Journal of Harbin Engineering University
ISSN: 1006-7043
Vol 44 No. 7
July 2023

Enhanced Multilingual Sentiment Analysis Using Ensemble Learning and Tree Structured Parzen Estimator Hyperparameter Optimization

Jehan Sharukh Bhathena, Angad Chaudhry, Dr. Vijayashree J*, Gautham Sai SA, Harsh Mantri

School of Computer Science and Engineering, Vellore Institute of Technology, Vellore
vijayashree.j@vit.ac.in

Abstract

A machine learning tool called sentiment analysis (SA) uses natural language processing (NLP) to infer people's attitudes from text. For a variety of reasons, including ambiguity, a wide range of dialects, a lack of assets, morphological variation, a lack of background information, and the concealing of sentimentality in the unspoken text, implementing Arabic SA is difficult. Convolutional neural networks (CNN) and long short-term memory (LSTM) are two deep learning models that have made major advancements in the Arabic SA sector. The enactment of single DL models has been further enhanced by hybrid models built on CNN coupled with LSTM or gated recurrent unit (GRU). In order to improve application performance, this paper uses the Tree-structured Parzen Estimator (TPE) algorithm for hyperparameter optimization of seven proposed NN models for multilingual sentiment analysis and Ensembles of various models. It also compares the differences in the model's predictive abilities. Only the tweets with negative and positive labels were included in the dataset that we acquired. The models were trained and tested using the Arabic Sentiment Tweets Dataset (ASTD), Depression Corpus of Arabic Tweets (DCAT), Arabic-Egyptian Corpus (AEC), and Hebrew Sentiment Dataset (HSD). The recommended model with TPE has the maximum accuracy for ASTD, DCAT, AEC, and HSD, with 97.7%, 92.2%, 91.1%, and 91.1%, respectively.

Keywords: Deep learning, Multilingual, Sentiment analysis, Tree-structured Parzen Estimator (TPE)

1. Introduction

Sentiment analysis is the process of gathering and analysing sentimental information (such as opinions, attitudes, and others) about a certain social issue. Sentiment analysis helps producers of social content or any kind of product to comprehend the opinions of consumers. This analysis aids in their continual progress and subject-quality enhancement. On commercial websites, for instance, users can publish their requirements, viewpoints, ratings, and reviews of any product [1].

There are many different social networking sites, including Twitter, Facebook, Instagram, and other for-profit services [2]. Opinions or feelings are expressed in the form of words, images, symbols, etc. Different projections of the sentiments are Positivity versus negativity, Good or bad, In favour or against, Emoji Feelings, Like or Dislike, Postulated judgments and Ratings.

The new technology manages communication

between computer systems and people utilising the natural language by using a variety of Artificial Intelligence (AI) techniques. This enables the computer system to comprehend and reply in human languages. AI-based technology's main goal is to read, decode, recognise, and observe language as humans do. In reality, AI-enabled interactions between computers and people can take any of the following forms such as Computer audio interactions, Computers and texting each other, Recognising audio and processing text, (Text to Audio or Audio to Text) data conversion, Computers' Responses to People, (Text or audio) translations across languages.

These AI traits support word processors, interactive responses, sentiment analysis, and personal digital assistants like Alexa. The fundamental building element of sentiment analysis is presented in Figure 1.

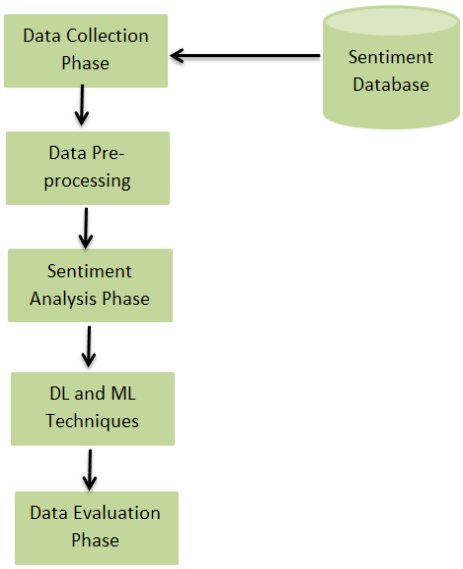

Figure 1 Sentiment Analysis

The phases of data gathering gather data from web databases that need to be pre-processed. Unwanted data are removed at the data pre-processing stage, and the data's structure is controlled. To provide evaluated outcomes, the pre-processed data is studied using DL and ML approaches. Training datasets are used in this evaluation by the ML and DL approaches to make choices. The block diagram shows the universal building blocks that are employed in sentiment analysis. Social news feeds are collected in a database for this study project that is being offered [3].

The goal of the projected system is to examine the multilingual sentiment analysis gathered from sources in multi-Language [4]. When compared to a single language system, this is an extremely difficult task. Analysis of multilingual news feeds requires numerous language patterns and more complex processing stages. Additionally, the suggested sentiment analysis system needs appropriate DL, ML, and AI methods. In particular, the ability to examine the news feeds of a multilingual dataset is necessary for the suggested social news evaluation system.

1.1 Research Gap

This paper fills that need by conducting a comprehensive comparison of sentiment analysis methods in the literature, as well as an experimental investigation to assess the effectiveness of deep learning models, ensembles and related approaches on different datasets.

1.2 Research Questions

Our research question seeks to determine whether it is possible to propose outperforming approaches for a variety of dataset kinds and sizes. We elaborate on earlier studies on Sentiment Analysis performance improvement by analysing the results using a combination of five criteria: Total Accuracy, Precision, F1-score, AUC Score, and Recall.

In all prior studies, sentiment analysis has been found to be a very valuable source of information, and researchers are looking for a strategy that would improve accuracy. In this paper, we compare four different datasets and hyper-tuning parameters using TPE (Tree-Structured Parzen Estimator) Machine Learning and Word embedding by Word2vec. Furthermore, assess the precision, recall, accuracy, AUC score, and F1 Score of each of the seven models individually.

2. Literature Review

2.1 Deep Learning on Multilingual Sentiment Analysis

Deep learning is the use of numerous layers of artificial neural networks for learning tasks [3]. Deep learning is a potent machine learning method taken in research. It can tackle both supervised and unsupervised learning tasks because of its capacity to learn various levels of representations and abstractions from data [4] [5]. For feature extraction and classification, deep learning employs many non-linear processing layers. This literature review discusses the use of several DL models for sentiment analysis, as seen in Figure 2.

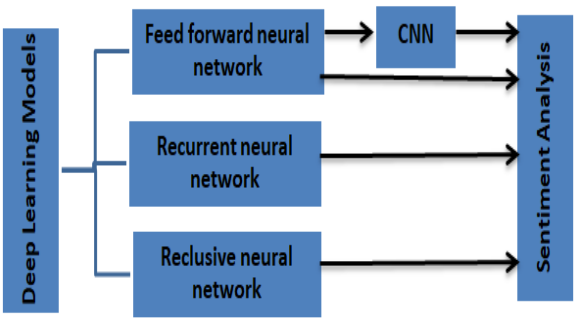

Figure 2 Applications of DL models in Sentiment Analysis

We present an MLP model for sentiment categorization and feature ranking based on decision trees. [6]. The author has suggested a hybrid strategy on the basis of differential evolution and GA (genetic algorithms) to optimize multilayer perceptron neural networks. The aforementioned method is evaluated on the IMDb dataset. An MLP is a feed-forward NN that can have one or more hidden layers in addition to input and output layers. A multilayer perceptron's nodes, which have nonlinear activation functions, each have a number of node layers along with the input layer. To train this kind of network, we use supervised learning. The MLP is shown in Figure 3. [7].

