Available Online at www.ijcsmc.com/journal

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 11, Issue. 5, May 2022, pg.222 – 233

Machine Learning for Multilabel Emotion Classification in Arabic Tweets

Enas A. H. Khalil[1]; Enas M.F. El Houby[2]; Hoda K. Mohamed[3]
[1]Systems & Information Department, Engineering Research Division, National Research Centre, Egypt
[2]Systems & Information Department, Engineering Research Division, National Research Centre, Egypt
[3]Ain Shams University, Faculty of Engineering, Computers & Systems Department, Egypt
[1] enaskhalil@gmail.com; [2] enas_mfahmy@yahoo.com; [3] hoda.korashy@eng.asu.edu.eg

DOI: https://doi.org/10.47760/ijcsmc.2022.v11i05.010

Abstract*— Multilabel emotion classification is a high priority because it mimics real-life scenarios in which people display a variety of emotions. The text could express a collection of emotions such as happiness, love, and optimism, or sadness, anger, and pessimism. In this framework, the Arabic tweets data provided by SemEval 2018-Task1, E-c subtask have been first preprocessed through different normalization steps, including stemming, stop word removal, special characters, and digits removal. An emotion lexicon has been built to replace the emotions with their meaning related to emotion classes. A word embedding pre-trained model Aravec has been implemented for the feature extraction process because word embedding performed better in this task than other features such as the N-gram model. In the classification process of our framework, different machine learning techniques have been implemented, including Multi-Layer Perceptron (MLP), Support Vector Machine SVM, K Nearest Neighbor (KNN), Ensemble Random Forest (RF), and Ensemble Extra Tree. The best performance was achieved using MLP, whereas SVM proved to perform best over other Traditional machine learning techniques such as KNN, RF, and Extra tree. Extra tree achieved a multilabel Jaccard accuracy of 26.2%, Nearest Neighbor (KNN) of 37.5%, Ensemble Random Forest (RF) of 29.1%, and SVM accuracy of 46.3%. A neural network model Multi-Layer Perceptron (MLP), achieved an accuracy of 48%. The proposed framework has been compared with different previous machine learning models built for this task; the results obtained by the proposed framework outperform other previous models in most cases.

Keywords*— Sentiment analysis, emotion analysis, Machine Learning, word embedding, multilabel classification.

I. INTRODUCTION

Arabic is one of the world's most famous languages worldwide. Arabic Sentiment Analysis and Text Classification for social networks are important issues in obtaining valuable Arabic text insights. For example, in the Arab Spring, social media such as Facebook, Twitter, and Instagram played a vital role in recording and spreading these variations.

Sentiment analysis S.A. plays an important role in many real-world applications. S.A. aids business intelligence in evaluating consumer reviews about a product [1], helps in decision making for stock market prediction[2], and in the classification of different dialects in different languages, as in Arabic[3]. The spam identification on social media is successfully checked by automatically identifying spam terms across many forums, emails, and blogs through S.A. systems [4], opinion summarization, and public opinion analysis [5, 6]. Tracking people's responses to political problems on social media by analysing offensive phrases across many online sites is an important application of the SA system. While sentiment analysis aims to detect positive, neutral, or negative opinions from text, emotion analysis (EA) is the most common sentiment analysis-related

field to see and recognize types of feelings through text expression. Emotions play a vital role in everyday communications. Simply saying 'ok' can have a different meaning depending on the context, conveying remorse, sadness, anger, happiness, or even disgust. However, the full meaning can be understood using facial expressions, hand gestures, and other non-verbal communication.

EA can be used in various applications as an empathic robot [7] to recognize human emotions through facial expressions. A multimodal affect recognition system in [8] has been developed to classify whether a customer likes or dislikes a product examined at a counter by analysing the consumer's facial expression, hand gestures, body posture, and voice after testing the product.

The need for accurate emotion mining models in modern standard Arabic and dialectal Arabic increases drastically due to the wide use of Arabic sites over the internet. However, Arabic suffers from the lack of a large-scale emotion lexicon due to the high morphological complexity and high ambiguity in the Arabic language.

Our proposed work implemented a multilabel emotion classification in the Arabic language. Several preprocessing steps include the removal of stop-words, non-Arabic words, digits, and applying a reliable Arabic light stemmer (ARLSTM). The dataset used is that of SemEval-2018 Task1.

The tweets are encoded as feature vectors using AraVec word embedding models [9].

These embedded vectors feed different machine learning models. This study's contribution may be described as follows:

The use of different machine learning models for multilabel Arabic emotion analysis.

The use of preprocessing procedures for social media colloquial text.

