Detecting Signs of Depression in Social Networks Users: A Framework for Enhancing the Quality of Machine Learning Models

Abir Gorrab, Nourhène Ben Rabah, Bénédicte Le Grand, Rébecca Deneckère, Thomas Bonnerot

Detecting Signs of Depression in Social Networks Users: A Framework for Enhancing the Quality of Machine Learning Models

Abir Gorrab[1], Nourhène Ben Rabah[2], Bénédicte Le Grand[2], Rébecca Deneckère[2], Thomas Bonnerot[2]
[1] RIADI laboratory, National School of computer Science, University of Manouba, Tunisia
Abir.Gorrab@riadi.rnu.tn
[2] Centre de Recherche en Informatique, Université Paris1 Panthéon-Sorbonne, France
Nourhene.Ben-Rabah | Benedicte.Le-Grand | Rebecca.Deneckere @univ-paris1.fr

Abstract. Depression is widely recognized as a major contributor to global disability and a significant factor in the emergence of suicidal tendencies. On social networks, individuals openly share their thoughts and emotions through posts, comments, and other forms of communication. The use of Artificial Intelligence, particularly Machine Learning methods, holds great potential for analyzing this data. However, it is imperative to exercise caution in the application of these methods to avoid biases and overfitting, two problems that could compromise the quality of Machine learning models. In this paper, we present a framework for detecting signs of depression among users of the X social network. This framework is based on four phases aimed at minimizing both biases and overfitting, resulting in models that generalize well to new data, thereby enhancing their applicability by healthcare professionals and patients. To validate our framework, we present the results of three detailed experiments using nine Machine Learning algorithms.

1 Introduction

Depression is a serious issue that profoundly affects mental health [30]. According to the French National Institute of Health and Medical Research (INSERM), it is defined as a major personal suffering that can lead to chronic illnesses, health problems, and, in the most severe cases, the risk of suicide. The World Health Organization (WHO) estimates that 3.8% of the global population suffers from depression [28], with rates of 5% among adults and 5.7% among individuals over 60 years old. Overall, approximately 280 million people are affected by depression [12].

The detection of depression through the exploration of social networks represents a continuously expanding research field [8, 2, 14, 7, 31, 25]. This approach is justified by the fact that people spend a lot of time on social networks, where they openly share their thoughts, emotions, and experiences [30]. The analysis of these online messages can be used to detect potential signals associated with depression [2, 23]. Researchers are exploring various approaches by combining Artificial Intelligence (AI) techniques, such as Machine Learning (ML) [7, 21, 13], Natural Language Processing (NLP)[14], and sentiment analysis methods [16, 18], to identify indicators of depression within social media posts.

However, due to the interdisciplinary nature of this issue, not all disciplines possess the same level of expertise, and their focus is not uniformly directed toward the challenges inherent in the generalization of machine learning algorithms. Moreover, Machine Learning should be used with care, as each algorithm may introduce potential biases and compromise the validity of results, thereby limiting its applicability by healthcare professionals and patients.

In this context, we propose a depression detection framework based on the analysis of users' tweets on the X social network. We start (1) with the data collection and annotation phase where we collect data and label it using Valence Aware Dictionary and sEntiment Reasoner (VADER). We then present (2) a data pre-processing phase consisting of (2.1) cleaning data by removing stopwords, data analyzing through tokenization and lemmatization; and data modeling with N-gram, to finally (2.2) encode data through BoW and TF-IDF methods. We then (3) train nine ML models, with a crucial phase of hyperparameter tuning, to end (4) with the model evaluation phase.

This paper presents several contributions:

- A comprehensive framework in which each step of the four phases is explained in detail, aiming to minimize both overfitting and bias issues. To our knowledge, no detailed study on depression detection while minimizing overfitting and bias has been conducted so far (see Related Works section). Moreover, our framework may also be useful for researchers in the healthcare area, who are not necessarily experts in Machine Learning.
- Detailed experimental results obtained from real data, comparing the performance of nine ML algorithms.

The remainder of this paper is organized as follows: Section 2 describes the proposed framework, while Section 3 presents three experiments to validate it. In Section 4, we discuss related works, before presenting our conclusions in Section 5.

