A novel ensemble approach for heterogeneous data with active learning

Mohamed Salama¹, Hatem Abdelkader¹ and Amira Abdelwahab¹,²

Abstract
At present, millions of internet users are contributing a huge amount of data. This data is extremely heterogeneous, and so, it is hard to analyze and derive information from this data that is considered an indispensable source for decision-makers. Due to this massive growth, the classification of data and analysis has become an important research subject. Extracting information from this data has become a necessity. As a result, it was necessary to process these enormous volumes of data to uncover hidden information and therefore improve data analysis and, in turn, classification accuracy. In this paper, firstly, we focus on developing an ensemble machine-learning model based on active learning which identifies the most effective feature extraction strategy for heterogeneous data analysis, and compare it with traditional machine-learning algorithms. Secondly, we evaluate the proposed model during the experiments; five heterogeneous datasets from various domains were used, such as a Health Care Reform dataset, Sander Frandsen dataset, Financial Phrase Bank dataset, SMS Spam Collection dataset, and Textbook sales dataset. According to the results, the novel approach for data analysis performed better than conventional methods. Finally, the study’s findings confirmed the validity of the suggested technique, meeting the study’s goal of using ensemble methods with active learning to raise the model’s overall accuracy for effectively classifying and analyzing heterogeneous data, reducing the time and money spent training the model, and delivering superior analysis performance as well as insights into other elements of extracting information from heterogeneous data.

Keywords
Heterogeneous data, text classification, active learning, machine learning, information extraction, ensemble method, natural language processing

Date received: 11 August 2021; accepted: 4 February 2022

Introduction
As Internet service penetration has increased, the number of Internet users has risen dramatically, which has resulted in an increase in the variety of material that can be accessed over the Internet. Text data extraction and categorization is a major study area in the modern world.

Data heterogeneity and high dimensionality are two of the biggest obstacles to solving this issue; however, feature selection may help. E-mail, online pages, e-documents, and SMS messaging are all forms of communication that rely heavily on text. It is impossible to dispute that supervised learning is the most common and effective method for tackling machine-learning difficulties.

¹Department of Information Systems, Faculty of Computers and Information, Menoufia University, Shibin Al Kawm, Egypt
²Department of Information Systems, College of Computer Science and Information Technology, King Faisal University, Al-Ahsa, Saudi Arabia

Corresponding author:
Mohamed Salama, Department of Information Systems, Faculty of Computers and Information, Menoufia University, Shibin Al Kawm 32511, Menoufia, Egypt.
Email: mohamed.salama@ci.menoufia.edu.eg

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
It is possible to classify text using supervised machine-learning techniques, which are based on previously labeled or annotated datasets from experts, to derive a general rule that can be applied to most datasets. Throughout the last decade, algorithmic advancements have been tremendous. However, model performance is largely dependent on data quality and volume, which costs time, money, and human resources, and thus, heterogeneous data analysis issues may occur.

It was possible to use machine-learning approaches to assess heterogeneous text data by enhancing the needed preprocessing processes that consider text data’s particular qualities. Preprocessing methods such as language-specific tokenization, normalization, and stemming are needed to analyze this data properly.

Because supervised learning requires human experts to manually annotate large amounts of data, it is expensive and time-consuming. The accuracy and stability of supervised modeling are ensured while the costs of human labeling operations are reduced by active learning.1

When dealing with large amounts of heterogeneous data, manual classification is impractical since it involves time, money, and resources. In recent years, the machine-learning technique in text classification research has gained widespread attention, allowing for the automated categorization of diverse data. That is, if some examples are selected at random and labeled by an experienced expert, then a supervised learning algorithm can use this training set to reproduce the labels for the whole collection of cases.

As a result of this work, we can use active learning techniques to obtain the necessary degree of accuracy without having to learn or label a whole dataset. This provides an opportunity to use active learning to improve the analysis of heterogeneous data-based machine-learning projects that were very challenging in terms of financial resources, as well as the technical and contextual background required for human annotators.