Each word in our tweets text data is represented by a pertained word embedding model (Aravec), which has been proved to be the best method for features representation other than other methods.

The model performance for different machine learning models has been evaluated and compared with the machine learning framework previously implemented for the same task.

The remainder of the paper is organized as follows: Section II browsed the related Work in Arabic sentiment and emotion analysis. The Methodology is described in detail in section III. The results are reported in Section IV. Finally, Section V concludes the work and suggests future work.

II. RELATED WORK

Detecting emotion in social networks is not an easy task; it faces many difficulties, including the casual style of text data in social networks, since the social media text data do not obey grammatical rules and the use of slang. Also, semantic ambiguity in human emotions makes them difficult to be inferred. Different emotions have fuzzy boundaries and cannot be distinguished easily, making emotion labelling so tedious and time-consuming. Finally, inconsistency in annotation or labelling data; different annotators may classify the same text message into different emotion classes[9].

The work in [10] focuses on investigating tweets data as either positive or negative it also considers the emotions in tweets data for sentiment analysis. Logistic Regression (L.R.) and Naïve Bayes (N.B.) are used with one data set, while RNN, GRU, and LSTM models are implemented with another dataset. The L.R. accuracy outperforms that of N.B. Logistic regression also the LSTM accuracy outperforms that of GRU.

EmotexStream [9] is a two-stage framework that identifies live streams of text messages for real-time emotion monitoring. In the first stage, a binary classifier separated tweets with or without emotion. The second stage is a multi-class emotion classification of tweets with explicit emotion (happy-active, happy-inactive, unhappy-active, unhappy-inactive). Emotex offline system including data acquisition, collecting training data. The feature selection process includes unigram features, Emoticon features, Punctuation features, and negation features. However, the feature set did not consider context or the data's sequential structure. The last stage is the emotion classifiers selection. Three different classifiers have been implemented Naive Bayes, a probabilistic classifier, SVM as a decision boundary classifier, and decision tree as a rule-based classifier. The labelling of tweets is done automatically by assigning the emotion from emotional hashtags inserted by authors. For example, a tweet with the hashtag "#excited" can be labelled as expressing excitement emotion. A large corpus of labelled messages has been built to train classifiers with no manual effort.

The difficulty in analysing emotions is that people may exhibit one or several emotions within a single expression. Humans can recognize these diverse emotions, but it is still difficult for an emotion analysis system to do so; this, of course, necessitated the building of multilabel text classification systems. Multilabel text classification has recently been the subject of some research on modest benchmarks that are not publicly accessible.

Because Arabic is the United Nations' sixth official language [17], the amount of online Arabic data grows every day, necessitating the development of effective automated text classification systems.

A multilabel Arabic dataset has been constructed using manual annotation and a semi supervised technique [11]. A balanced dataset of 30,000 posts from Facebook and 14,000 Twitter tweets have been collected from the most-liked Arabic Facebook pages and Arabic Twitter accounts. They are divided into eleven balanced classes corresponding to different topics: politics, sports, economics, religion, weather, technology, TV, food, health, ads, and porn. This dataset can be used for short text classification, sentiment analysis, and multilabel classification. This dataset is the first Arabic data to include more than one label for each post. One label for topic classification and the other for classifying sentiment as positive, neutral, or negative. Nine machine learning classifiers with three feature representations were applied to evaluate the performance of the dataset. The LinearSVC classifier with N-gram (1, 2) outperformed other classifiers with parameter tuning and chi-square feature selection, with an accuracy score of 97, 92. The effect of dataset size on the effectiveness of the results was also studied; the experiments proved that the larger the dataset size, the better the results. Hate speech on social networking sites has also been investigated in this research. A dataset for vulgar Arabic speech contained 6,000 posts, including comments and tweets manually labelled as hate or non-hate speech.

[14] Provided a new multilabel Arabic news article benchmark dataset for text classification and other supervised learning applications. The news site Russia Today "RTAnews" was employed as a baseline in this study. This dataset contains a big unbalanced (real-world scenario) multilabel Arabic text benchmark data set. Four multilabel transformation based algorithms are evaluated in this benchmark: Binary Relevance, Classifier Chains, Calibrated Ranking by Pairwise Comparison, and Label Powerset, with three base learners (Support Vector Machine, k-Nearest-Neighbors, and Random Forest); and four adaptation-based algorithms are evaluated (Multi-label kNN, Instance-Based Learning by Logistic Regression Multi-label, Binary Relevance kNN and RFBoost).

Both RFBoost and Label Powerset using Support Vector Machine as the core learner outperformed other comparable methods, according to the baseline data. Adaptation-based algorithms were also shown to be quicker than transformation-based algorithms.

