Advances in Technology Innovation, vol. 9, no. 2, 2024, pp. 129-142

Improving Healthcare Communication: AI-Driven Emotion Classification in Imbalanced Patient Text Data with Explainable Models

Souaad Hamza-Cherif[1,*], Lamia Fatiha Kazi Tani[1], Nesma Settouti[2]

[1]Biomedical Engineering Laboratory, University of Tlemcen, Tlemcen, Algeria

[2]LabISEN - Yncréa Ouest, Caen, France

Received 31 March 2024; received in revised form 29 April 2024; accepted 30 April 2024

DOI: https://doi.org/10.46604/aiti.2024.13523

Abstract

Sentiment analysis is crucial in healthcare to understand patients' emotions, automatically identifying the feelings of patients suffering from serious illnesses (cancer, AIDS, or Ebola) with an artificial intelligence model that constitutes a major challenge to help health professionals. This study presents a comparative study on different machine learning (logistic regression, naive Bayes, and LightGBM) and deep learning models: long short-term memory (LSTM) and bidirectional encoder representations from transformers (BERT) for classify health feelings thanks to textual data related to patients with serious illnesses. Considering the class imbalance of the dataset, various resampling techniques are investigated. The approach is complemented by an explainable model, LIME, to understand the shortcomings of the classification results. The results highlight the superior performance of the BERT and LSTM models with an F1-score of 89%.

Keywords: sentiment analysis, data re-sampling, LSTM, BERT, LIME

1. Introduction

Sentiment analysis has become an important area of research in natural language processing (NLP), driven by the proliferation of social media platforms and online tools for sharing information. These platforms have revolutionized modern communication, allowing individuals to express their emotions and opinions online and providing valuable information about their thoughts and attitudes. Consequently, several works have focused on the analysis of sentiments from data extracted from the web: for example, during the COVID pandemic, as studied by Madani et al. [1], Chakraborty et al. [2], Xu et al. [3] focused on the analysis of the emotions of people faced with the virus, vaccination or the popularity of the type of vaccine and were able to provide interesting information on the ground.

In addition, analyzing people's emotions, especially those facing serious health problems such as cancer, dementia, and AIDS, proves to be very important in the fields of mental health and cognitive psychology. Many professionals in the field highlight the fact that emotional health is a crucial aspect of overall well-being. Positive emotions can promote the healing process, whereas negative emotions may worsen both the emotional and physical state of patients. Thus, automatically analyzing patient feelings using robust artificial intelligence (AI) models on textual data is a major challenge: on one hand, the data relating to patients with serious illnesses are few. Identifying the polarity of a feeling remains difficult because of the ambiguity that an emotion can produce in textual data format. In this context, producing models from AI for sentiment analysis is a major asset to help professionals in the field improve the emotional well-being of patients during the healing process, by detecting their feelings whatever they are.

* Corresponding author. E-mail address: souad.hamzacherif@univ-tlemcen.dz

English language proofreader: Chih-Wen Teng

However, some people, particularly healthcare professionals, are somehow hostile to the idea of using AI; because AI models can be seen as black boxes, that distrust its performance. Understanding the mechanisms underlying these predictions, particularly in health applications, is a major asset for the usability of diagnostic support and data analysis systems that use AI models. Ensuring transparency and explaining the results of the inference produced by artificial learning models is a factor of confidence in the developed tool, and a pillar of AI ethics, particularly for the classification of textual data which have the disadvantage of often being ambiguous and can lead to confusion.

In this paper, emotion classification is delved deeper into health-related text data, specifically the EmoHD dataset [4], using learning models based on neural networks and deep neural networks. This dataset includes 4,202 text samples, collected from different online sites, containing information related to more than eight classes of serious diseases such as HIV/AIDS, dengue, hepatitis, malaria, influenza, coronavirus, cancer, etc. This data is classified into six different categories. The course of emotions (happy, sad, angry, excited, bored, scared). The importance of studying such a dataset is obvious, as automated emotion recognition in patient health data is essential for advancing to improve research in this area but can also be an interesting diagnostic aid tool. for cognitive psychology, favoring the analysis of the emotional impact of patients. Suffering from a serious illness in the healing process. Such an analysis could offer an ideal framework to follow the emotional evolution of patients about the progression of their illness. Hence the importance. to analyze sentiments in such a context, especially since datasets related to sentiments related to serious illnesses are not available.