Traditional machine-learning algorithms need a large quantity of labeled data to be fed to the algorithm, as seen in Figure 1. Rather than classifying the whole dataset as a whole, in the case of active learning, we ask a human expert to name only a few of the elements. Following some measures, an algorithm successively selects the most informative examples and sends them to a labeling expert, who delivers the real labels for those questioned samples as shown in Figure 2.

This leads us to the design of the proposed approach, which enables us to manage the selection process so that we only need to label a small sample of text to achieve a specific degree of classification accuracy, rather than randomly selecting instances that constitute text data. This is the actual issue of active learning; Table 1 illustrates the difference between traditional machine learning (ML) and ML with active learning.

As the table indicates, the distinction between machine learning and machine learning with active learning lies in the query approach used to choose the training dataset for labeling from the unlabeled pool of data.

The learning ensemble is a successful branch of machine learning that incorporates a range of learning algorithms to boost prediction capacity and accuracy.

Feature extraction is necessary for ensemble classification to turn the text into numerical representations. As a result, categorization performance suffers greatly. Ensemble classifiers integrate numerous basic classifiers into a single strong classifier in the context of machine learning.2 Natural language processing techniques such as bag of words (BOW), term frequency-inverse document frequency (TF-IDF), and word2vec are used to preprocess the data.

Support vector machines (SVM), logistic regression (LR), Naive Bayes, and random forest (RF) were among the classifiers used in the suggested ensemble technique. An ensemble classifier applying TF-IDF outperformed the feature extraction BOW or word2vec.
This paper is structured according as follows: literature reviews are presented in the Literature Review section. The Active Learning Approach and Heterogeneous Text Classification section presents an active learning approach and heterogeneous text classification; the methodology and the design using active learning for heterogeneous text data are described in the Methodology section. The experiment and achieved results discuss in The Results and Experiments section. Finally, we summarize the conclusions and future work in The Results and Experiments section.

**Literature review**

Reducing the number of labeled samples is necessary for heterogeneous data analysis and classification due to the high costs of hand labeling, and several recent studies have focused on this topic. Using supervised ways to classify data is regarded as an impediment, and hence, unsupervised techniques like clustering are employed instead. Since heterogeneous data analysis is a hot topic among academics, several studies are produced each year.

As a result, the accuracy of the classification model is greatly impacted when random samples of data are manually labeled, commonly known as passive learning. The model’s accuracy is low because of the reliance on crowdsourcing to identify unlabeled data. Here are several literature reviews that are relevant to our work.

According to references 3 and 4, k-nearest neighbors (k-NNs), TF-IDF, Naive Bayes, decision tree, hybrid approaches, and support vector machines have all been presented as a broad method for automatic text classification. Managing a large number of features and working with unstructured text, dealing with missing information, and picking an effective machine-learning strategy for text classifier training were also discussed. In testing, the TF-IDF algorithm yielded proper findings. Both generated satisfactory outputs. The framework must be enhanced and improved to get better results.

According to references 5–7, statistical methods such as Naive Bayes, k-NNs, and SVMs are the most appropriate approaches for obtaining text classification outcomes. In this work, the classifier has classified the text with good results and a passive approach to learning through a random selection of samples before the labeling process by an annotator or expert system, a technique that has weaknesses.

Active learning methods also allow human experts to start with a small subset of data that will assist in creating an initial model. Machine learning selects the most useful data samples based on the modeling findings, thereby improving the precision. This method is continued until the required level of precision is reached.

Although in many fields, such as references 8 and 9, the highest levels of accuracy have been achieved in a majority of the classifiers, the ensemble classifiers are successful, especially in the text classification. In the literature, many extraction approaches have been developed, including BOW, TF-IDF, and word2vec, for the ensemble classification.

BOW is one of the most popular and frequently employed feature extraction methods in the classification process. BOW is mainly designed to represent the text in the N-value collection known as vector numbers. The words and the frequency counts of these vectors.10 Reference 11 offered a text classification ensemble approach, in which BOW was utilized for the extraction of features, and used the Naive Bayes classifier, linear discriminant analysis, and support vector machine; the results were promising.