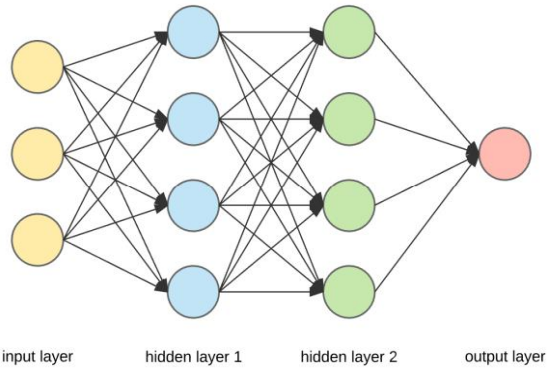

Figure 3 Structure of MLP

The author introduced a DL method for sentiment analysis of brief texts using the ConvLstm neural network architecture, which builds pre-trained word vectors on top of a convolutional neural network and long short-term memory. Stanford Sentiment Treebank (SSTb) and the IMDb dataset were used to assess the suggested model. Pre-trained word vectors for feature abstraction and a composite deep-learning model for text sorting in the movie domain make up the author's primary contribution to the technique's proposal. The proposed method was tested at the phrase level on the IMDb and SSTb datasets, achieving performance accuracy of up to 88.3%. Social media or real-time data were not used in the technique's evaluation [9].

2.2 Ensemble Learning methods

The phrase "ensemble methods" is typically used to describe groups of classifiers. These classifiers are merely slight modifications of the original classifiers. It is regarded as a broader category that includes the blending of various models in various classifier systems. The suggested approach focuses on data-related variation-based ensemble algorithms. Each classifier is learned using a distinct training set as a result of manipulating the training examples [9]. The most popular ensemble learning algorithms are AdaBoost and Bagging; however, there are several variations and additional strategies.

Some researchers introduced a deep learning method for decision mining or sentiment analysis. It is the computational analysis of how people think, feels, and act in relation to various things, including things, services, relationships, people, topics, events, themes, and their features. The field's beginnings and rapid development are comparable to those of internet-based life on the Web, including audits, argument blogs, smaller-scale sites, Twitter, and interpersonal organisations, which have never before existed in the history of mankind. It is typically taken into account in applications for data recovery, web mining, content mining, and data mining [10].

Some studies described a deep learning technique for sentiment analysis in social applications that are improving. The authors proposed combining two ways of examining fundamental assumptions using a few outfit models and data from several sets of highlights [11]. The proposed model is tested on the Twitter and movie reviews datasets, and it provided a quantified report on the outcomes of these coupled models. The results of machine learning are improved by ensemble approaches, which combine many models. In the majority of circumstances, the ensemble of models performs better than the individual models. The ensemble approach typically takes a long time to compute and design. In order to increase classification in sentiment analysis, model selection is the key [12].

Classifiers are typically combined by majority voting. This method should also be capable of providing a confidence interval for the classification result. The goal of an ensemble classifier is to build a collection of individual classifiers that are diverse and accurate enough

to make a more precise classification choice [13][14].

3. Methodology

The datasets that are obtained includes only the tweets with negative and positive labels. The stop words and characters (such as emojis) are removed from this pre-processed dataset. Subsequently, word embedding is performed using the word2vec model. A training-testing development split is performed in the ratio 80:20 and the train set is further divided into the same ratio as shown in Figure 3. Finally, the 7 neural network architectures are created using this dataset.

These architectures are as follows:

Model 1: CNN-GRU-ATTENTION-GRU-ATTENTION

Model 2: CNN-BiLSTM-ATTENTION-GRU-ATTENTION-GRU-ATTENTION

Model 3: CNN-LSTM-GRU-ATTENTION-GRU-ATTENTION

Model 4: CNN-GRU-ATTENTION-LSTM-ATTENTION

Model 5: CNN-BiGRU-ATTENTION-BiLSTM-ATTENTION-CNN

Model 6: BiGRU-ATTENTION-BiLSTM-ATTENTION-CNN

Model 7: CNN-BiGRU-ATTENTION-BiLSTM-ATTENTION

Model 1: CNN-GRU-ATTENTION-GRU-ATTENTION: CNNs capture local patterns and spatial dependencies and GRUs capture sequential information and contextual dependencies and Attention mechanisms emphasize important parts of the input sequence.

Model 2: CNN-BiLSTM-ATTENTION-GRU-ATTENTION-GRU-ATTENTION: BiLSTMs capture past and future context while Attention mechanisms refine attention and focus and Multiple GRU layers allow for deeper feature extraction.

Model 3: CNN-LSTM-GRU-ATTENTION-GRU-ATTENTION: Combination of CNNs, LSTMs, and GRUs captures different aspects of the input sequence and the attention mechanism enhances focus on relevant information.

Model 4: CNN-GRU-ATTENTION-LSTM-ATTENTION: Combination of CNNs, GRUs, LSTMs, and attention mechanisms captures different aspects of sentiment

Model 5: CNN-BiGRU-ATTENTION-BiLSTM-ATTENTION-CNN: CNN layers at the beginning and end capture local patterns and extract features while BiGRUs capture contextual information in both directions. Attention mechanisms emphasize important parts of the sequence and BiLSTM captures long-term dependencies. The final CNN layer summarizes learned representations.

Model 6: BiGRU-ATTENTION-BiLSTM-ATTENTION-CNN: BiGRUs capture contextual information in both directions while attention mechanisms refine focus. BiLSTM is used to capture long-term dependencies and the CNN layer summarizes learned representations.

Model 7: CNN-BiGRU-ATTENTION-BiLSTM-ATTENTION: A combination of CNNs, BiGRUs, BiLSTMs, and attention mechanisms captures context and emphasizes relevant information

For one Arabic dataset and one Hebrew dataset, the identical procedure is repeated.

