo-Antolín, A., 2020. Automatic detection of depression in speech using ensemble convolutional neural networks. Entropy 22 (6), 688.

Haghighi, F., Omranpour, H., 2021. Stacking ensemble model of deep learning and its application to persian/arabic handwritten digits recognition. Knowl.-Based Syst. 220, 106940.

Al-Omari, H., Abdullah, M., AlTiti, O., Shaikh, S., 2019. Justdeep at nlp4if 2019 task 1: propaganda detection using ensemble deep learning models. In: Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda, pp. 113–118.

Araque, O., Corcuera-Platas, I., Sánchez-Rada, J.F., Iglesias, C.A., 2017. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. Expert Syst. Appl. 77, 236–246.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.

Cruz, R.M., Sabourin, R., Cavalcanti, G.D., Ren, T.I., 2015. Meta-des: a dynamic ensemble selection framework using meta-learning. Pattern Recogn. 48 (5), 1925–1935.

Džeroski, S., Ženko, B., 2004. Is combining classifiers with stacking better than selecting the best one? Mach. Learn. 54 (3), 255–273.

Kuncheva, L.I., 2005. Diversity in multiple classifier systems. Inf. Fusion 1 (6), 3–4.

Brown, G., Wyatt, J., Harris, R., Yao, X., 2005. Diversity creation methods: a survey and categorisation. Inf. Fusion 6 (1), 5–20.

Tumer, K., Ghosh, J., 1996. Error correlation and error reduction in ensemble classifiers. Connect. Sci. 8 (3–4), 385–404.

Ali, K.M., Pazzani, M.J., 1996. Error reduction through learning multiple descriptions. Mach. Learn. 24 (3), 173–202.

Mayr, A., Binder, H., Gefeller, O., Schmid, M., 2014. The evolution of boosting algorithms-from machine learning to statistical modelling, arXiv preprint arXiv:1403.1452.

Chandra, A., Yao, X., 2006. Evolving hybrid ensembles of learning machines for better generalisation. Neurocomputing 69 (7–9), 686–700.

Liu, Y., Yao, X., 1999. Ensemble learning via negative correlation. Neural Networks 12 (10), 1399–1404.

Bühlmann, P., 2012. Bagging, boosting and ensemble methods. In: Handbook of Computational Statistics, Springer, pp. 985–1022.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Opitz, D.W., Shavlik, J.W., et al., 1996. Generating accurate and diverse members of a neural-network ensemble. Adv. Neural Inf. Process. Syst. 535–541.

Vilalta, R., Drissi, Y., 2002. A perspective view and survey of meta-learning. Artif. Intell. Rev. 18 (2), 77–95.

Prodromidis, A., Chan, P., Stolfo, S., et al., 2000. Meta-learning in distributed data mining systems: Issues and approaches. Adv. Distrib. Parallel Knowl. Discovery 3, 81–114.

Chawla, N. v., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P., 2004. Learning ensembles from bites: A scalable and accurate approach. J. Mach. Learn. Res. 5, 421–451.

Seewald, A.K., 2002. How to make stacking better and faster while also taking care of an unknown weakness. In: Proceedings of the Nineteenth International Conference on Machine Learning, pp. 554–561.

Fersini, E., Messina, E., Pozzi, F.A., 2014. Sentiment analysis: Bayesian ensemble learning. Decis. Support Syst. 68, 26–38.

Fersini, E., Messina, E., Pozzi, F.A., 2016. Expressive signals in social media languages to improve polarity detection. Inf. Process. Manage. 52 (1), 20–35.

Perikos, I., Hatzilygeroudis, I., 2016. Recognizing emotions in text using ensemble of classifiers. Eng. Appl. Artif. Intell. 51, 191–201.

Chalothom, T., Ellman, J., 2015. Simple approaches of sentiment analysis via ensemble learning. In: Information Science and Applications, Springer, pp. 631–639.

Prusa, J., Khoshgoftaar, T.M., Dittman, D.J., 2015. Using ensemble learners to improve classifier performance on tweet sentiment data. In: 2015 IEEE International Conference on Information Reuse and Integration, IEEE, pp. 252–257.

Go, A., Bhayani, R., Huang, L., 2009. Twitter sentiment classification using distant supervision. CS224N project report, Stanford, vol. 1, no. 12, p. 2009.

Onan, A., Korukoğlu, S., Bulut, H., 2016. A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. Expert Syst. Appl. 62, 1–16.