TF-IDF is one of the important methods of a scaled-up BOW approach paradigm. Several earlier research studies have confirmed the preference of TF-IDF for certain specific classifiers over other feature extraction technologies.12 The influence of four feature techniques such as word2vec, TF-IDF, doc2vec, and the SVM; LR; NB; and RF counters was studied in reference 13. For example, the TF-IDF and counter-vector functionality offer the highest precision, support vector machine and logistic regression with TF-IDF or counter vectorizer have reached the maximum precision, and many earlier publications examine the TF-IDF for functional extraction for ensemble classifications.14,15

In the case of unbalanced class datasets, reference 16 advocated the usage of word2vec with ensemble classifiers. In addition, reference 17 employed word2vec for text categorization with an ensemble classifier that integrated models of neural networks.

According to references 18 and 19, the classifier achieved satisfactory results in classifying the text by Naive Bayes, k-nearest neighbors, and support vector machines, as well as by preprocessing, feature selection, semantic

---

**Table 1. Comparison between traditional ML and ML with active learning.**

|                      | Traditional ML                                      | ML with active learning                        |
|----------------------|-----------------------------------------------------|------------------------------------------------|
| Data sample          | Randomly select data from the unlabeled pool to train and test | Select data by (query strategy) for a train set |
| Labeling             | Assign labels to selected data using a human expert  |                                                 |
| Training             | Train the ML model                                  |                                                 |
| Evaluation           | Check model performance                             |                                                 |

Note: ML, machine learning.
techniques, and machine-learning techniques such as NB, decision tree, and SVM classification.

Active learning approach and heterogeneous text classification

Active learning receives great interest from researchers who reduce the amount of time, cost, and effort for labeling data in many applications by increasing the efficiency of supervised machine-learning algorithms during the training phase using minimal human involvement; by selecting the smallest possible amount of training data, this ensures strong classification performance in the test phase.\(^{20}\)

Active learning is typically used with supervised machine-learning methods to further minimize the effort of labeling data and, as a result, reduce the amount of human work required to discover relevant information and make decisions.

Active learning is a reactive, repeated method that produces high-performance classification using small labeled data. Whereas passive learning works with a fixed collection of labeled samples provided to the learning algorithm that is used to build a model, active learning paradigms demand that the learning algorithm select the results of learning by selecting the most informative examples.\(^{21}\) Active learning is commonly utilized when large volumes of unlabeled data are accessible.

The active learning approach is based on the concept that a learning algorithm chooses the data samples to be manually labeled instead of relying on random sampling or predefined selection criteria by a human expert.\(^{22}\) There are different types of active learning such as pool-based and stream-based active learning.

Pool-based active learning is the most common form of active learning, the learning and labeling process is done in batches extracted from the unlabeled data pool.\(^{20}\) As in Figure 3, the learning algorithm is trained using the labeled samples after every batch has been labeled and a selection of new samples to train, which may enhance learning performance; the algorithm can be written as shown in Figure 4. But the data is given to the algorithm as a stream with all training samples in stream-based active learning. Each case is independently provided to the algorithm to be considered. This sample has to be labeled or not immediately by the algorithm. Some of the instances of training in this pool are labeled by the human expert, and right before the next example is displayed, the algorithm receives the label.

The goodness of the data utilized has a considerable impact on the results of the classification task. The data labeling process is a key obstacle to classification, especially with the current availability of data resources. Hence, data heterogeneity became a new challenge for data, the choice of the most useful data relies on the measure of uncertainty used.\(^{23}\) The active learning algorithm in pool-based sampling selects instances to contribute to the increasing training set, the most useful samples of which are those that are least certain.

Query strategies that involve uncertainty sampling are among the most common. In the sampling of uncertainty,\(^{24}\) the most uncertain sample will be those that have the least difference with the highest probability between the two most trusted forecasts.

Uncertainty samplings are utilized that choose or query some of our unlabeled data pools for the least certain samples, the labeled data samples are then classified using the ensemble classifier, and more unlabeled samples are constantly tagged until the accuracy of the model is increased to an acceptable level; after each sample batch is
processed using learning methods on the complete training dataset, a comparison is continually carried out.