A hierarchical multilabel Arabic text classification (HMATC) model based on the HOMER algorithm has been implemented [12]. The essential parameters of the HOMER algorithm have been optimized using different sets of multilabel classifiers, clustering algorithms, and different numbers of clusters. The dataset was a raw dataset related to the Islamic field and written in Modern Standard Arabic (MSA). The data contained about 100,000 text instances (Islamic Fatwa requests); the impact of feature selection methods and feature set dimensions on the model performance has been studied. This work introduces a multilabel Arabic dataset in an appropriate format for a hierarchical classification and makes it publicly available online. The model outperforms all models considered in the experiments.

A multilabel emotion mining on Arabic tweets has been implemented in [13]. The data set used is that of Semeval 2018 task1 E-c subtask. In this subtask E-C, the emotional state of the tweeter is annotated and classified as either neutral (no emotion) or one or more of eleven emotions: Anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise and trust. E-c Arabic tweets data set consists of 2278 tweets for training, 1518 tweets for the test, and 585 tweets for development sets. First, normalization rules include deletion of tashkeel or diacritics, hamza, elongations, and non-Arabic letters [18]. Then the most frequently used emojis are substituted with the Arabic word corresponding to them, using a manually generated lexicon. Finally, for stemming, ARLSTEM [19] has been employed. Many features have been tried throughout the feature selection step. However, the finest feature was AraVec's word embedding [10]. Finally, the tweet has been classified as neutral or one of the eleven emotions. With an accuracy of 48.9%, the support vector classifier (SVC) outperformed all other classifiers.

TeamUNNC [14] also implemented multilabel emotion classification using Semeval 2018 task1 E-c dataset. Individual words were tokenized, white space was removed, and also punctuation marks were. Word2vec embedding AraVec has been used for feature selection. Finally, a fully connected NN with three dense hidden layers using stochastic gradient descent (SGD) optimizer is used to accomplish the classification. The model's accuracy was 44.6%, higher than that of the baseline model. A Weka-package Affective Tweets was applied for this model.

In TW-Star [15], a binary relevance (BR) multilabel classifier is used to classify tweets with SVM and TF-IDF features. Different preprocessing stop words removal, lemmatization (Lem), stemming (Stem), and common emoji recognition (Emo) have been tried. Different combinations of preprocessing have been tried out. The best accuracy results of 46.5% have been achieved by combining Emo, Stem, and Stop (Emo+Stem+Stop) preprocessing procedures.

The Random Forest (RF) method was employed [16]. The forest's decision trees varied from ten to one hundred, with each iteration increasing by ten. The algorithm's maximum tree depth ranged from 2 to 20, with each iteration increasing by a factor of one. Using the Doc2Vec model, feature vectors were generated. The Doc2Vec size ranged from ten to one thousand, with each iteration increasing by ten. The accuracy of this model was 25%.

III. METHODOLOGY

A. Experimental Setup

The work has been implemented on a Dell G5 15 laptop Intel i7 10th Gen, CPU 2.6 GHz with 6 GB GPU Nvidia Geoforce RTX2060. Libraries of Scikit learn 0.24.2, Genism 3.8.3, and Keras 2.3.0 libraries under the tensorflow2.1.0 platform in Python 3.6.13 have been used. The pre-trained word embedding model "Twitter-CBOW/tweets cbow 300" is loaded by gensim libraries in Python. Dataset is provided publicly by SemEval 2018 task1, the E-C subtask for the Arabic language [16]. This task has 2278 tweets for training, 585 tweets for development, and 1518 tweets for test data. We concatenated three datasets with a total size of 4381 tweets for the cross-validation process.

All experiments conducted using cross-validation have the number of splits k = 10. Each time one split is taken as a test, another nine splits are for training, and the number of repeats = 3, giving 30 trials in each experiment.

B. Preprocessing of Data:

Different preprocessing strategies have been created to fit the structure of social media colloquial text. The Twitter dataset was preprocessed as follows:

-The removal of the extension (Tatweel) from Arabic nouns like يكتبـــــون changed to يكتبون

- The removal of digits, special characters, non-Arabic letters as English characters (a–z A–Z) and French characters (àéèoeç)

- The removal of Arabic duplicate letters as كثيييير changed into كثير.

- Replacement of the mentions (@) into اسم_حساب_شخصي.

- stop words are eliminated from the text prior to training the model. Our Arabic stop word list is updated based on the NLTK Arabic stop word; Our revised list took into account the stop word change caused by the elimination of Hamza and Yaa as ('اليكم', 'اليكما', 'اليكن', 'اما', 'ان', 'انا', 'انت', 'انتم', 'انتما', 'انتن', 'انما', 'انه', 'انى'). The updated list also takes into account not to add some ambiguous words to this list as كان instead of كأن.