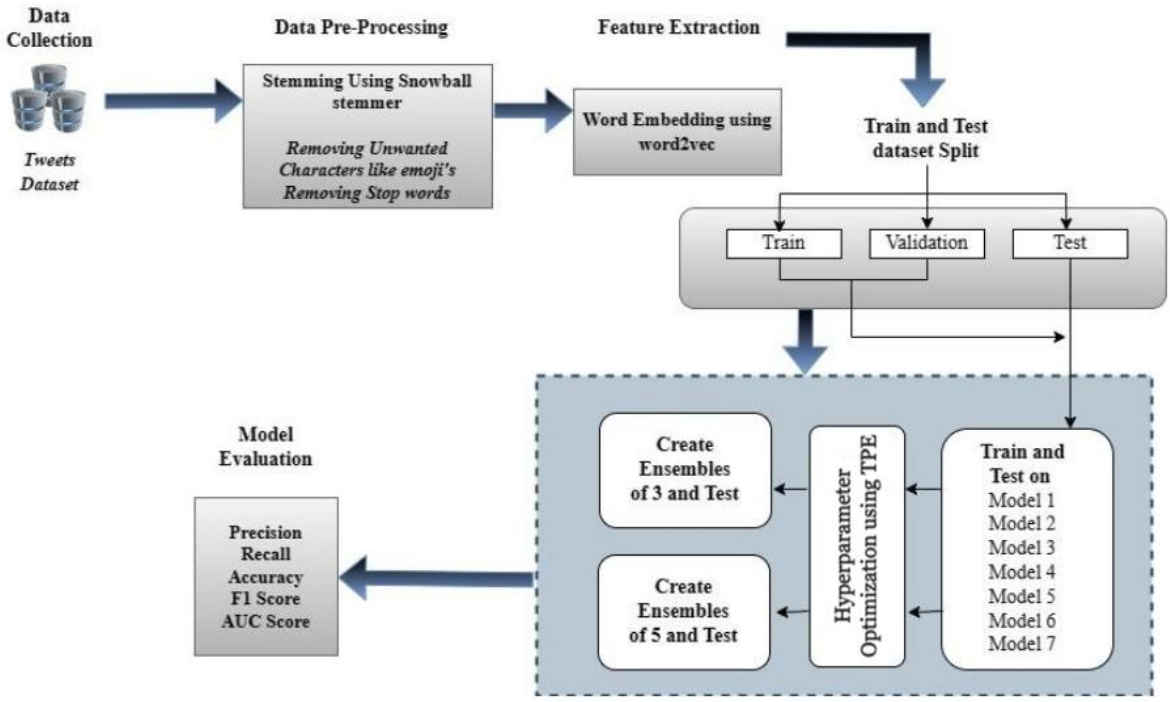

Figure: 3 Proposed Framework

3.1 Dataset of Arabic Sentiment Tweets

It is an uneven dataset because in our article, 1642 of the sampled tweets were favorable and 777 were negative. There are 2419 rows in all. There are no missing values in it. The dataset of ASTD was then split into training and testing halves.

3.2 Depression Corpus of Arabic Tweets

This is an expansion of the Arabic corpus for Egyptian tweets, known as the "Arabic-Egyptian corpus." This corpus comprises 10,000 tweets separated into 5,000 good tweets and 5,000 negative tweets covering a wide range of general Twitter topics.

3.3 Arabic-Egyptian Corpus

Twitter has released a new Arabic corpus for depression detection. This corpus has 10.000 tweets divided into 5.000 of 1 meaning "depressed" tweets and 5.000 of 0 meaning "non-depressed" tweets. Furthermore, the retrieved tweets spanned a wide spectrum of distinct Twitter synonyms for despair and cheerfulness. To the best of our knowledge, this is the first manually annotated Arabic corpus for depression detection from Twitter.

3.4 Hebrew Sentiment Dataset

Hebrew Sentiment is a data set made up of 12,804 user responses to postings on Mr. Reuven Rivlin's official Facebook page in Israel. 12,804 comments were made, of which 370 are neutral, 8,512 are favorable, and 3,922 are negative.

3.5 TPE (Tree-Structured Parzen Estimator) Algorithm

For machine learning models, particularly for black box models like neural network models, hyperparameter optimisation has proven crucial. Correcting hyperparameters when the model turns explicitly becomes a key way to allow the enhancement of model accuracy because it cannot be done during model training [19]. It was time-consuming and ineffective throughout, from the original manual fine-tuning to the subsequent progression of grids and arbitrary search. Numerous techniques for automatic parameter tuning have since been developed, all of which are based on the concepts of accuracy and efficiency. The goal function is minimised

using a model and a function minimization technique called Bayesian optimisation [8]. Given that it makes use of the findings of the previous assessment when attempting the following set of hyperparameters, it is very time-efficient and highly performant [15].

Applying learning hyperparameter models using the Gaussian Mixture Model, TPE is a Bayesian optimisation approach. The Bayes theory's conditional probability idea is first introduced. The chance that the hyperparameter will be x if the model loss is y is denoted by the symbol p(x|y). In the 1st phase, we choose a loss threshold (y) depending on the information at hand, such as the median. For data above and below the threshold, respectively, two probability densities are learned: '(x) and g(x).

$$p(x|y) = \begin{cases} l(x) if\ y < y^* \\ g(x)\ if\ y \geq y^* \end{cases} \quad (1)$$

where g(x) is the density created using the rest of the data and l(x)is the density created using the annotations $\{x^i\}$ that resulted in a loss $f(x^i)$ that was smaller than y^*. To make it easier to optimise Expected Improvement (EI), the TPE technique was used to parametrize p(x, y) as p(y)p(x|y).

$$EI_{y^*}(x) = \int_{-\infty}^{y^*} (y^* - y)p(y|x)dy = \int_{-\infty}^{y^*} (y^* - y)\frac{p(x|y)p(y)}{p(x)}dy \quad (2)$$

Through construction,

$$\gamma = p(y < y^*), and\ p(x) = \int p(x|y)p(y)dy = \gamma l(x) + (1-\gamma)g(x),$$

$$\int_{-\infty}^{y^*} (y^* - y)p(y|x)p(y)dy = l(x)\int_{-\infty}^{y^*} (y^* - y)p(y)dy = \gamma\, y^* l(x) - l(x)\int_{-\infty}^{y^*} p(y)dy \quad (3)$$

The final $EI_{y^*}(x)$ can be written as follows:

$$EI_{y^*}(x) = \frac{\gamma y^* l(x) - l(x)\int_{-\infty}^{y^*} p(y)dy}{\gamma^l(x) + (1-\gamma)g(x)}\alpha\left(\gamma + \frac{g(x)}{l(x)}(1-\gamma)\right)^{-1} \quad (4)$$

3.6 Evaluating Models

The following measurement techniques were employed: recall, f1-score, accuracy, and precision. The performance of the suggested framework and the DL and ML models are assessed using AUC and ROC. The following is a definition of each:

The fraction of accurate predictions compared to all tweets is used to calculate accuracy.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (5)$$

The proportion of correctly categorised positive tweets relative to the total no. of positive tweets is used to calculate the precision.

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

The recall is divided to determine the proportion of correctly categorised positive tweets by the total no. of tweets.

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

The F1 score is the weighted mean of recall and precision,

$$F1 - score = \frac{2.precision.recall}{precision + recall} \quad (8)$$

where TP represents the proportion of correctly formed positively predicted sentences, FP represents incorrectly formed negatively predicted sentences, TN represents correctly formed positively anticipated negative sentences and FN represents correctly formed positively predicted sentences [21].

Additionally, to produce the receiver operating characteristic curve (ROC), the true positive rate (TPR) and false positive rate (FPR) are plotted against one another at various threshold levels [17]. Furthermore, it maps the classification results from most to least favourable [18]. The area under the curve (AUC) was also calculated. AUC measures how well a model can distinguish b/w models, where s_p denotes the number of positive entries and n_p, n_n denotes the number of negative records. [16].

$$\frac{s_p - n_p + n_{(n+1)/2}}{n_p n_n} \quad (9)$$

4. Results

This part of the paper evaluates the effectiveness of the suggested model in comparison to conventional models of machine and deep learning. Additionally, it shows the outcomes of the suggested archetypal for the 3 Arabic datasets and one Hebrew dataset. Accuracy, f1-score, recall, precision, and ROC curve are used to express the results.