2 Proposed Framework

The proposed framework relies on four key phases: (i) data collection and annotation, (ii) data pre-processing, (iii) hyperparameter tuning and model training, and (iv) model evaluation. Fig.1 describes these phases.

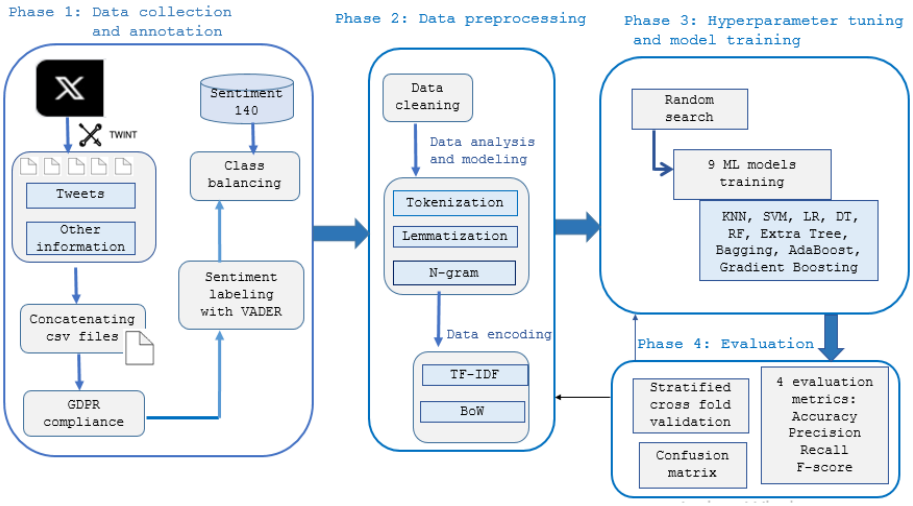


Fig. 1. Depression detection proposed framework.

2.1 Phase 1: Data collection and annotation

In this first phase, our objective is to collect various data describing tweets from users in the X social network. The main aim of this collection is to obtain a balanced representation of *alarming messages*, i.e. reflecting a depressive state, and *normal messages*, i.e. containing no sign of depression. To achieve this objective, we build our database, following the steps described below:

- **Step 1: Data Collection with Twint.** We use Twint, a web scraping tool for tweets on X, to search for tweets containing one of these five specific keywords: 'depressed', 'depressive', 'hopeless', 'lonely', and 'suicide'. Each keyword-based query generates a distinct CSV file, which is concatenated with the others into a single one. The resulting file consists of 25 columns, including 'conversation_id', 'created at', 'date', 'time', 'timezone', 'user-id', 'user_name', 'name', 'place', 'tweet', 'mentions', 'urls', 'photos', 'replies_count', 'retweets_count', 'likes_count', 'hashtags', 'cashtags', 'links', 'retweet', 'quote_url', 'video', 'user_rt_id', 'near' and 'geo'.

- **Step 2: GDPR Compliance and removal of sensitive data.** Respecting GDPR rules is essential when dealing with personal data. To ensure confidentiality and privacy protection, sensitive columns such as 'user_name', 'name', 'place', 'near', 'geo', and 'user_rt_id' are removed. This deletion is a key measure to avoid any privacy violations. Non-exploitable data, typically null values, are also eliminated, leaving only 'user_id' and tweets. Moreover, only tweets in English are retained.

- **Step 3: Application of VADER for sentiment labeling.** VADER (Valence Aware Dictionary and sEntiment Reasoner) is used to label tweets based on sentiments.

Each tweet can contain positive, negative, and neutral elements. VADER aggregates these elements to calculate an overall score that reflects the overall sentiment of the tweet. From VADER scores which range from -1 (very negative) to 1 (very positive), we assign *class 0* to negative score values, corresponding to 'alarming tweets' and *class 1* to positive score values – normal tweets. As a result, our dataset comprises 52,139 alarming tweets and 17,821 normal tweets.