- Emojis are substituted with emotion-related meanings using the emoji lexicon. The emoji lexicon is a dictionary of the most popular Twitter emojis that is manually produced, with each emoji is first converted to its Arabic equivalent

- The Tweets are then stemmed using ARLSTEM stemmer [21]. Stemming is effective in tweets since they are largely dialectal Arabic and not in MSA form [17]. Prefixes at the start of words and suffixes at the end are removed. Stemming also entails changing the word's feminine to masculine and stemming the verb prefixes and suffixes. The stemmer also removed the hamza, changed alif maqsura to yaa, and trims waaw at the start of a word.

- some special words frequently repeated in tweets have been excluded from stemming the word "الله," and it is all derivations as "لله", "اللهم", "بالله", "فالله" are excluded from stemming with ARLSTEM. The frequency of occurrence of these special words is 16% out of the number of words in our dataset, so normalizing these words, not stemming them, has a good impact on performance as this reduces the ambiguity.

C. Feature extraction:

The AraVec embedding model [28] has been used for feature extraction. AraVec is a large-scale dataset (about 205,000 words) comprised of several Arabic dialects and trained on the Twitter data domain. Word embedding is crucial since it solves the sparsity issue in n-grams models and simplifies semantics by giving accurate representations for words that may appear in the same context.

Gensim libraries in Python were used to load the pre-trained model "Twitter-CBOW/tweets cbow 300." The word is represented as a 300-dimensional real number vector, and the tweet embeddings are computed by averaging the embeddings of all words in the tweet. The average embedding vector of each tweet is then sent to the classifier, which labels it as neutral or one of the 11 emotions (disgust, anger, fear, pessimism, anticipation, joy, love, optimism, sadness, trust, and surprise).

D. The Proposed classification model:

We built machine learning models using the SKLearn libraries for the multilabel emotion classification task. Five machine learning algorithms have been experimented with, including Ensemble Random Forest, Ensemble extra tree, KNN, SVM, and MLP.

1) Ensemble RF model

Random Forest and Extra Trees are two extremely comparable ensemble approaches. Both of them are made up of many decision trees, where the final decision is obtained considering every tree's prediction.

Random Forest Classifier has been implemented using Sklearn libraries. A random forest employs averaging to increase predicted accuracy and control over-fitting by fitting several decision tree classifiers on different sub-samples of the dataset.

The most parameters that affect the model performance are the max_depth of the tree, n_estimators, and crietion. Max_depth is the longest path between the root node and the leaf node. The nodes are expanded if max_depth is set to none until all leaves are pure. N_estimators is the number of trees in the forest. Crietion parameter measures the splitting quality; it is either "Gini" for the impurity or "Entropy" for the information gain. It is a tree-specific parameter.

The bootstrap parameter controls whether or not the max samples argument regulates the sub-sample size. The sub-sample size is regulated if bootstrap is True (default). Otherwise, each tree is created using the whole dataset.

The random state parameter controls the bootstrapping of the samples when building trees; it also controls the splitting at each node for the best sampling of features. A random state is better assigned to a fixed value.

2) Ensemble Extra Tree Model

According to the standard top-down approach, the Ensemble Extra-Trees technique creates an ensemble of unpruned classification or regression trees. There are two primary distinctions from previous tree-based ensemble approaches as RF. Ensemble Extra-Tree divides nodes randomly, and grows trees using the whole learning sample (instead of a bootstrap replica).

This model has been built using the sklearn Ensemble Extra Trees Classifier. This classifier performs a meta estimator that employs averaging to increase predicted accuracy and control over-fitting by fitting several randomized decision trees (a.k.a. extra-trees) on different sub-samples of the dataset [17]. It has the same parameters as the RF classifier, such as max depth of the tree, n estimators, crietion, random state, and other factors that assist regulate the tree's size, such as min samples leaf, which specifies the minimum amount of samples at a leaf node. The default settings for the parameters determining the size of the trees, such as max depth and min samples leaf, result in fully grown and unpruned trees, which for certain data sets can be enormous. To limit memory usage, such parameter values should be used to manage the complexity and size of the trees.

3) SVM model

The model used a Linear Support vector Classifier "Linear SVC", which is similar to a support vector classifier "SVC" but with a" linear" kernel parameter. LinearSVC is more flexible than SVC in choosing penalties and

loss functions. Also, LinearSVC better handles a large number of data samples. It treats both sparse and dense inputs. It handles multi-class using a one vs. rest scheme.