Wang, G., Sun, J., Ma, J., Xu, K., Gu, J., 2014. Sentiment classification: the contribution of ensemble learning. Decis. Support Syst. 57, 77–93.

Dedhia, C., Ramteke, J., 2017. Ensemble model for twitter sentiment analysis. In: 2017 International Conference on Inventive Systems and Control (ICISC), IEEE, pp. 1–5.

Saleena, N. et al., 2018. An ensemble classification system for twitter sentiment analysis. Procedia Comput. Sci. 132, 937–946.

Oussous, A., Lahcen, A.A., Belfkih, S., 2018. Improving sentiment analysis of moroccan tweets using ensemble learning. In: International Conference on Big Data, Cloud and Applications, Springer, pp. 91–104.

Pasupulety, U., Anees, A.A., Anmol, S., Mohan, B.R., 2019. Predicting stock prices using ensemble learning and sentiment analysis. In: 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), IEEE, pp. 215–222.

Alrehili, A., Albalawi, K., 2019. Sentiment analysis of customer reviews using ensemble method. In: 2019 International Conference on Computer and Information Sciences (ICCIS), IEEE, pp. 1–6.

Seker, S.E., Ocak, I., 2019. Performance prediction of roadheaders using ensemble machine learning techniques. Neural Comput. Appl. 31 (4), 1103–1116.

Erdoğan, Z., Namli, E., 2019. A living environment prediction model using ensemble machine learning techniques based on quality of life index. J. Ambient Intell. Human. Comput., 1–17

Cai, R., Han, T., Liao, W., Huang, J., Li, D., Kumar, A., Ma, H., 2020. Prediction of surface chloride concentration of marine concrete using ensemble machine learning. Cem. Concr. Res. 136, 106164.

Elnagar, A., Al-Debsi, R., Einea, O., 2020. Arabic text classification using deep learning models. Inf. Process. Manage. 57, (1) 102121.

Einea, O., Elnagar, A., Al Debsi, R., 2019. Sanad: Single-label arabic news articles dataset for automatic text categorization. Data Brief 25, 104076.

Abdi, A., Shamsuddin, S.M., Hasan, S., Piran, J., 2019. Deep learning-based sentiment classification of evaluative text based on multi-feature fusion. Inf. Process. Manage. 56 (4), 1245–1259.

Alayba, A.M., Palade, V., England, M., Iqbal, R., 2017. Arabic language sentiment analysis on health services. In: 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), IEEE, pp. 114–118.

Alayba, A.M., Palade, V., England, M., Iqbal, R., 2018. A combined cnn and lstm model for arabic sentiment analysis. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction, Springer, pp. 179–191.

Abdulla, N.A., Ahmed, N.A., Shehab, M.A., Al-Ayyoub, M., 2013. Arabic sentiment analysis: Lexicon-based and corpus-based. In: 2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), IEEE, pp. 1–6.

Nabil, M., Aly, M., Atiya, A., 2015. Astd: arabic sentiment tweets dataset. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2515–2519.

Mohammed, A., Kora, R., 2019. Deep learning approaches for arabic sentiment analysis. Social Network Anal. Min. 9 (1), 52.

Kora, R., Mohammed, A., 2019. Corpus on Arabic Egyptian tweets. [Online]. Available: doi: 10.7910/DVN/LBXV9O.

Abdul-Mageed, M., Ungar, L., 2017. Emonet: fine-grained emotion detection with gated recurrent neural networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 718–728.

Samy, A.E., El-Beltagy, S.R., Hassanien, E., 2018. A context integrated model for multi-label emotion detection. Procedia Comput. Sci. 142, 61–71.

Rosenthal, S., Farra, N., Nakov, P., 2017. Semeval-2017 task 4: sentiment analysis in twitter. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 502–518.

Mohammad, S., Bravo-Marquez, F., Salameh, M., Kiritchenko, S., 2018. Semeval-2018 task 1: affect in tweets. In: Proceedings of the 12th International Workshop on Semantic Evaluation, pp. 1–17.

Wang, W., Chen, L., Thirunarayan, K., Sheth, A.P., 2012. Harnessing twitter big data for automatic emotion identification. In: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, IEEE, pp. 587–592.

Smetanin, S., Komarov, M., 2021. Deep transfer learning baselines for sentiment analysis in russian. Inf. Process. Manage. 58, (3) 102484.

Kim, Y., 2014. Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882.