- **Step 4: Class balancing.** Due to the imbalance between both classes and to prevent overfitting, we decided to retain 30,000 alarming tweets and we supplemented the 17,821 tweets with positive VADER scores (normal tweets) by adding positive tweets from an existing dataset called Sentiment140. Sentiment140 [9] is a dataset containing tweets labeled with polarity (negative, neutral, positive). From this dataset, we selected positively polarized tweets containing other depressive keywords than those used in our initial dataset ('depressed', 'depressive', 'hopeless', 'lonely', and 'suicide'), in order to add diversity in the dataset.

Our final dataset consists of three columns: 'user_id', 'tweet', and 'label'. It includes 30,000 normal tweets containing terms related to depression and positive sentiment (class 1) and 30,000 alarming tweets with depressive words and negative sentiment (class 0).

2.3 Phase 2: Data pre-processing

- **Step 1: Data cleaning:** Stop words refer to common words in a language that lack substantial meaning. To clean data, we remove stop words using the NLTK library (Natural Language Toolkit) [19]. We then eliminate unnecessary characters such as non-alphanumeric characters using Regex. We also remove short lines (< 2 characters), to finally keep 29,671 alarming tweets and 29,238 normal ones.

- **Step 2: Data analysis and modeling.** We proceed with a textual analysis composed of two stages: **tokenization** [26] and **lemmatization**. Tokenization consists of dividing a text into a collection of individual words named tokens. In the second step, lemmatization reduces words to their base or root form, known as the lemma. This allows us to consider the different forms of a term as a unique concept (eg. 'depressed', 'depression', 'depressing'). We then split our lemmas into uni-grams using N-gram modeling [17]. An N-gram is a sequence of N elements in data, such as characters or words, in the context of Natural Language Processing.

  Step 3: Data encoding: When dealing with textual data, it is necessary to encode it into numerical format for ML algorithms to function properly. For this, we use two encoding methods such as **TF-IDF** (Term Frequency-Inverse Document Frequency) and **BoW** (Bag of Words). TF-IDF is a term weighting measure in a document, text, or other content**.** A term that appears frequently in a document

but rarely in the entire set of documents will have a high TF-IDF score, indicating its relative importance in the specific document. In our context, a term is a word, and a document is a tweet. Therefore, we calculate the score of each word in a tweet relative to its frequency in the corresponding tweet and in the overall set of tweets. The Bag of Words (BoW) model transforms arbitrary text into fixed-length vectors by counting the number of occurrences of each word.

2.4 Phase 3: Hyperparameter tuning and model training

In this phase, we train and test nine ML models for depression detection from X data. These models are the k-nearest neighbors (**KNN**), support vector machine (**SVM**), logistic regression (**LR**), Decision Tree (**DT**), Random Forest (**RF**), Extra Tree (**ET**), **Bagging**, **AdaBoost**, and **Gradient Boosting**. In this study, we deliberately choose not to use deep learning algorithms and start with simpler models, thus facilitating the interpretation of results. Additionally, the selected algorithms do not require the access to graphics processing units (GPUs), unlike deep learning models for an efficient training. We use the implementations of these algorithms provided by the Scikit-learn library. To reduce the risk of bias when evaluating model performance, we use stratified 10-fold cross-validation; this method divides the dataset into 10 folds with the same distribution of classes, thus ensuring that the relative frequencies of the classes are approximately preserved in each training and test set.

Before training these models, it is important to select the right values of hyperparameters as they have a significant impact on the performance of machine learning models. Optimal hyperparameters values contribute to preventing overfitting and enable the models to generalize correctly to new data. For time efficiency purposes, we apply an automatic method called Random Search, which randomly selects combinations of hyperparameters.

2.5 Phase 4: Model Evaluation

The performance of each model is measured using the following performance metrics: accuracy, precision, recall, and F1-score. These metrics are calculated from four values represented in a confusion matrix as shown in Table 1:

- True Positive (TP) refers to alarming tweets that are correctly predicted.
- True Negative (TN) refers to normal tweets that are correctly predicted.
- False Negative (FN) refers to alarming tweets that are predicted as normal tweets.
- False Positive (FP) refers to normal tweets that are incorrectly predicted as alarming tweets.

Table 1. Confusion matrix.