This work aims to compare the performance of different machine learning with deep learning models: logistic regression, naive Bayes (NB), LightGBM, long short-term memory (LSTM), and bidirectional encoder representations from transformers (BERT), which are often used in the field of sentiment analysis. and explore them on the EmoHD dataset. The studied dataset is unbalanced; different resampling methods are considered as well as their impact on the classification and results. Additionally, the analysis of the classification results was obtained using a local interpretable model agnostic explanation (LIME) local explainability model to understand the results of the worst classifier, having obtained the wrong classification score. The key stages of this study concern:

(1) Implementing data preprocessing tasks and various data re-sampling techniques (NearMiss, SMOTE, ADASYN).

(2) Conducting sentiment classification on input data using feature selection methods (continuous bag-of-words (CBOW), term frequency-inverse document frequency (TF-IDF), Word2vec) and machine learning classification through base models (logistic regression, NB, LightGBM).

(3) Comparing the classifications obtained after rebalancing the database using both basic models and recurrent deep learning networks, specifically LSTM and BERT from transformers.

(4) Analyzes classification predictions utilizing word embeddings with LIME.

The rest of this article is structured as follows: Section 2 covers a study of related work in the field, Section 3 presents the approach proposed in this study covering all the steps followed as well as the presentation of the classification models used and the model of LIME explainability. In Section 4, the experimental results are presented and analyzed. Finally, in the last section conclusions and a discussion on potential directions for future work are drawn.

2. Related Works

Sentiment analysis is a field in constant evolution. Several current works have focused on this problem, especially with the advent of the social and semantic web which has offered sharing tools allowing different users to express their opinion and their feelings, thus offering the researcher in the field the possibility of extrapolating people's feelings using different AI and NLP tools. The task of sentiment analysis from textual data encompasses different techniques such as lexicon: these approaches use dictionaries, or corpora, to assign a polarity score to each word in lists of manually classified words (positive or negative).

For example, in Srinivasan et al. [5], researchers used a lexicon-based approach to predict election results using knowledge of emotion classification from Twitter data related to the election of Hillary Clinton and Donald Trump. Lin and Liao [6] proposed a lexicon-based method for sentiment analysis specific to the financial domain. Catelli et al. [7] used the Italian sentiment lexicon containing polarized words, expressing a semantic orientation, to identify the sentiments of people vaccinated against COVID-19 which highlighted an overall negative sentiment.

Other newswork uses various machine and deep learning techniques to classify sentiment from text data. Whatever the method used, it follows a pipeline of steps ranging from preprocessing the data, which is generally noisy and unstructured, to feature selection and vectorization, which consists of extracting certain distinct features from the text on which the model can focus. training, by converting the text into digital vectors, this step is necessary in the case of using machine learning models, and finally the sentiment classification step. Works in this case are abundant in the literature, for example, Wise et al. [8] proposed a method based on latent semantic analysis (LSA) of tweets in social media for the classification of cyberbullying texts and achieved an accuracy score of 91% on training data.

Bhaskaran et al. [9] presented a new modified red deer algorithm (MRDA) extreme learning machine sparse autoencoder (ELMSAE) model for sentiment analysis on a benchmark dataset including d mobile applications for Google, their classification process includes vectorization by the TF-IDF model reaching an accuracy score of 98%. Tan et al. [10] present an approach for sentiment analysis and sarcasm detection simultaneously from online textual data using multi-task learning which consists of training both tasks simultaneously with a shared layer of the bidirectional long short-term memory (Bi-LSTM) model. The proposed framework aims to improve the performance of autonomous sentiment classification by adding an auxiliary task which is sarcasm detection and obtained an F1-score of 94%.