4.1 Arabic Sentiment Tweets Dataset (ASTD)

The performance data for the Arabic Sentiment Tweets dataset is shown in this section. Figure 4 shows the ROC curve, AUC values, and proposed model for the dataset of Arabic Sentiment Tweets. The highest AUC scores of 0.964 were attained using model 5, and the lowest AUC scores of 0.909 were achieved by model 3. In comparison to DL models and the proposed models, other models had AUC scores of 0.951, 0.935, 0.909, 0.941, 0.964, and 0.949 for models 1, 2, 3, 4, 5, and 6, respectively. The second-best AUC score, recorded by the suggested model, is 0.959.

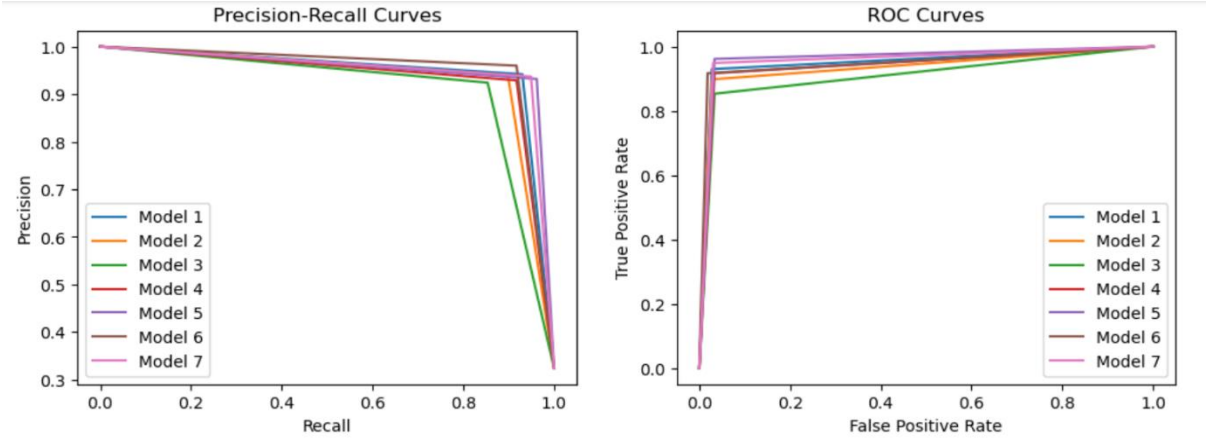

Figure 4 The Scores of the AUC and ROC curve for the Arabic Sentiment Tweets dataset

Table 1 lists the results of the testing for Arabic Sentiment Tweets Datasets in terms of four metrics: accuracy, f1-score, recall, and precision. Model 5 performed the best in terms of accuracy, precision, recall, and f1-score (0.964, 0.932, 0.961, and 0.946, respectively), whereas Model 3 got the lowest results (0.929, 0.924, 0.853, and 0.887, respectively).

Table1 Results of model performance for the Arabic Sentiment Tweets Dataset

	Models	Accuracy	Precision	Recall	F1-Score	AUC
0	[model1]	0.958678	0.941935	0.929936	0.935897	0.951207
1	[model2]	0.948347	0.940000	0.898089	0.918567	0.935283
2	[model3]	0.929752	0.924138	0.853503	0.887417	0.909932
3	[model4]	0.950413	0.929032	0.917197	0.923077	0.941779
4	[model5]	0.964876	0.932099	0.961783	0.946708	0.964072
5	[model6]	0.960744	0.960000	0.917197	0.938111	0.94942
6	[model7]	0.962810	0.937107	0.949045	0.943038	0.959232

Ensemble Deep Learning Output:

	Models	Accuracy	Precision	Recall	F1-Score	AUC
0	[model1, model2, model3, model4, model5]	0.969008	0.967105	0.936306	0.951456	0.960508
1	[model1, model2, model3, model4, model6]	0.966942	0.966887	0.929936	0.948052	0.957323
2	[model1, model2, model3, model4, model7]	0.969008	0.967105	0.936306	0.951456	0.960508
3	[model1, model2, model3, model5, model6]	0.973140	0.973684	0.942675	0.957929	0.965221
4	[model1, model2, model3, model5, model7]	0.973140	0.973684	0.942675	0.957929	0.965221
5	[model1, model2, model3, model6, model7]	0.971074	0.973510	0.936306	0.954545	0.962037
6	[model1, model2, model4, model5, model6]	0.973140	0.961538	0.955414	0.958466	0.968533
7	[model1, model2, model4, model5, model7]	0.973140	0.961538	0.955414	0.958466	0.968533
8	[model1, model2, model4, model6, model7]	0.973140	0.961538	0.955414	0.958466	0.968533
9	[model1, model2, model5, model6, model7]	0.977273	0.962025	0.968153	0.965079	0.974902

10	[model1, model3, model4, model5, model6]	0.977273	0.967949	0.961783	0.964856	0.973246
11	[model1, model3, model4, model5, model7]	0.975207	0.967742	0.955414	0.961538	0.970062
12	[model1, model3, model4, model6, model7]	0.977273	0.967949	0.961783	0.964856	0.973246
13	[model1, model3, model5, model6, model7]	0.973140	0.961538	0.955414	0.958466	0.968533
14	[model1, model4, model5, model6, model7]	0.975207	0.961783	0.961783	0.961783	0.971717
15	[model2, model3, model4, model5, model6]	0.973140	0.967532	0.949045	0.958199	0.966877
16	[model2, model3, model4, model5, model7]	0.973140	0.967532	0.949045	0.958199	0.966877
17	[model2, model3, model4, model6, model7]	0.975207	0.967742	0.955414	0.961538	0.970062
18	[model2, model3, model5, model6, model7]	0.971074	0.961290	0.949045	0.955128	0.965348
19	[model2, model4, model5, model6, model7]	0.973140	0.961538	0.955414	0.958466	0.968533
20	[model3, model4, model5, model6, model7]	0.973140	0.961538	0.955414	0.958466	0.968533

4.2 Depression Corpus of Arabic Tweets (DCAT)

The performance data for the Depression Corpus Arabic Tweets dataset is shown in this section. Figure 5 shows the ROC curve, AUC values, and proposed model for the dataset of Arabic Sentiment Tweets. The highest AUC scores of 0.915 were attained using the proposed model 7, and the lowest AUC scores of 0.841 were achieved by model 3. In comparison to DL models and the proposed models, other models had AUC scores of 0.896, 0.894, 0.841, 0.903, 0.901, and 0.901 for models 1, 2, 3, 4, 5, and 6, respectively. The best AUC score, recorded by the suggested model, is 0.915.