The active learning approach is illustrated in Figure 5 as follows: Using the query technique generated by the active learning model, a human expert requests unlabeled instances, which the active learning model selects and presents to the human expert. The expert labels these instances and returns them to the active learning model. After each update, the active learning model is retrained. This procedure is repeated until a stopping condition is satisfied, such as a maximum number of iterations or a minimum threshold of classification accuracy change.

The objective of active learning methods consists of creating a model using the fewest feasible labeled examples to minimize the expert engagement without affecting the classification performance; this will produce a reduction in the cost and time of sample labeling. The proposed ensemble model based on active learning is so intelligent that it can select whatever data the model wants to train, but it still is a good idea to train the model on samples that considerably affect its performance.

Methodology

This section covers the general approach utilized in this research that includes the proposed approach, collection and preprocessing of the datasets, feature extraction, ensemble methods for classification, and evaluations.

In our novel pool-based approach, the labeling process is done in batches selected from the unlabeled data pool; after labeling each batch, the algorithm is trained using these batches; and this process is repeated with a set of new samples until the learning performance improves.

We use uncertainty sampling that selects the least certain samples from a pool of unlabeled data. Then, these samples are processed by a human expert, and this process is repeated till all batches are labeled. After each batch, the accuracy is checked until the desired degree of accuracy is reached. The labeled samples of data are then classified. This process is repeated to label more of the unlabeled samples till the accuracy of the model is improved to an acceptable level.

Data gathering

The gathered datasets comprise diverse sizes and sources; five heterogeneous datasets from various domains were used, such as a Health Care Reform dataset, Sander Frandsen dataset, Financial Phrase Bank dataset, SMS Spam Collection dataset, and Textbook sales dataset. In this section, we provide a brief description of each dataset.

In this paper, we used many datasets. The first one was a Health Care Reform (HCR) dataset of crawled tweets containing the hashtag. This dataset contains a collection of more than 1050 tweets. The second dataset is the Sander Frandsen dataset from the Crowdflower’s Data for Everyone library. This dataset contains a collection of more than 850 tweets. The third dataset, Financial Phrase Bank, contains sentiments for financial news headlines. This dataset contains a collection of more than 870 news headlines’ data. The fourth dataset, SMS text messages from the UCI datasets, contains a collection of more than five thousand SMS text messages to detect spam messages, and the final dataset is a textbook sales dataset for a Dataquest Guided project. This dataset contains a collection of more than 2150 reviews’ data.

Preprocessing

The preprocessing step includes techniques to remove redundant and insignificant data to minimize the volume of the feature. This step improves the machine-learning algorithm’s accuracy.

Preprocessing is the main phase of big data analytics and a highly essential data preparation procedure. Data cleaning is an important stage, and it is necessary to train all ML models. To improve the quality of data analysis, preprocessing and integration of data are carried out. These methods decrease the uncertainty of information and offer superior analytical criteria, enhance the understanding of the information complexity, and make the data analysis process robust and efficient; the findings differ substantially without cleaning and preprocessing. Five popular text data preparation steps, including data cleaning, stop word removal, lowercase conversation, tokenization, and stemming, are considered in this work. A brief description of these steps is as follows:

- Data cleaning: Unwanted observations such as dots, digits, short phrases, and certain characters have been deleted to improve the quality of the data.
- Stop word removal: Stop words are high-frequency words that are not dependent on a certain subject such as prepositions. In studies of text analysis, stop words
are often considered redundant and useless as they have high frequencies of recurrence but do not give valuable meaning.

- Conversion to lowercase: As both lowercase and uppercase word forms have no clear meaning, all signed letters are converted into smaller letters.
- Tokenization: It is the process of segmenting text into meaningful segments, such as words or phrases.
- Stemming: It is the operation of getting all words’ root shapes. In the processing of natural languages, stemming is necessary to address the many forms of derived words as a single stem.

**Features extraction**

This step involves analysis of the text to identify certain traits which are described as a vector that the classification can comprehend. Text representation is an essential and powerful phase in natural language processing.