Meena et al. [11] proposed a hybrid model based on deep learning to know users' opinions on the Monkeypox infection on social networks, combining convolutional neural networks (CNN) and LSTM, and obtained an F1-score of 95%. Das et al. [12] compared different deep learning and hybrid models for the sentiment analysis of comments on an e-commerce site, for the classification of texts in English and Bengal, and demonstrated that the model support vector machine (SVM) outperformed other models, achieving an accuracy of 82.56% for sentiment analysis of English texts and 86.43% for sentiment analysis of Bengali texts. Umair et al. [13] proposed an approach to analyze tweets related to COVID-19 vaccines and combined the BERT + NBSVM model to classify people's sentiments towards vaccines.

This choice is motivated by taking advantage of both bidirectional BERT and NBSVM functionalities from transformers and circumventing the limitations of BERT-based approaches, which only leverage encoder layers, resulting in lower performance on short texts. The model achieved a performance of 73% accuracy, 71% precision, 88% recall, and 73% F-measure for positive sentiment classification, while 73% accuracy, 71% precision, 74% recall, and 73% F-measure for classification of negative feelings respectively.

After comparison with this state of the art, some points can be got:

(1) Although lexicon-based and machine-learning approaches offer valuable insights into sentiment analysis, this study does not aim at sentiment extrapolation using lexicon as seen in previous works [5-7]. Instead, it introduces a unique perspective by applying machine learning methods to health-related text data.

(2) Furthermore, compared to works based on artificial learning [8-13], this study not only improves the transparency of results by integrating explainability into analysis but also contributes to the growing field of explainable AI in sentiment analysis. This approach not only advances this understanding of sentiment classification in a critical area but also sets a precedent for future research in applying explainability to deeply understand model decisions.

Based on these expertise, this study focuses on the EmoHD dataset to demonstrate the practical application of sentiment analysis techniques in a healthcare context. This study framework therefore differs from the works cited [8-13], as its focus is on applying various sentiment analysis methodologies to the EmoHD dataset, which contains a unique compilation of textual data. derived from patient feedback on serious illnesses. This dataset, originally introduced by Azam et al. [4], encompasses feedback from individuals diagnosed with a range of diseases, including measles, dengue, typhoid, malaria, acute hepatitis, HIV, Ebola, and cancer. The pioneering study on EmoHD achieved an impressive accuracy of 87% using the multi-layered perceptron algorithm, setting a benchmark for further research. Encouraged by these early successes, subsequent research has sought to push the boundaries of sentiment analysis in healthcare even further. In Mohammad et al. [14], for example, the intelligent water drop algorithm was used to select informative features from EmoHD, demonstrating the potential of innovative algorithms to improve feature selection and, therefore, feelings of classification accuracy in health-related texts.

This exploration of the EmoHD dataset is part of a broader scientific interest, as evidenced by numerous studies aimed at leveraging NLP and machine learning to deepen this understanding of patient experiences in various healthcare settings, including health care. Several studies have addressed emotional recognition [15-17] and its applications in healthcare [18]. This work highlights the broad interest in leveraging NLP and machine learning techniques to understand patients' experiences, emotions, and feelings, which is crucial to improving outcomes for the patients. healthcare and patient support.

By leveraging insights from the EmoHD dataset, this study (in comparison with [4, 14]) aims to contribute further by evaluating the performance of different learning algorithms such as logistic regression, Bayes naive, and LightGBM. integrating a variety of metrics (CBOW, TF-IDF, and Word2vec) and comparing them to recurrent deep learning networks, including LSTM and BERT models which have shown good results in sentiment classification [10-11, 13].

Given the dataset's inherent imbalance across six sentiment classes (happy, sad, scared, angry, excited, bored), this approach includes exploring various resampling methods (naive subsampling, NearMiss, SMOTE, and ADASYN) to address associated challenges and evaluate their impact on sentiment classification results. The other contribution of this approach is the incorporation of explainable artificial intelligence (XAI), a response to the critical need for transparency and understandability in healthcare sentiment analysis, particularly when navigating nuanced sentiments within the EmoHD dataset. Explainability offers a certain transparency on the behavior of the classifier and thus makes it possible to better explain its decisions and shortcomings in its performance. Explainability techniques can be classified based on their conceptual capacity [19].