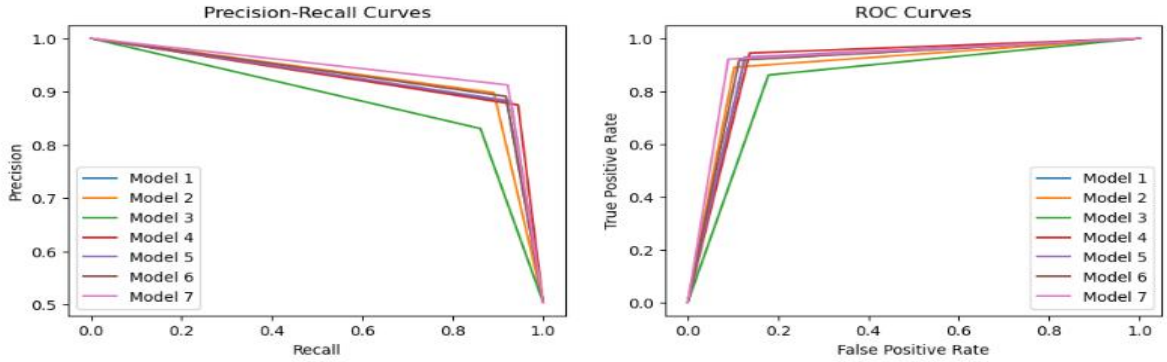

Figure 5 The Scores of the AUC and ROC curve for the Depression Corpus of Arabic tweets

Table 2 lists the results of the testing for Depression Corpus of Arabic Tweets Datasets in terms of four metrics: accuracy, f1-score, recall, and precision. Proposed Model 7 performed the best in terms of accuracy, precision, recall, and f1-score (0.916, 0.912, 0.921, and 0.917, respectively), whereas Model 3 got the lowest results (0.841, 0.830, 0.861, and 0.845, respectively).

Table2 Results of model performance for the Depression Corpus of Arabic Tweets

Models	Accuracy	Precision	Recall	F1-Score	AUC
0 [model1]	0.8970	0.881292	0.919722	0.900097	0.896794
1 [model2]	0.8940	0.898102	0.890981	0.894527	0.894027
2 [model3]	0.8415	0.830784	0.861249	0.845742	0.841321
3 [model4]	0.9040	0.875115	0.944500	0.908484	0.903632
4 [model5]	0.9015	0.882298	0.928642	0.904877	0.901254
5 [model6]	0.9020	0.891242	0.917740	0.904297	0.901857
6 [model7]	0.9160	0.912659	0.921705	0.917160	0.915948

Ensemble Deep Learning Output:

Models	Accuracy	Precision	Recall	F1-Score	AUC
[model1, model2, model3, model4, model5]	0.9150	0.895382	0.941526	0.917874	0.914759

[model1, model2, model3, model4, model6]	0.9130	0.901057	0.929633	0.915122	0.912849
[model1, model2, model3, model4, model7]	0.9180	0.906641	0.933598	0.919922	0.917858
[model1, model2, model3, model5, model6]	0.9095	0.897313	0.926660	0.911750	0.909344
[model1, model2, model3, model5, model7]	0.9120	0.897045	0.932607	0.914480	0.911813
[model1, model2, model3, model6, model7]	0.9110	0.903007	0.922696	0.912745	0.910894
[model1, model2, model4, model5, model6]	0.9145	0.899809	0.934589	0.916869	0.914318
[model1, model2, model4, model5, model7]	0.9195	0.903810	0.940535	0.921807	0.919309
[model1, model2, model4, model6, model7]	0.9165	0.904808	0.932607	0.918497	0.916354
[model1, model2, model5, model6, model7]	0.9135	0.902697	0.928642	0.915486	0.913362
[model1, model3, model4, model5, model6]	0.9120	0.893296	0.937562	0.914894	0.911768
[model1, model3, model4, model5, model7]	0.9185	0.902091	0.940535	0.920912	0.918300
[model1, model3, model4, model6, model7]	0.9160	0.903939	0.932607	0.918049	0.915849
[model1, model3, model5, model6, model7]	0.9105	0.896750	0.929633	0.912895	0.910326
[model1, model4, model5, model6, model7]	0.9180	0.901235	0.940535	0.920466	0.917795
[model2, model3, model4, model5, model6]	0.9185	0.904398	0.937562	0.920681	0.918327
[model2, model3, model4, model5, model7]	0.9205	0.906310	0.939544	0.922628	0.920327
[model2, model3, model4, model6, model7]	0.9185	0.905950	0.935580	0.920527	0.918345

[model2, model3, model5, model6, model7]	0.9170	0.906461	0.931615	0.918866	0.916867
[model2, model4, model5, model6, model7]	0.9230	0.908309	0.942517	0.925097	0.922823
[model3, model4, model5, model6, model7]	0.9195	0.905354	0.938553	0.921655	0.919327

4.3 Arabic-Egyptian Corpus (AEC)

The performance data for the Arabic-Egyptian Corpus dataset is shown in this section. Figure 6 shows the ROC curve, AUC values, and proposed model for the dataset of Arabic Sentiment Tweets. The highest AUC score of 80.6 was attained using model 4, and the lowest AUC score of 0.748 was achieved by model 3. In comparison to DL models and the proposed models, other models had AUC scores of 0.786, 0.775, 0.748, 0.806, 0.784, and 0.796 for models 1, 2, 3, 4, 5, and 6, respectively. The Second-best AUC score, recorded by the suggested model 7, is 0.804.

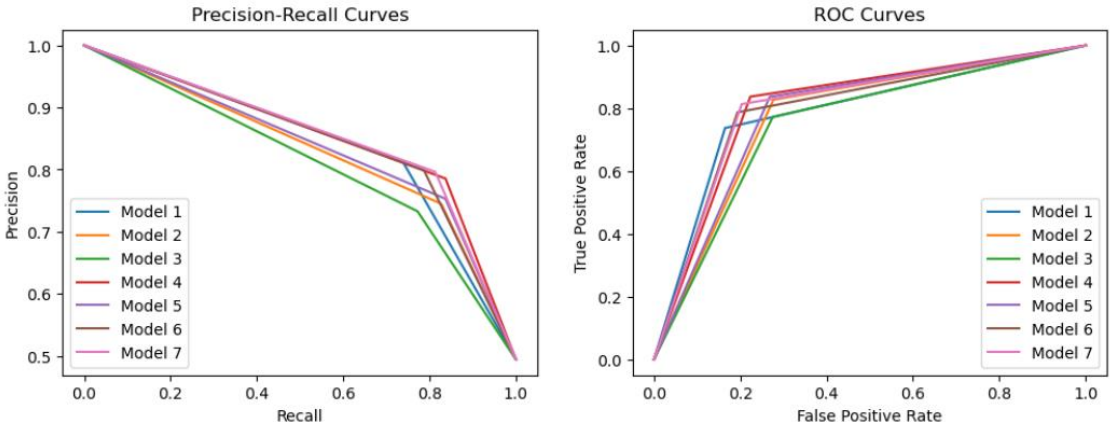

Figure 6 The Scores of the AUC and ROC curve for the Arabic-Egyptian Corpus

Table 3 lists the results of the testing for Arabic-Egyptian Corpus Datasets in terms of four metrics: accuracy, f1-score, recall, and precision. Model 4 performed the best in terms of accuracy, precision, recall, and f1-score (0.806, 0.785, 0.836, and 0.810, respectively), whereas Model 3 got the lowest results (0.748, 0.732, 0.772, and 0.751, respectively).