BOW is the most common, widespread technique used with ensemble learning in previous studies, including various extraction strategies. The word frequency of BOW and TF-IDF will be treated as discreet symbols. The word order and meaning of the words are not taken into account. The word2vec overcomes this issue by collecting additional word similarity data. Therefore, the experimental variables are BOW, TF-IDF, and word2vec. The BOW and TF-IDF methods were utilized by using the scikit-learn library, and the Genism library was used to import word2vec models using the Python programming language. Training for the ensemble classification was conducted, and the test set was used for assessment purposes.

**Ensemble classification**

An ensemble classifier was built and assessed using all created features. To satisfy the main objective of this study, we need to define the basic classifiers and the technique of combining such classifiers for the production of an ensemble model. Among the various combination techniques, voting is a way for generating ensemble learning that is efficient, simple, and often utilized. Each of the basic classifier sets must make the simplest vote, called majority voting, an equal contribution, and a single vote.

The most frequently voted are the ultimate result of the ensemble model classification. Therefore, the procedure of majority voting was applied as a combination. The SVM, logistic regression, Naive Bayes, and the random forest was the basis of the ensemble model as it is widely known and popular in the classification of heterogeneous text data. The ensemble classifier was trained using each feature’s training subset. The exactness metric was regarded as an evaluation measure in each feature’s testing sub-sets for evaluation purposes.

**Performance evaluation measures**

The confusion matrix is the most common method used to summarize the performance of a classification method. The ratio of properly categorized instances \((TP + TN)/(TP + TN + FP + FN)\) can be described as several actions based on outcomes of a confusion matrix such as the accuracy of correctly categorized cases. Where TP, FN, FP, and TN denote the numeral of true positives, false negatives, false positives, and true negatives, and specific measurements are used to verify each model and assess its accuracy, recall, and F1 score for per classifier.

Accuracy (1), precision (2), recall (3), and F1-score (4) are our evaluation measures

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1} \\
\text{Precision} = \frac{TP}{TP + FP} \tag{2} \\
\text{Recall} = \frac{TP}{TP + FN} \tag{3} \\
\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}
\]

\(TP, FN, FP,\) and TN are determined by the numbers of real positive instances, the numbers of false-negative cases, and the number of true negative cases.

a. **Accuracy** is the most intuitive measure of performance, and the percentage of correct classification of items for all objects assessed is determined.

b. **Recall:** Correct positive ones, distinguished from true positives and false negatives, are defined.

c. **Precision:** Positive accuracy split between true and false positives.

d. **F1-Score (F1):** This involves calculating the exactness of the model, utilizing precision and memory; therefore, it is a perfect step to analyze the text.

**The results and experiments**

This study proposes that active learning be utilized to construct a new ensemble technique to improve the analysis of diverse data. This method provides a fast and effective way to deal with unlabeled data that is deemed heterogeneous, reducing the processing and labeling effort by cleverly picking samples without affecting the accuracy of the classification. This problem may be mitigated by using a large number of unlabeled samples, and then picking a batch that is suitable for ML classification based on probabilities and query methods, which are both time-consuming and error-prone tasks during the training stage. This study used the provided data and tried the recommended method for analyzing and understanding data. All of the previously specified datasets were used. The programming was implemented in the Anaconda environment using a Jupyter
notebook, which is a versatile Python language development environment. The experiments were carried out on a device with the following specifications: Windows 10 platform, 8 GB RAM, core i5 processor. We have evaluated the ensemble model and other traditional classification techniques using BOW, TF-IDF, and word2vec features, as demonstrated in Table 2.

The results show how effective the TF-IDF feature is for the ensemble model in all datasets. Also, the BOW feature is the least efficient choice in all datasets for the ensemble model.

The experiment discussion and evaluation

In this section, we present our experimental results achieved; evaluate performance based on results from various traditional ML classifiers and the proposed ensemble model using TF-IDF achieved the highest accuracy; and run classifiers for all datasets and compare performance based on recall, F1-score, and precision.

The experimental results of the selected dataset have been validated using this measurement of precision, recall, and F1-score. The findings have shown that the proposed approach is capable of improving performance considerably.