In this case, two classes of methods are distinguished: ante-hoc (intrinsic) approaches that automatically generate the explanation as part of the prediction process, referring to machine learning models considered interpretable due to their simple structure, such as short or sparse linear decision trees; and post hoc models and approaches that apply interpretation methods after model training. Others classify XAI methods based on their scope, whether global or local: local explanations focus on the classification result of a given instance, while global explanations provide an overview of the entire model.

Through this targeted application of XAI techniques on the EmoHD dataset, the aim is not to only improve the clarity of model decisions but also to lay the foundation for future explorations of the complex relationship between AI decisions and healthcare outcomes. In this study, the understanding of the local scope of the learning model which gave us the worst result in terms of F1-score using the LIME model [20] to have clear and complete explanations of the predictions obtained and to identify gaps related to annotation and sentiment polarity.

3. Proposed Approach

In this section, a description of the approach to sentiment classification in health-related text data, as well as explaining the classifier's predictions. This study is based on the EmoHD dataset [4], which includes 4,202 text samples classified into eight disease classes and six emotion classes, from various online platforms.

Fig. 1 illustrates this 5-step approach: the first is the preprocessing of the textual data which is essential especially since the data is often ambiguous and noisy, after that, a process of rebalancing the extremely unbalanced data. This can be observed in Fig. 2, which displays an exploration of different re-sampling methods, then a comparison of basic classification models (logistic regression, NB, LightGBM) with different vectorization models (CBOW, TF-IDF, and Word2vec) and in parallel an implementation of recurrent deep learning networks is placed: the LSTM sequential model and BERT to compare the performance of the convolutional classifier with the base models. The last step of this approach concerns the explainability of the learning model having obtained the worst classification results through the analysis of the predictions using the LIME model.

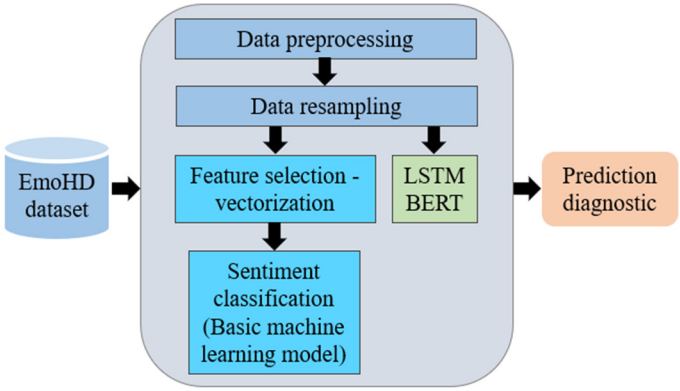

Fig. 1 Classification and analysis of sentiment predictions on health-related imbalanced textual data approach

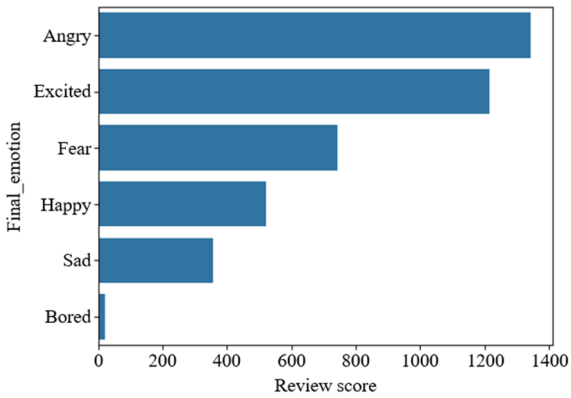

Fig. 2 EmoHD dataset class distribution

3.1. *Data preprocessing*

To classify the textual data in the EmoHD dataset, it is necessary to perform preprocessing to remove noise, and make it usable, especially since this data comes from the web and is unstructured. This cleaning process aims to improve sentiment classification results. Here are the main measures taken:

(1) **Lowercase conversion:** The first step in preprocessing is to convert uppercase terms to lowercase to avoid redundant words. Even if the spelling of two upper- and lower-case words like “Health” and “health” are semantically identical, they are treated as two different lexical units.