Table3 Results of model performance for the Arabic-Egyptian Corpus

Models		Accuracy	Precision	Recall	F1-Score	AUC
0	[model1]	0.786723	0.813915	0.736909	0.773500	0.786151
1	[model2]	0.774472	0.744817	0.826967	0.783745	0.775075
2	[model3]	0.748219	0.732773	0.772072	0.751909	0.748493
3	[model4]	0.806476	0.785562	0.836833	0.810387	0.806825
4	[model5]	0.783598	0.752960	0.836580	0.792570	0.784207
5	[model6]	0.796850	0.799075	0.786744	0.792862	0.796733
6	[model7]	0.804851	0.796333	0.813053	0.804606	0.804945

Ensemble Deep Learning Output:

	Models	Accuracy	Precision	Recall	F1-Score	AUC
0	[model1, model2, model3, model4, model5]	0.9055	0.901198	0.909366	0.905263	0.905527
1	[model1, model2, model3, model4, model6]	0.9085	0.902584	0.914401	0.908454	0.908541
2	[model1, model2, model3, model4, model7]	0.9100	0.914373	0.903323	0.908815	0.909954
3	[model1, model2, model3, model5, model6]	0.9035	0.899202	0.907351	0.903258	0.903527
4	[model1, model2, model3, model5, model7]	0.9000	0.904179	0.893253	0.898683	0.899953
5	[model1, model2, model3, model6, model7]	0.9040	0.907426	0.898288	0.902834	0.903960
6	[model1, model2, model4, model5, model6]	0.9045	0.901804	0.906344	0.904068	0.904513
7	[model1, model2, model4, model5, model7]	0.9080	0.913177	0.900302	0.906694	0.907946
8	[model1, model2, model4, model6, model7]	0.9115	0.917178	0.903323	0.910198	0.911443
9	[model1, model2, model5, model6, model7]	0.9050	0.909276	0.898288	0.903749	0.904953

10	[model1, model3, model4, model5, model6]	0.9035	0.891389	0.917422	0.904218	0.903597
11	[model1, model3, model4, model5, model7]	0.9055	0.903614	0.906344	0.904977	0.905506
12	[model1, model3, model4, model6, model7]	0.9110	0.907908	0.913394	0.910643	0.911017
13	[model1, model3, model5, model6, model7]	0.9020	0.897308	0.906344	0.901804	0.902030
14	[model1, model4, model5, model6, model7]	0.9075	0.903194	0.911380	0.907268	0.907527
15	[model2, model3, model4, model5, model6]	0.9030	0.888997	0.919436	0.903960	0.903114
16	[model2, model3, model4, model5, model7]	0.9040	0.900100	0.907351	0.903711	0.904023
17	[model2, model3, model4, model6, model7]	0.9075	0.898422	0.917422	0.907823	0.907569
18	[model2, model3, model5, model6, model7]	0.9020	0.896517	0.907351	0.901902	0.902037
19	[model2, model4, model5, model6, model7]	0.9055	0.901198	0.909366	0.905263	0.905527
20	[model3, model4, model5, model6, model7]	0.9050	0.894789	0.916415	0.905473	0.905079

4.4 Hebrew sentiment dataset (HSD)

The performance data for the Hebrew sentiment dataset is shown in this section. Figure 7 shows the ROC curve, AUC values, and proposed model for the dataset of Arabic Sentiment Tweets. The highest AUC scores of 0.882 were obtained using the proposed model 7, and the lowest AUC scores of 0.500 were achieved by model 4. In comparison to DL models and the proposed models, other models had AUC scores of 0.864, 0.860, 0.832, 0.500, 0.872, and 0.883 for models 1, 2, 3, 4, 5, and 6, respectively. The Second-best AUC score, recorded by the suggested model 7, is 0.883.

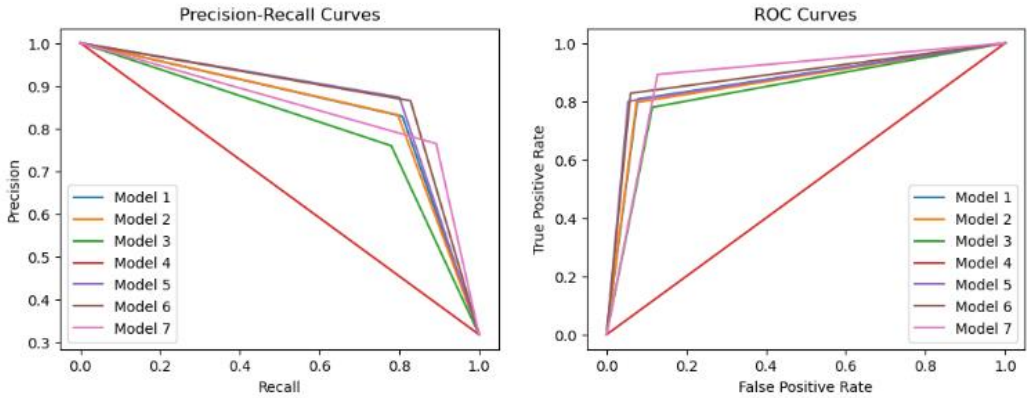

Figure 7 The Scores of the AUC and ROC curve for the Hebrew sentiment dataset

Table 4 lists the results of the testing for Hebrew sentiment Datasets in terms of four metrics: accuracy, f1-score, recall, and precision. Model 5 performed the best in terms of accuracy, precision, recall, and f1-score (0.899, 0.872, 0.800, and 0.834, respectively), whereas Model 4 got the lowest results (0.682, 0.000, 0.000, and 0.000, respectively). The second-best scores are achieved for model 1, accuracy is 0.885.

Table4 Results of model performance for the Hebrew sentiment dataset

	Models	Accuracy	Precision	Recall	F1-Score	AUC
0	[model1]	0.885852	0.828571	0.807595	0.817949	0.864928
1	[model2]	0.884244	0.832011	0.796203	0.813713	0.860704
2	[model3]	0.851688	0.759556	0.779747	0.769519	0.832453
3	[model4]	0.682476	0.000000	0.000000	0.000000	0.500000
4	[model5]	0.899518	0.872928	0.800000	0.834875	0.872909
5	[model6]	0.904341	0.865079	0.827848	0.846054	0.883889
6	[model7]	0.878617	0.764642	0.892405	0.823598	0.882304

Ensemble Deep Learning Output:

	Models	Accuracy	Precision	Recall	F1-Score	AUC
0	[model1, model2, model3, model4, model5]	0.899116	0.905263	0.762025	0.827491	0.862461
1	[model1, model2, model3, model4, model6]	0.900723	0.898678	0.774684	0.832087	0.867024
2	[model1, model2, model3, model4, model7]	0.896704	0.874824	0.787342	0.828781	0.867464
3	[model1, model2, model3, model5, model6]	0.907154	0.881310	0.817722	0.848326	0.883242
4	[model1, model2, model3, model5, model7]	0.903939	0.861075	0.831646	0.846104	0.884610
5	[model1, model2, model3, model6, model7]	0.904743	0.855856	0.841772	0.848756	0.887906
6	[model1, model2, model4, model5, model6]	0.903939	0.902190	0.782278	0.837966	0.871410
7	[model1, model2, model4, model5, model7]	0.902733	0.891429	0.789873	0.837584	0.872557
8	[model1, model2, model4, model6, model7]	0.903135	0.888260	0.794937	0.839011	0.874206
9	[model1, model2, model5, model6, model7]	0.908762	0.874834	0.831646	0.852693	0.888143