	Predicted Alarming tweets	Predicted Normal tweets
Actual Alarming tweets	TP	FN
Actual Normal tweets	FP	TN

In the following, we present the evaluation metrics:

- **Accuracy** represents the proportion of correct predictions:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (1)$$

- **Precision** is the proportion of actual alarming tweets among the tweets predicted as alarming. A low precision means that a high proportion of normal tweets are detected as alarming tweets (false positive).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

- **Recall** is the proportion of actual alarming tweets with regard to the actual number of alarming tweets. A low recall indicates that a large proportion of alarming tweets have been classified as normal (false negative).

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

- **F_score** is the harmonic mean of precision and recall values, reaching its best value at 1 and its worst value at 0. It is calculated as follows:

$$F\_score = \frac{2 * Recall * Precision}{Recall + Precision} \quad (4)$$

3 Experiments

This section outlines three experiments to validate our proposed framework. In the initial experiment (section 3.1), we compare tweet encoding methods, namely BoW and TF-IDF, by presenting the top 30 and 100 most frequent unigrams for each method. The second experiment (3.2) employs the Random Search method to identify optimal hyperparameter values associated with the nine selected ML algorithms: Random Forest, ExtraTree, KNN, SVM, Logistic Regression, Decision Tree, AdaBoost, GradientBoosting, and Bagging. These adjustments are executed using both TF-IDF and BoW. The Third experiment (3.3) trains, tests and evaluates the different models with these optimized hyperparameters for both TF-IDF and BoW.

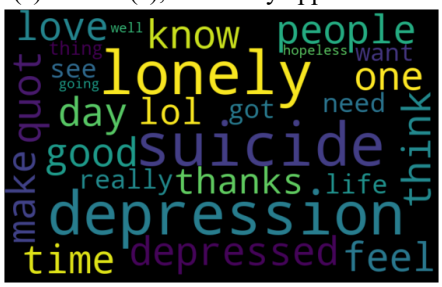
3.1. The most frequent unigrams in our dataset

To compare the tweet encoding methods, namely BoW and TF-IDF, we selected the top 30 and 100 unigrams for each method. In Fig.2, (a) and (b) illustrate word clouds corresponding to the top 30 and 100 unigrams (most frequent terms) in the dataset using the BoW method. Similarly, (c) and (d) represent word clouds of the top 30 and 100 unigrams using the TF-IDF method. These visualizations allow for a visual comparison of the most important terms according to each method. These terms are displayed with a larger font size.

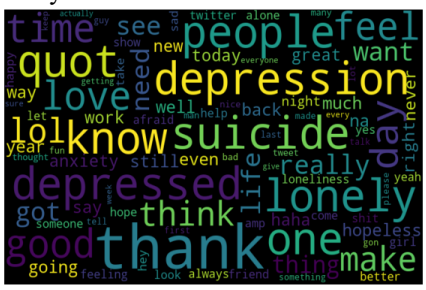
The BoW word cloud highlights the frequently used terms in our dataset. Looking at (a) and (b), we can see terms that describe a depressive state, such as “lonely”, “suicide”, “depression”, “depressed”, “hopeless”, as well as terms associated with verbs used to express feelings or needs, such as “need”, “want”, “know”, “make”, “feel”. It

also includes normal terms that don't express depression, such as "love", "people", "lol", "good", and "better".

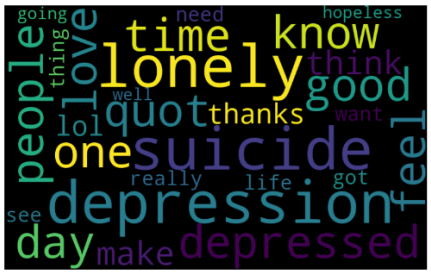
Looking at (c) and (d), we essentially see the same terms, but we note a difference in the frequency of these terms. We can see that the terms "people" and "know" are larger in (c) than in (a), since they appear more frequently in the dataset.