(2) **Punctuation removal:** This step involves removing any punctuation from the text that does not provide any useful information to improve data classification performance.

(3) **Delete stop words:** These are very common words in the language studied that do not provide any informative value for understanding the “meaning” of a document and a corpus, which are isolated and deleted.

(4) **Removal of rare and common words:** To avoid the noise that rare words can generate in a text and the impact that too frequent words can also have in the classification, remove rare and common terms by counting their frequencies in the text.

(5) **Lemmatization:** It consists of replacing each word with its canonical form, for example, the known word refers to its canonical form namely. This step is useful for the thematic classification of texts because it allows the different variants resulting from the same form or canonical root to be treated as a single word.

(6) **Tokenization:** The process of dividing text into tokens or linguistic units such as words, punctuation marks, numbers, and alphanumeric data. Each element corresponds to a token which will be useful for the analysis.

3.2. *Data re-sampling*

As shown in Fig. 2, EmoHD data is unbalanced, which may impair classification performance, so resampling is necessary. There are several ways to handle imbalanced data, such as undersampling and oversampling. In this study different resampling methods are tested to see their impact on the results of the classification, the choice of methods used covers those best known

in the additional field: naive undersampling, which consists of removing certain minority classes (in the case of EmoHD data this is the minority class Sad), and oversampling which involves adding additional copies of observations from the minority class to balance the class distribution, and in this perspective this study tests different oversampling approaches:

(1) **Naive oversampling:** This involves randomly duplicating the observations of the minority class to strengthen its signal by resampling with replacement.

(2) **NearMiss:** There are 3 versions of NearMiss. In this study, an experiment with NearMiss1 selects examples from the majority class that are close to three of the closest examples from the minority class and removes them [21].

(3) **SMOTE:** Synthetic minority oversampling technique [22], is a method for oversampling minority observations. The SMOTE algorithm generates new minority individuals similar to existing ones, without being strictly identical. This helps to evenly distribute the population of minority individuals.

(4) **ADASYN:** The goal of ADASYN is to generate an appropriate number of synthetic alternatives for each minority class observation [23]. The concept of "appropriate number" depends on the difficulty of learning the original observation.

3.3. Feature selection

As textual data cannot be used directly in machine learning algorithms including classical neural networks, they must be converted into digital representations. This process, called vectorization, transforms raw text data obtained after cleaning into feature vectors. Several approaches can be used to perform this task, each extracting different types of features from the text. This study, compares different vectorization methods, including count vectorization, TF-IDF, and Word2vec, to obtain relevant features from text data. First, an implementation of the CBOW model, which is a textual representation that describes the occurrence of words in a document, ignoring the order or structure of the document. The model only cares whether known words appear in the document [24].

Furthermore, experiment with TF-IDF which is a statistical technique used in information retrieval and data mining to quantify words in a set of documents [25]. TF-IDF is based on the frequency of words in a text, as described by the Zipf law [26]. TF measures the importance of a term in a document, while IDF measures whether the term is discriminative (i.e., not widely distributed). Therefore, a term with a high TF-IDF value should be prominent in the current document and also appear rarely in other documents.

The last model tested is the Word2vec [27] embedding model based on a two-layer neural network trained to predict the vector representation of words in context. Simply put, Word2vec takes a corpus of text as input and generates a set of vectors for the words in that corpus. Its goal is to group vectors of similar words into a vector space. With enough data, usage, and context, Word2vec can accurately guess the meaning of a word based on its past appearances. They used the Google learning model available online†, consisting of 300 vector dimensions for 3 million words and sentences, as a training basis.