The most important parameters that affect the performance of LinearSVC are the classifier max_iteration parameter, the regularization parameter C, penalty, and the multi-class parameter.
-The strength of regularization is inversely proportional to C. C must be strictly positive.
-Penalty specifies the norm used in the penalization. The L2 penalty is standardly used in SVC.
-The multi_class parameter determines the multi-class strategy. For the output of more than two classes, the multi-class parameter "ovr" trains n classes one vs. rest. The multi-class parameter "crammer_singer" optimizes a joint objective over all classes. However, it is rarely used in practice as it is more expensive to compute and sometimes leads to worse accuracy results.

4) MLP model

A multilayer perceptron (MLP) is a feed forward artificial neural network (ANN). There are at least three layers of nodes in an MLP: an input layer, a hidden layer, and an output layer. Each node is a neuron with a nonlinear activation function except for the input nodes. Our MLP model has been implemented using 'MLP Classifier'; the classifier has a regularization term added to the loss function to avoid overfitting by shrinking model parameters. This model has optimized the log loss function using stochastic gradient descent [18]. Using either SGD or ADM' optimizer. 'Adam' optimizes well on relatively large datasets of thousands or more training samples. A maximum number of iterations parameter determines the number of iterations the optimizer iterates until convergence. This number of iterations determines the number of epochs (how many times each data point will be used), not the number of gradient steps. Tolerance allowed for the optimization is adjusted using the tol parameter. When the loss or score does not improve by at least tol, convergence is attained, and training is terminated.

5) Multiple KNN model

KNN is one of the simplest supervised machine learning algorithms to learn. Class membership is the result of k-NN categorization. A majority vote of its neighbours' classifies an object. The object is assigned to the class with the most members among its k closest neighbors [19], where k is a small positive integer. This model tried a "KNeighborsClassifier" the most common parameters that affect the classifier are the number of neighbors to use K; the algorithm used to compute the nearest neighbor. There are three algorithms to select from BallTree, KDTree, and brute-force search to decide which algorithms use the fit method; and the weight function that assigns a weight to each neighbour.

E. Evaluation Metrics:

The proposed model has been evaluated using the following metrics1.

- Precision: $$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \quad (1)$$

Recall R (true Positive Rate TPR or sensitivity): $R = \frac{TP}{P} = TP/(TP + +FN)$ (2)

F1 score is the harmonic mean of precision and recall $F1 = 2 * (P * R)/(P + R)$ (3)

- Multilabel accuracy (or Jaccard index): It is computed for each tweet t and averaged over all tweets in the dataset T:

$$\text{Accuracy} = \frac{1}{|\mathrm{T}|}\sum_{\mathrm{t}\in\mathrm{T}}\frac{\mathrm{Gt}\cap\mathrm{Pt}}{\mathrm{Gt}\cup\mathrm{Pt}} \quad (4)$$

Where:

TP (True Positive) is the number of genuine or true positive outputs.

FP (False Positive) is the number of incorrectly recognized outputs as positive.

FN (False Negative) is the number of incorrectly recognized outputs as negative.

[1] https://competitions.codalab.org/competitions/17751

TP+FP is the total number of positive outputs (including incorrectly recognized ones).

TP+FN is the number of samples that should have been identified as positive.

Pt is the set of the predicted labels for tweet t, Gt is the set of the actual (gold) labels for tweet t, and T is the set of tweets.

When the classifications are multilabel, averaged metrics vary from total accuracy.

- In Macro Average, the metrics are generated individually for each class. The average is then determined (all classes are treated equally).
- Micro-average computes the average metric by combining the contributions of all classes. In a multi-class classification system, micro-average is favoured, particularly in imbalance class scenarios (i.e., having more examples of one class than other classes).
- Weighted average calculates the metrics for each label and gets their average weighted by support using a weighted average (the number of instances for each label); this modifies macro to compensate for the label imbalance.
- Samples average compute each instance's metrics and determine the average (only meaningful for multilabel classification

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

IV. RESULTS AND DISCUSSION

A. Ensemble RF model:

The Random Forest model has been tried using the ensemble Random Forest classifier library both with the default features values and the optimized ones until the best accuracy values change, as shown in Table 1 below.

The best results were obtained when increasing number of estimators from the default of 100 to 200, Entropy instead of Gini (default), setting the random state to 10, with Min_samples_leaf =2,and adjusting the max depth to 50. However, enhancing the performance by about 1% in Jaccard accuracy takes much more time in running the model due to increasing the number of estimators and max depth of the tree.

Choosing a large number of estimators in a random forest model is not the best idea because increasing the number of estimators beyond a certain value increases the model complexity.