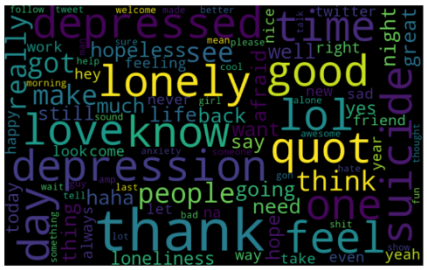
(a) The word clouds for the top 30 unigrams using the **BoW** method.



(b) The word clouds for the top 100 unigrams using the **BoW** method.



(c) The word clouds for the top 30 unigrams using the **TF-IDF** method



(d) The word clouds for the top 100 unigrams using the **TF-IDF** method.

Fig. 2. Word clouds for top 30 and 100 unigrams using **BoW** and **TF-IDF** methods.

3.2 Hyperparameter tuning using Random Search

A hyperparameter is a configuration variable of the machine learning algorithm and does not depend on the specifically trained model. The meaning of each hyperparameter is provided in the documentation of scikit-learn [22]. Tuning the hyperparameters of an algorithm allows to find the optimum values for a given set of data. Machine learning algorithms have several hyperparameters that can influence their performance. The hyperparameters of the nine algorithms we are comparing are shown in Table 2. It is generally accepted that these algorithms perform reasonably well with the default values of the hyperparameters specified in the software packages. However, adjusting the hyperparameters can improve their performance. One approach to choosing an optimal combination of values for our hyperparameters is to build a model for each possible combination of hyperparameter values. This method can be expensive and slow. To overcome these limitations, we have opted for a Random Search method for

both TF-IDF and BoW. Table 2 shows the optimal hyperparameters for both methods. We may note that the encoding method impacts the optimal values of certain hyperparameters (represented in bold and italics).

Table 2. Hyperparameter tuning.

Algorithms	Hyperparameters with their possible values	Hyperparameters optimized for TF_IDF	Hyperparameters optimized for BoW
KNN	n_neighbors: [3, 5, 7, 9]	n_neighbors= 9	n_neighbors= 9
	weights: [uniform, distance]	weights=distance	weights=distance
SVM	C : [0.1, 1, 10]	***C=1***	***C=0.1***
	Kernel : [linear, rbf, poly]	***Kernel=rbf***	***Kernel=linear***
	gamma: [scale, auto, 0.1, 1, 10]	***Gamma=1***	***Gamma=10***
LR	Penalty : [l1, l2]	Penalty=l2	Penalty=l2
	C : [0.1, 1, 10]	***C=10***	***C=1***
DT	max_depth: [None, 10, 20, 30]	max_depth=30	max_depth=30
	min_samples_split: [2, 5, 10]	***min_samples_split=10***	***min_samples_split=5***
	min_samples_leaf: [1, 2, 4]	***min_samples_leaf =4***	*min_samples_leaf =2*
RF	n_estimators: [50, 100, 200]	n_estimators= 50	n_estimators= 50
	max_depth: [None, 10, 20, 30]	max_depth: None	max_depth: None
	min_samples_split: [2, 5, 10]	min_samples_split=5	min_samples_split=5
	min_samples_leaf: [1, 2, 4]	min_samples_leaf=1	min_samples_leaf=1
	bootstrap: [True, False]	bootstrap= False	bootstrap= False
ET	n_estimators: [50, 100, 200]	n_estimators= 50	n_estimators= 50
	max_depth: [None, 10, 20, 30]	max_depth: None	max_depth: None
	min_samples_split: [2, 5, 10]	min_samples_split=5	min_samples_split=5
	min_samples_leaf: [1, 2, 4]	min_samples_leaf=1	min_samples_leaf=1
	bootstrap: [True, False]	bootstrap= False	bootstrap= False
Bagging	base_estimator: RF	base_estimator: RF	base_estimator: RF
	n_estimators: [10, 20, 30]	n_estimators=10	n_estimators=10
	max_samples: [0.5, 0.7, 1.0]	max_samples=0.7	max_samples=0.7
	max_features: [0.5, 0.7, 1.0]	max_features=1	max_features=1
Ada-Boost	base_estimator: RF	base_estimator: RF	base_estimator: RF
	n_estimators: [1, 5]	***n_estimators: 1***	***n_estimators: 5***
Gradient-Boosting	n_estimators: [1,5]	n_estimators= 1	n_estimators= 1
	learning_rate: [0.01, 0.1, 0.2]	***learning_rate=0.01***	***learning_rate=0.2***
	max_depth: [3, 5, 7]	max_depth=7	max_depth=7
	min_samples_split: [2, 5, 10]	min_samples_split=2	min_samples_split=2
	min_samples_leaf: [1, 2, 4]	min_samples_leaf=1	min_samples_leaf=1