Table 3 displays the traditional ML classifier’s precision results, and compares the output of each classifier with the results of the ensemble model used on the HCR dataset with TF-IDF during model development.

As a result of this evaluation, it has been determined that the ensemble model classifier is the most accurate of all ML classifiers. The final experiment results depict that the ensemble classifier achieved the best performance among all the other classifiers with an accuracy of 85%, precision of 0.84%, recall of 85%, and F-measure of 85%, using the feature extraction techniques (TF-IDF). In contrast, the logistic regression classifier achieves the lowest overall accuracy of 78%, precision of 79%, recall of 79%, and F-measure of 79% among all the classifiers. Thus, this result is considered a good result, according to our sample size of data.

Table 2. Classifiers’ result summary among different feature extraction techniques with traditional ML classifier and proposed ensemble model.

| Dataset name                  | Feature extraction technique | Traditional ML classifier accuracy | Ensemble model accuracy |
|------------------------------|-----------------------------|-----------------------------------|-------------------------|
|                              |                             | LR  | RF  | SVM | NB  |                  |
| Health Care Reform            | BOW                         | 0.79| 0.77| 0.75| 0.78| 0.79              |
|                              | TF-IDF                      | 0.78| 0.80| 0.79| 0.80| 0.85              |
|                              | Word2vec                    | 0.80| 0.79| 0.79| 0.80| 0.80              |
| Sander Frandsen dataset       | BOW                         | 0.80| 0.75| 0.77| 0.79| 0.79              |
|                              | TF-IDF                      | 0.79| 0.79| 0.80| 0.78| 0.84              |
|                              | Word2vec                    | 0.79| 0.78| 0.79| 0.78| 0.80              |
| Financial Phrase Bank         | BOW                         | 0.80| 0.75| 0.77| 0.79| 0.79              |
|                              | TF-IDF                      | 0.80| 0.79| 0.81| 0.80| 0.86              |
|                              | Word2vec                    | 0.80| 0.79| 0.77| 0.78| 0.81              |
| SMS text messages             | BOW                         | 0.81| 0.79| 0.77| 0.79| 0.81              |
|                              | TF-IDF                      | 0.78| 0.79| 0.80| 0.81| 0.85              |
|                              | Word2vec                    | 0.80| 0.79| 0.77| 0.79| 0.82              |
| Textbook dataset              | BOW                         | 0.80| 0.75| 0.77| 0.79| 0.80              |
|                              | TF-IDF                      | 0.80| 0.81| 0.79| 0.80| 0.86              |
|                              | Word2vec                    | 0.80| 0.79| 0.77| 0.78| 0.81              |

Note: ML, machine learning; NB Naive Bayes; k-NN, k-nearest neighbor; RF, random forest; RNN, recurrent neural network; SVM, support vector machine; LR, logistic regression; BOW, bag of words; W2V, word2Vec; TF-IDF, term frequency-inverse document frequency.
Table 3. Comparative analysis of the traditional classifiers’ and the ensemble model’s results: HCR dataset with TF-IDF.

| Technique          | Accuracy | Precision | Recall | F1-score |
|--------------------|----------|-----------|--------|----------|
| Logistic regression| 0.78     | 0.79      | 0.79   | 0.79     |
| Random forest      | 0.80     | 0.79      | 0.79   | 0.79     |
| SVM                | 0.79     | 0.79      | 0.80   | 0.79     |
| Naive Bayes        | 0.80     | 0.78      | 0.79   | 0.78     |
| Ensemble model     | **0.85** | **0.84**  | **0.85** | **0.85** |

Table 4. Comparative analysis of the classifiers’ results summary: SF dataset with TF-IDF.

| Technique          | Accuracy | Precision | Recall | F1-score |
|--------------------|----------|-----------|--------|----------|
| Logistic regression| 0.79     | 0.79      | 0.80   | 0.79     |
| Random forest      | 0.79     | 0.78      | 0.79   | 0.79     |
| SVM                | 0.80     | 0.79      | 0.80   | 0.79     |
| Naive Bayes        | 0.78     | 0.78      | 0.80   | 0.78     |
| Ensemble model     | **0.84** | **0.85**  | **0.84** | **0.86** |