TABLE 1 ENSEMBLE RF MODEL RESULTS

Ensemble Random Forest			
Features n_estimators, max_depth, random_state			
Best Results		**Default**	
n_estimators	200	**n_estimators**	100
Criterion	Entropy	**Criterion**	Gini
Max_depth	50	**Max_depth**	None
Random_state	10	**Random_state**	None
min_samples_leaf	2	**min_samples_leaf**	1
Results			
micro_precision:	0.799	**micro_precision:**	0.80
micro_recall	0.282	**micro_recall**	0.269
Micro_F1	0.417	**Micro_F1**	0.409
samples_multilabel_accuracy	0.291	**samples_multilabel_accuracy**	0.278
Time in minutes	29.8	**Time in minutes**	3.96

B. Ensemble Extra Tree Model:

The ensemble extra tree model has been tried with its default parameters in Sklearn and adjustable values. The results are reported below in table2. The same parameters are used as in the RF model.

The best results were obtained when increasing number of estimators from the default of 100 to 200, Entropy instead of Gini(default, setting the random state to fixed value (10), and adjusting the max depth to 50 with Min_samples_leaf =2. Random Forest gets better results than Extra Trees using the values of the same hyperparameters. Extra Trees is much faster because it splits the tree randomly instead of looking for the optimal split at each node.

TABLE 2 ACCURACY RESULTS USING ENSEMBLE EXTRA TREE MODEL

Ensemble Extra Tree			
Features n_estimators, max_depth, random state, Criterion, min_samples_leaf			
Default		**Best results**	
n_estimators	200	**n_estimators**	100
Criterion	Entropy	**Criterion**	Gini
Max_depth	50	**Max_depth**	None
Random state	10	**Random state**	None
min_samples_leaf	2	**min_samples_leaf**	1
Results			
micro_precision:	0.81	**micro_precision:**	0.812
micro_recall	0.25	**micro_recall**	0.246
micro_F1	0.381	**micro_F1**	0.377
samples_multilabel_accuracy	0.262	**samples_multilabel_accuracy**	0.257
Time in minutes	5.68	**Time in minutes**	3.19

C. Multilabel KNN model ML-KNN:

A Binary Relevance multilabel classifier has been implemented. It used the k-Nearest Neighbors method with cross-validation. The best performance obtained for the k=5 has been reported in table 3. It is a very simple, very fast algorithm with better performance than decision tree algorithms but still does not give satisfactory accuracy results.

TABLE 3 ACCURACY RESULTS USING MULTILABEL KNN MODEL

ML-KNN	
KNN Feature	K=5
Cross validation	n_splits = 10, n_repeats = 3
micro_precision	0.587
micro_recall	0.419
micro_F1	0.489
samples_multilabel_accuracy	0.363
elapsed_time in minutes	0.86

D. SVM Model:

Different experiments have been implemented using a Linear SVC classifier.

Different values of the max_iteration parameter have been tried; the best performance was obtained using max_iteration =5000. The regularization parameter C takes a positive value, and a penalty is "L2".

The effect of changing regularization parameter C has been reported below in table 4; the best results are obtained with C= 0.1. The regularization strength is inversely proportional to C, but decreasing C beyond this value leads to performance degradation as the C value is strictly positive. C=0.1 gives the best Jaccard accuracy with the least running time.

TABLE 4 RESULTS OF SVM MODEL WITH DIFFERENT VALUES OF REGULARIZATION PARAMETER C

LinearSVC							
cross validation :10 splits, 3 repeats							
Parameters							
max_iter = 5000							
Multi_class="ovr"							
C	0.001	0.05	0.1	0.5	1	5	10
all other parameters are set to default : penality='L2', Loss=" Squared Hing", tol =1e-4							
Results							
Micro_Pe	0.779	0.717	0.71	0.697	0.693	0.687	0.684
Micro_Re	0.404	0.507	0.512	0.515	0.515	0.516	0.518
Micro_F1	0.531	0.594	**0.594**	0.592	0.591	0.589	0.589
Samples_jaccard	0.40	0.462	**0.463**	0.461	0.46	0.459	0.46
Elapsed time	0.47	1.06	**2.13**	8.14	12.85	12.70	13.03

Also, the effect of using the multi-class parameter "Crammer_singer" compared with the "ovr" has been tried and shown in Table 5 below. Although "Crammer_singer" is more optimized for multi-class case, "OVR" resulted in better accuracy with less running time due to the complexity of "Crammer_singer".