3.3 Models training, testing and evaluation for TF-IDF and BoW encodings

Table 3 and 4 show the performances of the nine ML models trained and tested with BoW and TF-IDF encoding methods respectively. As shown in Table 3, for BoW encoding, LR model gives the highest accuracy in the test set (95.48%), followed by SVM (95.12%), AdaBoost (95.04%) and DT (94.94%). For TF-IDF encoding, LR also presents the highest accuracy in the test set (95.52%). We then find SVM (95.41%), AdaBoost (95.34%) and then RF (95.09%).

We note that no overfitting occurs for LR and SVM, as the accuracy on the test set does not decrease significantly as compared to the accuracy on the training set. Moreover, we observe the TF-IDF encoding method provides better results than BoW.

Table 3. Performances of ML models trained and tested on BoW encoding.

	Training set				Test set			
	Precision	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**	**Accuracy**
KNN	99.97	99.70	99.84	99.84	96.28	85.02	90.30	90.94
SVM	99.87	91.29	95.38	95.61	99.72	90.42	94.84	95.12
LR	99.79	94.70	97.18	97.27	98.85	91.97	95.28	**95.48**
DT	99.82	90.39	94.87	95.15	99.49	90.27	94.65	94.94
RF	99.94	99.48	99.71	99.71	96.36	93.15	94.73	94.86
ET	99.95	99.56	99.75	99.75	95.10	93.51	94.30	94.39
Bagging	99.12	98.50	99.11	99.74	93.25	92.69	92.97	93.05
AdaBoost	99.93	99.74	99.84	99.84	97.11	92.78	94.89	95.04
Gradient-Boosting	100	89.03	94.19	94.55	99.98	89.03	94.18	94.54

Table 4. Performances of ML models trained and tested on TF-IDF encoding.

	Training set				Test set			
	Precision	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**	**Accuracy**
KNN	99.97	99.70	99.83	99.84	86.06	77.69	81.67	82.70
SVM	99.95	97.60	98.76	98.78	99.61	91.11	95.17	95.41
LR	99.67	97.81	98.73	98.75	98.14	92.73	95.36	**95.52**
DT	99.79	90.35	94.83	95.12	99.58	90.07	94.58	94.88
RF	99.90	99.75	99.83	99.83	97.01	92.96	94.94	95.09
ET	99.95	99.70	99.82	99.83	96.06	93.44	94.74	94.85
Bagging	99.85	98.12	98.98	99.00	97.91	91.78	94.74	94.95
AdaBoost	99.94	99.72	99.83	99.83	98.03	92.47	95.17	95.34
Gradient-Boosting	100	89.03	94.55	94.20	99.97	89.02	94.18	94.54

In addition, Tables 5 and 6 represent the confusion matrices for the training and test sets respectively of the LR model using TF-IDF encoding. In our framework for detecting depression signs from social networks' users messages, the priority is to maximize the number of True positive (TP) and minimize the number of False Negative (FN), that is, tweets predicted as normal although being actually alarming. This is exactly what we can see in the confusion matrices, which shows the efficiency and relevance of our framework.

Table 5. LR confusion matrix for the training set with TF-IDF

	Predicted Alarming tweets	Predicted Normal tweets
Actual Alarming tweets	266181 (TP)	858 (FN)
Actual Normal tweets	5761 (FP)	257381 (TN)

Table 6. LR confusion matrix for the test set with TF-IDF

	Predicted Alarming tweets	Predicted Normal tweets
Actual Alarming tweets	29156 (TP)	515 (FN)
Actual Normal tweets	2125 (FP)	27113 (TN)

4 Related works and discussion

To examine existing work on depression detection in social networks using artificial intelligence, specifically machine learning algorithms, we conducted queries based on keywords such as 'depression,' 'social network,' and 'machine learning.' We excluded articles published before 2018 to focus on recent works. In the Scopus bibliographic database, these keyword-based queries returned over 513 research articles. We read their abstracts and selected 16 articles that appeared as the most relevant for in-depth analysis. In this section, we present the observations from these studies for each step of our proposed framework, including the phases of (1) data collection and annotation, (2) preprocessing, (3) hyperparameter tuning and (4) model training and evaluation.