Vizcarra, G., Mauricio, A., Mauricio, L., 2018. A deep learning approach for sentiment analysis in spanish tweets. In: International Conference on Artificial Neural Networks, Springer, pp. 622–629.

Mahmood, Z., Safder, I., Nawab, R.M.A., Bukhari, F., Nawaz, R., Alfakeeh, A.S., Aljohani, N.R., Hassan, S.-U., 2020. Deep sentiments in roman urdu text using recurrent convolutional neural network model. Inf Process. Manage. 57, (4) 102233.

Huang, M., Cao, Y., Dong, C., 2016. Modeling rich contexts for sentiment classification with lstm, arXiv preprint arXiv:1605.01478.

Hassan, A., Amin, M.R., Al Azad, A.K., Mohammed, N., 2016. Sentiment analysis on bangla and romanized bangla text using deep recurrent models. In: 2016 International Workshop on Computational Intelligence (IWCI), IEEE, pp. 51–56.

Baly, R., El-Khoury, G., Moukalled, R., Aoun, R., Hajj, H., Shaban, K.B., El-Hajj, W., 2017. Comparative evaluation of sentiment analysis methods across arabic dialects. Procedia Comput. Sci. 117, 266–273.

Akhtyamova, L., Ignatov, A., Cardiff, J., 2017. A large-scale cnn ensemble for medication safety analysis. In: International Conference on Applications of Natural Language to Information Systems. Springer, pp. 247–253.

Akhtar, M.S., Ghosal, D., Ekbal, A., Bhattacharyya, P., Kurohashi, S., 2018. A multi-task ensemble framework for emotion, sentiment and intensity prediction, arXiv preprint arXiv:1808.01216.

Minaee, S., Azimi, E., Abdolrashidi, A., 2019. Deep-sentiment: sentiment analysis using ensemble of cnn and bi-lstm models, arXiv preprint arXiv:1904.04206.

Haralabopoulos, G., Anagnostopoulos, I., McAuley, D., 2020. Ensemble deep learning for multilabel binary classification of user-generated content. Algorithms 13 (4), 83.

Mohammadi, A., Shaverizade, A., 2021. Ensemble deep learning for aspect-based sentiment analysis. Int. J. Nonlinear Anal. Appl. 12, 29–38.

Oghina, A., Breuss, M., Tsagkias, M., De Rijke, M., 2012. Predicting imdb movie ratings using social media. In: European conference on information retrieval, Springer, pp. 503–507.

Müller, M., Salathé, M., Kummervold, P.E., 2020. Covid-twitter-bert: a natural language processing model to analyse covid-19 content on twitter, arXiv preprint arXiv:2005.07503.

Aggarwal, K., Sadana, A., 2019. Nsit@ nlp4if-2019: Propaganda detection from news articles using transfer learning. In: Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda, pp. 143–147.

van Aken, B., Risch, J., Krestel, R., Löser, A., 2018. Challenges for toxic comment classification: an in-depth error analysis, arXiv preprint arXiv:1809.07572.

Alomari, K.M., ElSherif, H.M., Shaalan, K., 2017. Arabic tweets sentimental analysis using machine learning. In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, pp. 602–610.

Júnior, E.A.C., Marinho, V.Q., dos Santos, L.B., 2017. Nilc-usp at semeval-2017 task 4: a multi-view ensemble for twitter sentiment analysis. In: Proceedings of the 11th International Workshop on Semantic Evaluation, pp. 611–615.

Patwa, P., Sharma, S., PYKL, S., Guptha, V., Kumari, G., Akhtar, M.S., Ekbal, A., Das, A., Chakraborty, T., 2020. Fighting an infodemic: Covid-19 fake news dataset, arXiv preprint arXiv:2011.03327.

Farha, I.A., Magdy, W., 2021. A comparative study of effective approaches for arabic sentiment analysis. Inf. Process. Manage. 58, (2) 102438.

Maas, A., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C., 2011. Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 142–150.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078.

Habimana, O., Li, Y., Li, R., Gu, X., Yu, G., 2020. Sentiment analysis using deep learning approaches: an overview. Sci. China Inf. Sci. 63 (1), 1–36.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al., 2011. Scikit-learn: machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.

Harvey, H.B., Sotardi, S.T., 2018. The pareto principle. J. Am. College Radiol. 15 (6), 931.

Dong, Y.-S., Han, K.-S., 2005. Text classification based on data partitioning and parameter varying ensembles. In: Proceedings of the 2005 ACM Symposium on Applied Computing, pp. 1044–1048.