3.4. Baseline modeling

In this step, basic classification approaches — logistic regression, NB, and LightGBM, using different feature selection models (CBOW, TF-IDF, and Word2vec). Logistic regression is a statistical approach used for classification problems when the dependent (target) variable is categorical. Logistic regression uses the sigmoid mathematical function to return the probability of a label [28]:

$$Sg(x) = \frac{1}{1+e^{-x}} \tag{1}$$

† https://code.google.com/archive/p/ word2vec

The second implemented NB model is a probabilistic machine learning model used for classification tasks. In this case. The classifier node is based on Bayes theorem 2:

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)} \tag{2}$$

Using this theorem, the probability that A will occur will be found, given that B has occurred (B is the evidence and A is the hypothesis) [29]. The last implemented model is LightGBM, it is a decision tree-based optimization ensemble method used in classification and regression. This model creates leaf-aware decision trees, such that the best-fit leaf of the tree will be split while further strengthening calculations split the depth of the tree into two parts, one wise and the other insightful, as opposed to leaves [30].

3.5. Recurrent neural networks

Given the impressive results of deep learning algorithms, particularly in the analysis of textual data [11-13], an exploration of their scope on the EmoHD dataset, so two types of recurrent neural networks (RNN) are implemented: LSTM and the BERT, and compared them to the learning algorithms implemented previously based on their classification performance.

3.5.1. Long short-term memory (LSTM)

LSTM is a variant of RNN, designed to handle time series data or sequences. What sets LSTMs apart is their ability to handle inputs of different lengths, thanks to their short-term memory capabilities. Additionally, LSTMs are excellent at understanding context because they can process data packets almost simultaneously. In this approach, it implemented a sequential LSTM model to solve a predictive modeling problem where it needed to predict a category for an input sequence. The implemented LSTM model consists of four layers presented in Fig. 3.

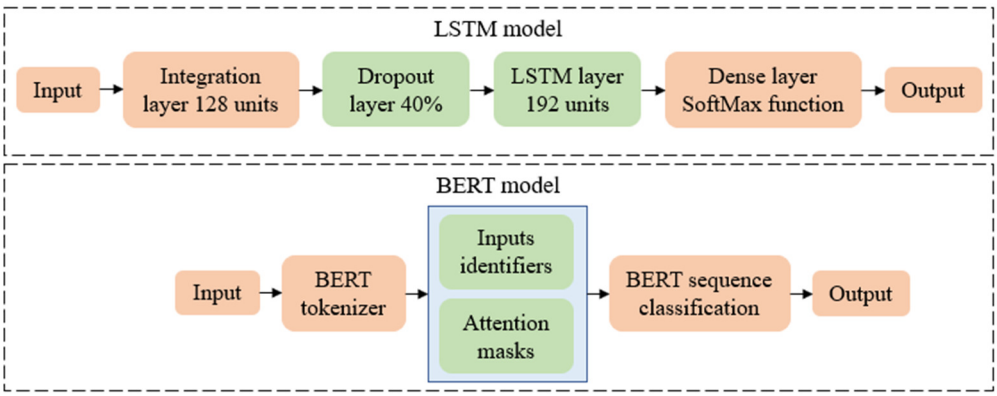

Fig. 3 LSTM and BERT model architecture

The input layer (integration layer) consists of 128 LSTM units which map the words of the text into real-valued vectors. This layer takes as input three entities, including: the vocabulary size dimension of each embedded word (input dimension), the maximum number of words in the vocabulary (maximum feature), and the maximum length of a sequence (input length). The second layer is the dropout layer: initialized this layer with a rate of 40% to reduce overfitting when training the model. This will apply random deactivation in each epoch, meaning that on each pass (forward propagation), the model learns with a configuration of different neurons activating and deactivating randomly (40%). The third layer (LSTM layer) contains 192 memory units (intelligent neurons) to optimize the performance of the model, this layer is configured with dropout which has been set at 40%, and recurrent dropout which concerns the suppression applied to the recurring input signal on LSTM units is set at 20%. The output layer (dense layer) implements the activation function. In this case, a softmax function is used since the model has 6 output classes this number is reduced to 4 after removing the minority classes.