10	[model1, model3, model4, model5, model6]	0.903135	0.901903	0.779747	0.836388	0.870144
11	[model1, model3, model4, model5, model7]	0.903135	0.887165	0.796203	0.839226	0.874544
12	[model1, model3, model4, model6, model7]	0.901527	0.878999	0.800000	0.837641	0.874382
13	[model1, model3, model5, model6, model7]	0.910772	0.870757	0.844304	0.857326	0.893000
14	[model1, model4, model5, model6, model7]	0.908360	0.896893	0.803797	0.847797	0.880403
15	[model2, model3, model4, model5, model6]	0.899920	0.907994	0.762025	0.828630	0.863050
16	[model2, model3, model4, model5, model7]	0.902331	0.895803	0.783544	0.835922	0.870571
17	[model2, model3, model4, model6, model7]	0.903537	0.893983	0.789873	0.838710	0.873146
18	[model2, model3, model5, model6, model7]	0.909968	0.876330	0.834177	0.854734	0.889703
19	[model2, model4, model5, model6, model7]	0.908762	0.900427	0.801266	0.847957	0.880020
20	[model3, model4, model5, model6, model7]	0.911174	0.903546	0.806329	0.852174	0.883141

5. Discussion

5.1 Performance Analysis

Every one of the four datasets, the top-performing ML, DL, and suggested models are displayed in Figures 8–10. For the four datasets, model 5 produced by ML algorithms performed the best in terms of 4 parameters for performance measurement. The model 7 obtained the best outcomes in terms of AUC scores from the DL models that were optimised. We can see that, when matched to models of machine and deep learning, the proposed models with the TPE algorithm generated the best results for all datasets. Table 5 is showing the units of parameters for all 7 models.

Table 5 List of parameters taken for DL models

	Convolution kernel size	Convolution units	Dropout Rate	Gru units	Lstm units
Model1	3	240	0.5	432,256	-
Model2	7	608	0.2	496,432	160
Model3	4	5	-	8,3	7
Model4	1	192	0.3949	272	384
Model5	3,3	528,224	0.777, 0.2948	368	496
Model6	3,6	544,320	0.6,0.3	336	464
Model7	1	592	0.2,0.4	560	432

According to Figure 8, the Arabic sentiment tweets dataset has shown a 97.7 accuracy percentage; the Depression corpus of Arabic tweets has shown a 92.2 accuracy percentage, the Arabic Egyptian corpus has shown 91.1% and the same accuracy percentages achieved for the last dataset which is Hebrew sentiment.

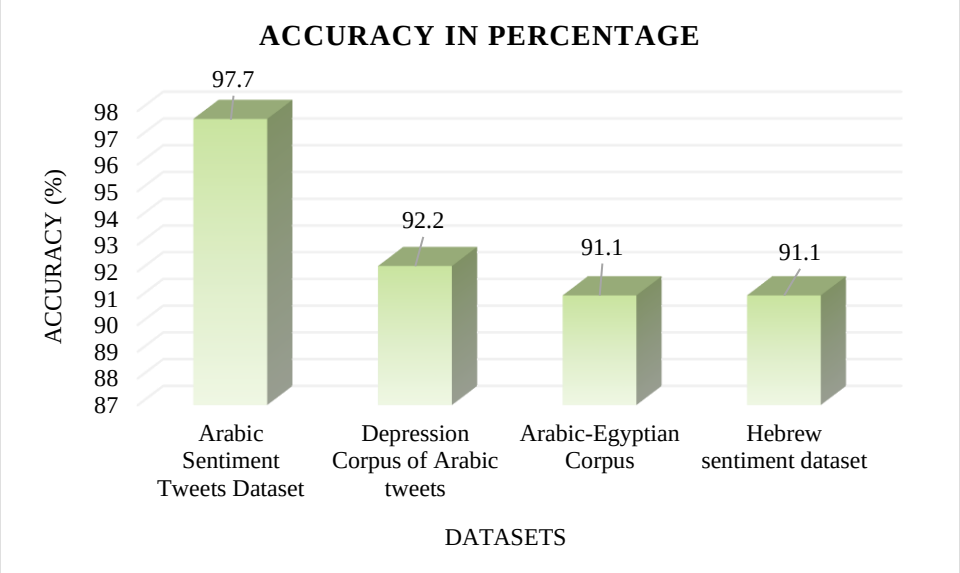

Figure 8 Accuracy percentages of Experimental Datasets

5.2 Validation

Table 6 Comparison between the suggested models and earlier studies.

Ref.	Method	Dataset	Accuracy %
[5]	CNN-LSTM	ASTD	79.18
[7]	CNN-LSTM	ASTD	65.05
[10]	CNN-LSTM	ASTD	66.0
[17]	Bi-LSTM	ASTD	79.25
The proposed models	**Ensemble based on Model1, Model3 Model4, Model6 Model7**	**ASTD**	**97.7**

In all four datasets listed in Table 6, the projected model is contrasted with the body of prior research. The efficacy of other approaches was enhanced by our model, as shown by comparisons between the suggested model and the current models [1], as presented in Figure 9. The accuracy of CNN-LSTM was recorded as 79.18% in comparison to researchers who made use of the ASTD dataset [5]. The accuracy performance for CNN-LSTM in [7] was recorded as 65.05%. CNN-LSTM's accuracy was 66.00% [10]. The accuracy of Bi-LSTM was recorded as 79.25% [15].

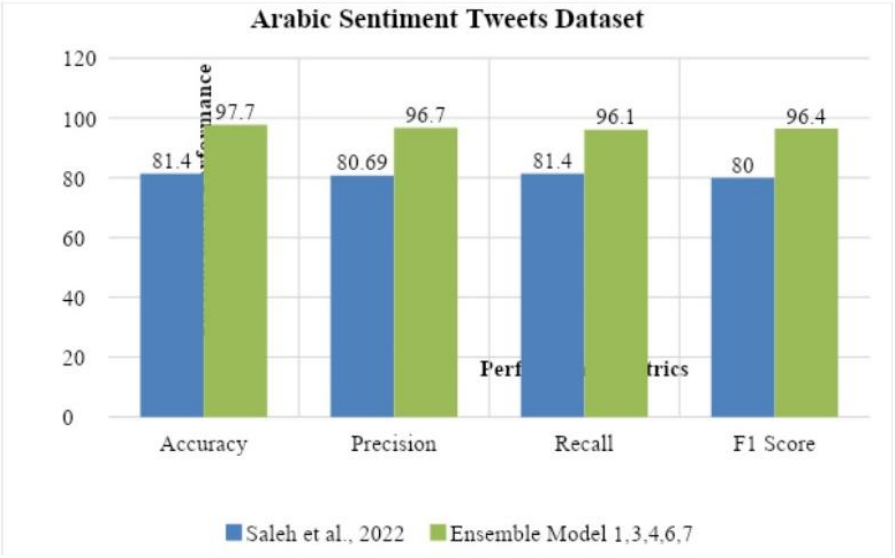

Figure 9 Validation of ASTD Ensembles of Model 1,3,4,6,7.