Regarding data collection and annotation, we observed a relative scarcity of publicly available datasets, leading many researchers to create their own datasets using web-scraping techniques [21, 16, 1, 30, 8]. However, these approaches are often limited by annotation issues and class imbalance problems, raising the risk of overfitting and resulting in poor model generalization.

The data preprocessing phase, including cleaning and encoding steps, varies according to the dataset. For example, cleaning is typically done through tokenization, stopword elimination, and stemming [23, 7, 5, 25]. For data encoding, various methods such as TF-IDF, bag-of-words, and LIWC (Linguistic Inquiry and Word Count) are employed [21, 23, 5].

Concerning hyperparameter tuning and models training, authors trained and tested various models [7, 5, 10]. Commonly used models include DT, LR, SVM, RF, AdaBoost, and MLP. We noted a complete absence of the hyperparameter tuning phase in these works. Authors present their results without addressing this phase, using default hyperparameter values specified in software packages. However, hyperparameter tuning can enhance model performance and contribute to preventing overfitting, allowing models to generalize well to new data. For evaluating model performance, researchers typically use four main metrics: accuracy, precision, recall, and F1 score [21, 1, 14]. However, the confusion matrix, despite being a powerful tool for evaluating model performance, is underutilized. As we showed in this paper, the confusion matrix provides a detailed understanding of how a model classifies instances into different categories.

Moreover, various evaluation methods are employed, including the division into

training and test sets [1, 2, 5], as well as k-fold cross-validation [27]. It is important to note that these methods can introduce overfitting issues, especially with imbalanced datasets. For example, a heavily underrepresented class can result in training or test sets that do not adequately capture the variability of that class.

5 Conclusion and future work

In this paper, we proposed a framework using Machine Learning techniques in order to detect signs of depression in tweets from the social Network X users. We have overcome several limits of existing works in terms of bias and overfitting. To this end, we have introduced phases to specifically address class imbalance issues, to optimize hyperparameters and to perform stratified cross-fold validation. The experiments conducted on the dataset we collected showed that the TD-IDF was better than BoW for encoding data and that Linear Regression (LR) and Support Vector Machine (SVM) models presented the best performances than the others in terms of accuracy. One significant advantage of LR is its explanability, which is essential for our future work, that will include collaboration with healthcare professionals for further experiments and evaluation.

References

1. Angskun, Jitimon, Suda Tipprasert, and Thara Angskun. "Big data analytics on social networks for real-time depression detection." *Journal of Big Data* 9.1 (2022): 69.8.
2. Ansari, Luna, et al. "Ensemble hybrid learning methods for automated depression detection." *IEEE transactions on computational social systems* 10.1 (2022): 211-219.
3. Al Asad, Nafiz, et al. "Depression detection by analyzing social media posts of user." *2019 IEEE international conference on signal processing, information, communication & systems (SPICSCON)*. IEEE, 2019.
4. Benamara, Farah, et al. "Automatic detection of depressive users in social media." *Conférence francophone en Recherche d'Information et Applications (CORIA)*. 2018.
5. Cacheda, Fidel, et al. "Early detection of depression: social network analysis and random forest techniques." *Journal of medical Internet research* 21.6 (2019): e12554.
6. Chancellor, Stevie, and Munmun De Choudhury. "Methods in predictive techniques for mental health status on social media: a critical review." *NPJ digital medicine* 3.1 (2020): 43.
7. Chiong, Raymond, et al. "A textual-based featuring approach for depression detection using machine learning classifiers and social media texts." *Computers in Biology and Medicine* 135 (2021): 104499.
8. Ghosh, Tapotosh, et al. "An attention-based hybrid architecture with explainability for depressive social media text detection in Bangla." *Expert Systems with Applications* 213 (2023): 119007.
9. Go, Ale., Richa Bhayani and Lei Huang. "Twitter sentiment classification using distant supervision". CS224N Project Report, Stanford, 1.12(2009) :2009.
10. Govindasamy, Kuhaneswaran AL, and Naveen Palanichamy. "Depression detection using machine learning techniques on twitter data." 2021 5th international conference on intelligent computing and control systems (ICICCS). IEEE, 2021.