Table 5. Comparative analysis of the traditional classifiers’ and the ensemble model’s results: FPB dataset with TF-IDF.

| Technique          | Accuracy | Precision | Recall | F1-score |
|--------------------|----------|-----------|--------|----------|
| Logistic regression| 0.80     | 0.79      | 0.80   | 0.79     |
| Random forest      | 0.79     | 0.79      | 0.79   | 0.79     |
| SVM                | 0.81     | 0.80      | 0.80   | 0.80     |
| Naive Bayes        | 0.81     | 0.78      | 0.80   | 0.78     |
| Ensemble model     | **0.86** | **0.85**  | **0.85** | **0.86** |

classifiers. The final experiment results depict that the ensemble classifier achieved the best performance among all the other classifiers with an accuracy of 86%, precision of 0.85%, recall of 85%, and F-measure of 86%, using the feature extraction techniques (TF-IDF). In contrast, the random forest classifier achieves the lowest overall accuracy of 79%, precision of 79%, recall of 79%, and F-measure of 79% among all the classifiers. Thus, this result is considered a good result, according to our sample size of data.

Table 6 displays the traditional ML classifier’s precision results and the ensemble model’s precision, and compares the output of each classifier using the SMS dataset with TF-IDF during model construction.

As a result of this evaluation, it has been determined that the ensemble model classifier is the most accurate of all ML classifiers. The final experiment results depict that the ensemble classifier achieved the best performance among all the other classifiers with an accuracy of 86%, precision of 0.86%, recall of 85%, and F-measure of 86%, using the feature extraction techniques (TF-IDF). In contrast, the SVM classifier achieves the lowest overall accuracy of 79%, precision of 80%, recall of 80%, and F-measure of 80% among all the classifiers. Thus, this result is considered a good result, according to our sample size of data.

Table 7 displays the traditional ML classifier’s precision results and the ensemble model’s precision, and compares the output of each classifier using a textbook dataset with TF-IDF during model construction.

Conclusions and future work

This work presented a novel approach that focuses on the development and discovery of the best ML model to analyze the heterogeneous text data, based on active learning and ensemble methods for machine-learning classifiers to solve the problems of heterogeneous data analyses and reduce data volumes. In this work, the convenient one for ensemble learning has been defined by three extraction techniques: BOW, TF-IDF, and word2vec. On five experimental datasets, several situations were investigated...
for the application of the three extraction strategies using the ensemble model. The findings of the experiment show that the TF-IDF method improves ensemble precision more than others; the results confirmed the proposed approach achieves the research goal by delivering high-performance heterogeneous textual data analysis. Outcomes indicate that the technique suggested is superior and produces better results in most situations than conventional ML algorithms. The studies demonstrate that it is important to use ensemble methods with active learning to increase the model’s overall precision. The huge benefits of active learning help minimize the cost and time required to train the model to classify heterogeneous datasets. The extension of this concept test to a broader dataset is anticipated in the future to focus on test development and to detect phrases by Bert, substituting word2vec with Quick Text. Moreover, by altering the combination technique, we may study more than one form of ensemble classification.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD
Mohamed Salama  https://orcid.org/0000-0002-6273-6748