TABLE 5 SVM EXPERIMENTS WITH DIFFERENT MULTI-CLASS FEATURES

LinearSVC		
cross validation :10 splits, 3 repeats		
Features Max_iter=5000, C=0.1, all other parameters are set to default		
Multi_class param	**Crammer_singer**	**Ovr**
Results		
micro_precision	0.73	0.71
micro_recall	0.49	0.512
micro_F1	0.586	**0.594**
samples_multilabel_accuracy	0.46	**0.463**
elapsed_time in minutes	11.3	**2.13**

E. MLP model:

Using the MLP libraries in SKLearn, with cross-validation of 10 splits and three repeats

The initial learning rate of value 0.001 was used for controlling the step size in updating the weights. Activation function for the hidden layer has been set to 'relu', the rectified linear unit function, returns f(x) = max (0, x). This classifier works with floating-point data used by NumPy arrays or sparse Scipy arrays. The best results were obtained with a max_iter value of 200 and feature tol=0.03. The best results obtained are F1=0.606 and samples-Jaccard =0.48, as shown in table 6. When using smaller values of max_iteratin, the results have not been converged to an optimized solution.

TABLE 6 ACCURACY RESULTS USING MLP MODEL

MLP						
cross validation :10 splits,3 repeats						
Features tol =0.03,						
Max_iter	20	50	100	200	500	1000
Results						
Micro precision	0.73	0.705	0.704	0.708	0.702	0.707
Micro recall	0.507	0.526	0.528	0.532	0.533	0.532
Micro F1	0.598	0.602	0.603	**0.606**	0.605	0.607
samples_Jaccard	0.469	0.476	0.476	**0.48**	0.477	0.479
elapsed_time in minutes	0.505	0.76	0.878	**0.82**	0.849	0.787

The best results from different machine learning are summarized below in Table 7

TABLE 7 SUMMARIZED RESULTS OF PROPOSED MODEL WITH DIFFERENT MACHINE LEARNING

Model	**Results**				
	Micro_P	Micro_R	Micro_F1	Samples Jaccard	Time
Ensemble RF	0.799	0.282	0.417	0.291	29.8
Ensemble Extra Trees	**0.81**	0.25	0.381	0.262	5.68
ML- KNN	0.595	0.43	0.499	0.375	0.92
SVM	0.71	0.512	0.594	0.463	2.13
MLP	0.708	**0.532**	**0.606**	**0.48**	**0.82**

F. Model Performance comparison with previous models

A comparison between the proposed model and previous models that studied the same task with the same dataset is performed in Tables 8,9,10.

TABLE 8 PROPOSED LINEAR SVC MODEL COMPARED WITH OTHER PREVIOUS LINEAR SVC MODELS

Model	Preprocessing	Features	Samples_Jaccard Accuracy	Micro F1
EMA	Normalization, a manual emoji lexicon + ARLSTEM	AraVec	0.489	0.618
TW-Star	Emo + Stem+ stop	TF-IDF	0.465	0.597
Proposed SVM	Normalization, a manual emoji lexicon + ARLSTEM	AraVec	0.463	0.594
SVM-Unigram	NA	Unigrams	0.38	0.516

In Table 8, the best model based on SVM is the EMA which achieved an accuracy of 0.489 and micro F1 of 0.618. The key difference between EMA and our proposed one is that the preprocessing steps in EMA include more normalization steps followed by [20] and the tuning of parameters.

TABLE 9 PROPOSED MLP MODEL COMPARED WITH OTHER PREVIOUS FCNN MODELS

Model	Features	Classification algorithm	Samples_Jaccard Accuracy	Micro F1
UNCC	AraVec	FCNN	0.446	0.572
Proposed MLP	AraVec	MLP	**0.48**	**0.606**

In table 9, the proposed MLP outperforms the FCNN model; although both use the same features, the more hidden layers in the MLP enhance the Jaccard accuracy results.

TABLE 10 PROPOSED RF, EXTRA TREE MODELS COMPARED WITH OTHER PREVIOUS RF MODEL

Model	Features	Classification algorithm	Samples_Jaccard Accuracy	Micro F1
proposed Extra tree	AraVec	Extra Tree	0.262	0.381
proposed RF	AraVec	RF	**0.291**	**0.417**
Amrita	Doc2Vec	RF	0.254	0.379

In table 10, the proposed ensemble decision tree models RF and extra trees beat the performance of the RF model used by the Amirta team. The main difference is that our models use the Aravec word embedding for feature selection, whereas the Amrita model uses Doc2vec embedding.

V. CONCLUSIONS

This work built a framework for multilabel emotion classification for Arabic tweets using traditional machine learning techniques. MLP achieved the best performance. Although the SVM model is less accurate, it is simpler and memory efficient. MLP is superior in time compared with other machine learning models except for ML-KNN, which is so simple and fast model but with low performance in this multilabel task. The decision tree algorithms include the Ensemble RF; the Extra tree model proved unsuitable for this multilabel classification task. Ensemble RF is a low-performance model and very slow. The Extra trees model is much faster than ensemble RF but failed to achieve satisfying performance.