11. Hutto, Clayton, and Eric Gilbert. "Vader: A parsimonious rule-based model for sentiment analysis of social media text." *Proceedings of the international AAAI conference on web and social media*. Vol. 8. No. 1. 2014.
12. Institute of Health Metrics and Evaluation. Global Health Data Exchange (GHDx). URL : https://vizhub.healthdata.org/gbd-results/
13. Islam, Md Rafiqul, et al. "Depression detection from social network data using machine learning techniques." *Health information science and systems* 6 (2018): 1-12.
14. Kabir, Muhammad Khubayeeb, et al. "Detection of Depression Severity Using Bengali Social Media Posts on Mental Health: Study Using Natural Language Processing Techniques." *JMIR Formative Research* 6.9 (2022): e36118.
15. Kour, Harnain, and Manoj K. Gupta. "An hybrid deep learning approach for depression prediction from user tweets using feature-rich CNN and bi-directional LSTM." *Multimedia Tools and Applications* 81.17 (2022): 23649-23685.
16. Lin, Chenhao, et al. "Sensemood: depression detection on social media." *Proceedings of the 2020 international conference on multimedia retrieval*. 2020.
17. Majumder, P., M. Mitra, and B. B. Chaudhuri. "N-gram: a language independent approach to IR and NLP." *International conference on universal knowledge and language*. Vol. 2. 2002.
18. Musleh, Dhiaa A., et al. "Twitter Arabic Sentiment Analysis to Detect Depression Using Machine Learning." *Computers, Materials & Continua* 71.2 (2022).
19. Powers DMW. NLTK: the natural language toolkit. In: Proceedings of the COLING/ACL 2006 interactive presentation sessions. Sydney: Association for Computational Linguistics; 2006. p. 69–72.
20. Ríssola, Esteban A., Seyed Ali Bahrainian, and Fabio Crestani. "A dataset for research on depression in social media." *Proceedings of the 28th ACM conference on user modeling, adaptation and personalization*. 2020.
21. Safa, Ramin, Peyman Bayat, and Leila Moghtader. "Automatic detection of depression symptoms in twitter using multimodal analysis." *The Journal of Supercomputing* 78.4 (2022): 4709-4744.
22. Scikit-learn. URL: https://scikit-learn.org/stable/
23. Tadesse, Michael M., et al. "Detection of depression-related posts in reddit social media forum." *Ieee Access* 7 (2019): 44883-44893.
24. Twint. URL :https://github.com/twintproject/twint
25. Vasha, Zannatun Nayem, et al. "Depression detection in social media comments data using machine learning algorithms." *Bulletin of Electrical Engineering and Informatics* 12.2 (2023): 987-996.
26. Vijayarani, S., Ms J. Ilamathi, and Ms Nithya. "Preprocessing techniques for text mining-an overview." *International Journal of Computer Science & Communication Networks* 5.1 (2015): 7-16.
27. Wong, Tzu-Tsung. "Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation." *Pattern recognition* 48.9 (2015): 2839-2846.
28. World Health Organization" (WHO). URL : https://www.who.int/fr/health-topics/depression
29. Yang, Kailai, Tianlin Zhang, and Sophia Ananiadou. "A mental state Knowledge–aware and Contrastive Network for early stress and depression detection on social media." *Information Processing & Management* 59.4 (2022): 102961.
30. Zeberga, Kamil, et al. "A novel text mining approach for mental health prediction using Bi-LSTM and BERT model." *Computational Intelligence and Neuroscience* 2022 (2022).
31. Zogan, Hamad, et al. "Depressionnet: learning multi-modalities with user post summarization for depression detection on social media." *proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 2021.