References
1. Goudjil M, Koudil M, Bedda M, et al. A novel active learning method using SVM for text classification. Int J Automation Comput 2018; 15(3): 290–298.
2. Wang Y, Zhou Z, Jin S, et al. Comparisons and selections of features and classifiers for short text classification. IOP Conf Ser Mater Sci Eng 2017; 261: 012018.
3. Mohammad AH, Alwada’n T and Al-Momani O. Arabic text categorization using support vector machine, Naïve Bayes and neural network. GSTF J Comput (Joc) 2016; 5(1): 108–115.
4. Trstenjak B, Mikac S and Donko D. KNN with TF-IDF based framework for text categorization. Proc Eng 2014; 69: 1356–1364.
5. Hartmann J, Huppertz J, Schamp C, et al. Comparing automated text classification methods. Int J Res Marketing 2019; 36(1): 20–38.
6. Perikos I and Hatzilygeroudis I. Aspect based sentiment analysis in social media with classifier ensembles. In: IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), Wuhan, China, 24–26 May 2017, pp. 273–278.
7. Miller B, Linder F and Mebane WR. Active learning approaches for labeling text: review and assessment of the performance of active learning approaches. Polit Anal 2020; 28(4): 532–551.
8. Budhi GS, Chiong R, Pranata I, et al. Using machine learning to predict the sentiment of online reviews: a new framework for comparative analysis. Arch Comput Methods Eng 2021; 28(4): 2543–2566.
9. Al-Fairouz EI and Al-Hagery MA. The most efficient classifiers for the students’ academic dataset. Int J Adv Comput Sci Appl 2020; 11(9): 501–506.
10. Onan A, Korukoğlu S and Bulut H. A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. Expert Syst Appl 2016; 62: 1–16.
11. Alhaj YA, Wickramaarachchi WU, Hussain A, et al. Efficient feature representation based on the effect of words frequency for Arabic documents classification. In: Proceedings of the 2nd International Conference on Telecommunications and Communication Engineering, 2018, Beijing China, 28–30 November 2018, pp. 397–401.
12. Lilleberg J, Zhu Y and Zhang Y. Support vector machines and word2vec for text classification with semantic features. In: IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC), Beijing, China, 6–8 July 2015, pp. 136–140.
13. Abdelaal HM, Elmahdy AN, Halawa AA, et al. Improve the automatic classification accuracy for Arabic tweets using ensemble methods. J Electr Syst Inf Techn 2018; 5(3): 363–370.
14. Wang G, Sun J, Ma J, et al. Sentiment classification: the contribution of ensemble learning. Decis Support Systems 2014; 57: 77–93.
15. Al-Hagery MA, Al-assaf MA and Al-kharboush FM. Exploration of the best performance method of emotions classification for arabic tweets. Indones J Electr Eng Comput Sci 2020; 19(2): 1010–1020.
16. Al-Azani S and El-Ally E-SM. Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short arabic text. Proced Comput Sci 2017; 109: 359–366.
17. Xu S, Liang H and Baldwin T. Unimelb at semeval-2016 tasks 4a and 4b: An ensemble of neural networks and a word2vec based model for sentiment classification. In: Proceedings of the 10th international workshop on semantic evaluation. SemEval-2016, San Diego, California, 16–17 June 2016, pp. 183–189.
18. Luo X. Efficient english text classification using selected machine learning techniques. Alexandria Eng J 2021; 60(3): 3401–3409.
19. Kowsari K, Jafari Meimandi K, Heidarysafa M, et al. Text classification algorithms: a survey. Information 2019; 10(4): 150.
20. Tharwat A and Schenck W. Balancing exploration and exploitation: a novel active learner for imbalanced data. Knowledge-Based Syst, 2020; 210: 106500.
21. Birman Y, Hindi S, Katz G, et al. Cost-effective ensemble models selection using deep reinforcement learning. *Infor Fus* 2021; 77: 133–148.

22. Wang M, Fu K, Min F, et al. Active learning through label error statistical methods. *Knowledge-Based Syst*, 2020; 189: 105140.

23. Kumar P and Gupta A. Active learning query strategies for classification, regression, and clustering: a survey. *J Comput Sci Techn* 2020; 35(4): 913–945.

24. Brunner U and Stockinger K. Entity matching on unstructured data: an active learning approach. In: 2019 6th Swiss Conference on Data Science (SDS), Bern, Switzerland, 14–14 June 2019, pp. 97–102.

25. Stieglitz S, Mirbabaie M, Ross B, et al. Social media analytics - Challenges in topic discovery, data collection, and data preparation. *Int J Info Manage* 2018; 39: 156–168.

26. Wu J-Y, Hsiao Y-C and Nian M-W. Using supervised machine learning on large-scale online forums to classify course-related Facebook messages in predicting learning achievement within the personal learning environment. *Interact Learn Environ* 2020; 28(1): 65–80.