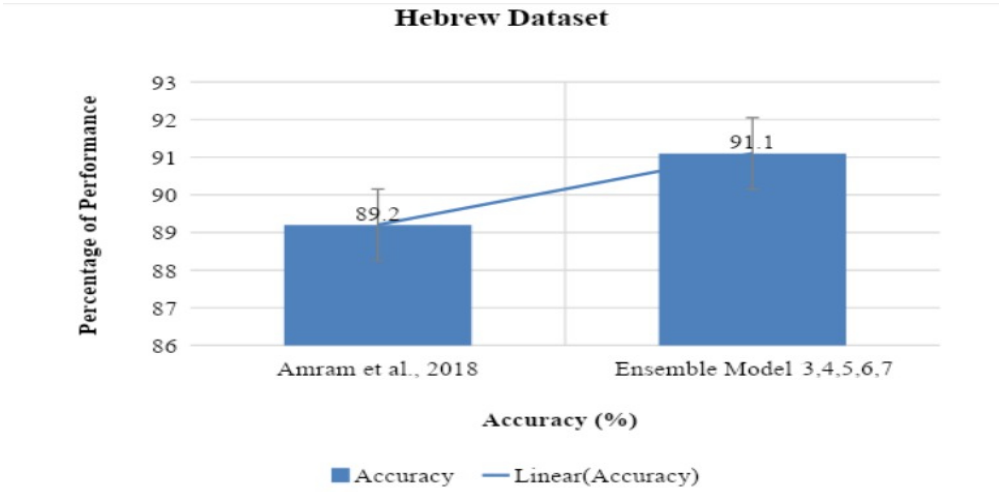

Figure 10 Performance percentage comparison of Ensemble of Model 3,4,5,6,7 on Hebrew dataset

Comparing the suggested model to methods employed in the existing literature, it performed the best overall. In comparison to other algorithms [20], the ensemble of model 3,4,5,6,7 achieved the highest accuracy, as shown in Figure 10.

6. Conclusion

6.1 Summary

The issue of multilingual sentiment analysis is addressed in this paper. A CNN-LSTM model, a CNN-GRU model, and ensemble learning models based on CNN were used to assess the performance of the multilingual sentiment analysis system. The ensemble learning models have additionally made significant improvements to NLP accuracy. The proposed framework is an ideal ensemble model to enhance the machine's predicting Arabic sentiment analysis performance. The tree-Structured Parzen Estimator (TPE) technique was used to extract features for DL models. The suggested model's performance is compared to that of DeepCNN, CNN-LSTM, CNN-GRU, and traditional ML algorithms. The results show that the suggested ensemble model performs well for all datasets when compared to additional models. For ASTD, DCAT, AEC, and HSD, respectively, the suggested model with TPE has the highest accuracy of 97.7%, 92.2%, 91.1%, and 91.1%.

6.2 Limitations of Existing Systems

However, when creating the suggested multilingual Sentiment analysis model, certain significant restrictions must be taken into account. The following list contains the works' restrictions.

No model for robust multilingual sentiment analysis was created.
Multilingual language models did not use efficient ML and DL-based sentiment analysis algorithms.
For sentiment analysis, heterogeneous datasets were not used.
Complex linguistic phrases were not considered when developing sequential DL structures.
There was no comparative research investigation.

The suggested multilingual sentiment analysis approach is created to address these important problems and raise technical demands. This study successfully applies traditional DL methods to deep-trained models.

References

1. Saleh, H., Mostafa, S., Gabralla, L. A., O. Aseeri, A., & El-Sappagh, S. (2022). Enhanced Arabic Sentiment Analysis Using a Novel Stacking Ensemble of Hybrid and Deep Learning Models. *Applied Sciences*, *12*(18), 8967.
2. El-Affendi, M. A., Alrajhi, K., & Hussain, A. (2021). A novel deep learning-based multilevel parallel attention neural (MPAN) model for

multidomain Arabic sentiment analysis. *IEEE Access*, *9*, 7508-7518.

3. Al-Hashedi, A., Al-Fuhaidi, B., Mohsen, A. M., Ali, Y., Gamal Al-Kaf, H. A., Al-Sorori, W., & Maqtary, N. (2022). Ensemble classifiers for Arabic sentiment analysis of social network (Twitter data) towards covid-19-related conspiracy theories. *Applied Computational Intelligence and Soft Computing*, *2022*, 1-10.

4. Salur, M. U., & Aydin, I. (2020). A novel hybrid deep learning model for sentiment classification. *IEEE Access*, *8*, 58080-58093.

5. Al Omari, M., Al-Hajj, M., Sabra, A., & Hammami, N. (2019, October). Hybrid CNNs-LSTM deep analyzer for Arabic opinion mining. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)* (pp. 364-368). IEEE.

6. Sabzevari, M., Martínez-Muñoz, G., & Suárez, A. (2022). Building heterogeneous ensembles by pooling homogeneous ensembles. *International Journal of Machine Learning and Cybernetics*, 1-8.

7. Heikal, M., Torki, M., & El-Makky, N. (2018). Sentiment analysis of Arabic tweets using deep learning. *Procedia Computer Science*, *142*, 114-122.

8. Rong, G., Li, K., Su, Y., Tong, Z., Liu, X., Zhang, J., ... & Li, T. (2021). Comparison of tree-structured parzen estimator optimization in three typical neural network models for landslide susceptibility assessment. *Remote Sensing*, *13*(22), 4694.

9. Sabzevari, M., Martínez-Muñoz, G., & Suárez, A. (2022). Building heterogeneous ensembles by pooling homogeneous ensembles. *International Journal of Machine Learning and Cybernetics*, 1-8.

10. Farha, I. A., & Magdy, W. (2019, August). Mazajak: An online Arabic sentiment analyser. In *Proceedings of the fourth Arabic natural language processing workshop* (pp. 192-198).

11. Elfaik, H. (2021, August). Deep attentional bidirectional LSTM for Arabic sentiment analysis in Twitter. In *2021 1st International Conference on Emerging Smart Technologies and Applications (eSmarTA)* (pp. 1-8). IEEE.

12. Oussous, A., Lahcen, A. A., & Belfkih, S. (2019, March). Impact of text pre-processing and ensemble learning on Arabic sentiment analysis. In *Proceedings of the 2nd International Conference on Networking, information systems & Security* (pp. 1-9).

13. Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1* (pp. 1-15). Springer Berlin Heidelberg.

14. Ludmila, I. (2004). *Combining pattern classifiers: methods and algorithms*. Wiley.

15. Kaddoura, S., Itani, M., & Roast, C. (2021). Analyzing the effect of negation in sentiment polarity of Facebook dialectal Arabic text. *Applied Sciences*, *11*(11), 4768.

16. Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information*, *10*(4), 150.

17. Alayba, A. M., Palade, V., England, M., & Iqbal, R. (2018, March). Improving sentiment analysis in Arabic using word representation. In *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)* (pp. 13-18). IEEE.

18. Wang, B., Wang, A., Chen, F., Wang, Y., & Kuo, C. C. J. (2019). Evaluating word embedding models: Methods and experimental results. *APSIPA transactions on signal and information processing*, *8*, e19.

19. Shawki, N., Nunez, R. R., Obeid, I., & Picone, J. (2021, December). On automating hyperparameter optimization for deep learning applications. In *2021 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)* (pp. 1-7). IEEE.

20. Amram, A., David, A. B., & Tsarfaty, R. (2018, August). Representations and architectures in neural sentiment analysis for morphologically rich languages: A case study from Modern Hebrew. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 2242-2252).

21. Sarker, I. H. (2019). A machine learning-based robust prediction model for real-life mobile phone data. *Internet of Things*, *5*, 180-193.

22. Saleh, H., Mostafa, S., Alharbi, A., El-Sappagh, S., & Alkhalifah, T. (2022). Heterogeneous ensemble deep learning model for enhanced Arabic sentiment analysis. *Sensors*, *22*(10), 3707.