The proposed framework with different machine learning algorithms has also been compared with previous models using the same machine learning algorithms. In most cases, our proposed framework outperforms the corresponding previous work for the same task using the same classification algorithm except for the SVM case; the previous EMA framework has more accurate results than that of our proposed LinearSVC.

REFERENCES

[1] M. Hatamian, J. Serna, and K. Rannenberg, "Revealing the unrevealed: Mining smartphone users privacy perception on app markets," *Computers & Security*, vol. 83, pp. 332-353, 2019.

[2] H. Wang, S. Lu, and J. Zhao, "Aggregating multiple types of complex data in stock market prediction: A model-independent framework," *Knowledge-Based Systems*, vol. 164, pp. 193-204, 2019.

[3] R. Baly *et al*., "Comparative evaluation of sentiment analysis methods across Arabic dialects," *Procedia Computer Science*, vol. 117, pp. 266-273, 2017.

[4] R. Barbado, O. Araque, and C. A. Iglesias, "A framework for fake review detection in online consumer electronics retailers," *Information Processing & Management*, vol. 56, no. 4, pp. 1234-1244, 2019.

[5] A. Abdi, S. M. Shamsuddin, S. Hasan, and J. Piran, "Machine learning-based multi-documents sentiment-oriented summarization using linguistic treatment," *Expert Systems with Applications*, vol. 109, pp. 66-85, 2018.

[6] E. Haihong, H. Yingxi, P. Haipeng, Z. Wen, X. Siqi, and N. Peiqing, "Theme and sentiment analysis model of public opinion dissemination based on generative adversarial network," *Chaos, Solitons & Fractals*, vol. 121, pp. 160-167, 2019.

[7] A. Ruiz-Garcia, M. Elshaw, A. Altahhan, and V. Palade, "A hybrid deep learning neural approach for emotion recognition from facial expressions for socially assistive robots," *Neural Computing and Applications*, vol. 29, no. 7, pp. 359-373, 2018/04/01 2018, doi: 10.1007/s00521-018-3358-8.

[8] A. S. Patwardhan and G. M. Knapp, "Multimodal affect analysis for product feedback assessment," *arXiv preprint arXiv:1705.02694*, 2017.

[9] M. Hasan, E. Rundensteiner, and E. Agu, "Automatic emotion detection in text streams by analyzing twitter data," *International Journal of Data Science and Analytics*, vol. 7, no. 1, pp. 35-51, 2019.

[10] T. Sharma, M. Diwakar, P. Singh, S. Lamba, P. Kumar, and K. Joshi, "Emotion Analysis for predicting the emotion labels using Machine Learning approaches," in *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, 2021: IEEE, pp. 1-6.

[11] A. Omar, T. M. Mahmoud, T. Abd-El-Hafeez, and A. Mahfouz, "Multi-label arabic text classification in online social networks," *Information Systems*, vol. 100, p. 101785, 2021.

[12] N. Aljedani, R. Alotaibi, and M. Taileb, "Hmatc: Hierarchical multi-label arabic text classification model using machine learning," *Egyptian Informatics Journal*, vol. 22, no. 3, pp. 225-237, 2021.

[13] G. Badaro *et al.*, "Ema at semeval-2018 task 1: Emotion mining for arabic," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, 2018, pp. 236-244.

[14] M. Abdullah and S. Shaikh, "Teamuncc at semeval-2018 task 1: Emotion detection in english and arabic tweets using deep learning," in *Proceedings of the 12th international workshop on semantic evaluation*, 2018, pp. 350-357.

[15] H. Mulki, C. B. Ali, H. Haddad, and I. Babaoğlu, "Tw-star at semeval-2018 task 1: Preprocessing impact on multi-label emotion classification," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, 2018, pp. 167-171.

[16] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, "Semeval-2018 task 1: Affect in tweets," in *Proceedings of the 12th international workshop on semantic evaluation*, 2018, pp. 1-17.

[17] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3-42, 2006/04/01 2006, doi: 10.1007/s10994-006-6226-1.

[18] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010: JMLR Workshop and Conference Proceedings, pp. 249-256.

[19] S. Patwardhan, "Simple understanding and implementation of KNN algorithm!," ed, April 21, 2021.

[20] A. Shoukry and A. Rafea, "Preprocessing Egyptian Dialect Tweets for Sentiment Mining," in *AMTA*, 2012.