During the model compilation phase: the LSTM model is configured with a categorical_crossentropy loss function (which measures the dissimilarity between the predicted probabilities and the true categorical labels guiding the model to minimize the difference between them) and the Adam optimizer which improves the training process by adapting the pace of learning. The training set was trained for 200 epochs which represents the number of times the model goes through the data set provided for training. This number of iterations seemed correct to us after several tests to give the model the possibility of analyzing the data several times and extracting the relationships and complex variations present in the data set to draw lessons from it. The batch size which determines the number of samples that will be fed into the model in each iteration, and which will allow it to calculate the loss and update its weights accordingly has been divided into 64 batches.

3.5.2. Bidirectional encoder representations from transformers (*BERT*)

BERT is a text representation model developed by Google that is context-aware, meaning that a word is represented based on its context in the text. Additionally, BERT's context is bidirectional, meaning that the representation of a word takes into account not only the words preceding it in a sentence but also the words following it. BERT uses the transformer attention mechanism, which learns the contextual relationships between words (or sub-words) in a text. In its traditional form, the transformer includes two mechanisms: an encoder that reads input text and a decoder that produces a prediction for a task. As BERT is used to generate a language model, only the coding mechanism is needed.

In this model, the use of the bert-base-cased model from the transformer's library allows easy access to transformation models and simplifies many technical implementation details. This model is advantageous for the classification of multi-label texts, it remains case sensitive, otherwise, an already solved of this problem during the preprocessing phase of data.

As you can see in Fig. 3, the model takes labeled training and test data as input and then loads BERT's tokenizer, which divides the text into subword units called tokens. These tokens are then encoded and converted to digital format, thus forming input identifiers (Input IDs) and attention masks. Input IDs are integer identifiers of each token in a sequence (a set of them to a length of 256, is chosen based on the token length distribution). In this case, each unique token of the vocabulary is assigned an integer identifier which represents a shortcut of a one-hot encoded vector: one-hot encoded vectors are only vectors in which the element at the index corresponding to the represented token has a value of one, and all other elements have a value of zero. Attention masks are applied to extend sequences shorter than the maximum sequence length with padding tokens. Fill tokens are just placeholders and do not influence the model output in any way.

Subsequently, a simple addition of a classification layer takes as input the sequence level embedding and generates the class label, for this, BertForSequence-Classification acts as a linear layer on top of the final layer of the transformer and classifies the input data. To refine this BERT classifier, an optimizer is used with fixed weight loss AdamW, which is parameterized with a learning rate lr = 1e-5, and fixed Adam's epsilon for numerical stability (eps = 1e-8). Adam trained this model on 3 epochs in batches of 3.

3.6. Explainable predictions diagnostic

XAI encompasses a set of processes and methodologies designed to improve this understanding of the results generated by machine learning algorithms. These methodologies aim to shed light on how classification models arrive at their predictions. Typically, this involves providing textual or visual explanations that explain the connection between the input features of a prediction (e.g., words in a text) and the model's output [20].

In this research approach, an implemented LIME is applied to analyze classifiers that exhibit lower performance than other baseline models. The goal is to better understand the contributions of specific words to the output class predictions, particularly regarding the classifier that gave the lowest scores among the implemented baseline models.

As shown in Fig. 4, the LIME model works independently from the classification model used. It can provide clear and precise explanations for individual predictions made by any classifier or regressor. LIME achieves this goal by locally approximating the model with an interpretable model, thereby facilitating the interpretability of complex model decision-making processes. The explainability process in LIME involves two fundamental steps: during the first step, employ a LIME explainer to calculate the contribution of each word to a class. The LIME explainer takes as inputs the perturbed data (obtained by permuting an observation), the labels (the classes for which an explanation is done), and the distances (similarity distance between the original observation and the perturbed observations). The selected machine learning model is applied to predict the results of the perturbed data.

At the output of this step, the weights of the resulting features are obtained to explain the behavior of the classifier (explanation of list [word, contributing weight, label]). Secondly, a calculation of the average contribution of each word to a class allows us to sort and classify words according to their impact as detractors or supporters in the classification model.

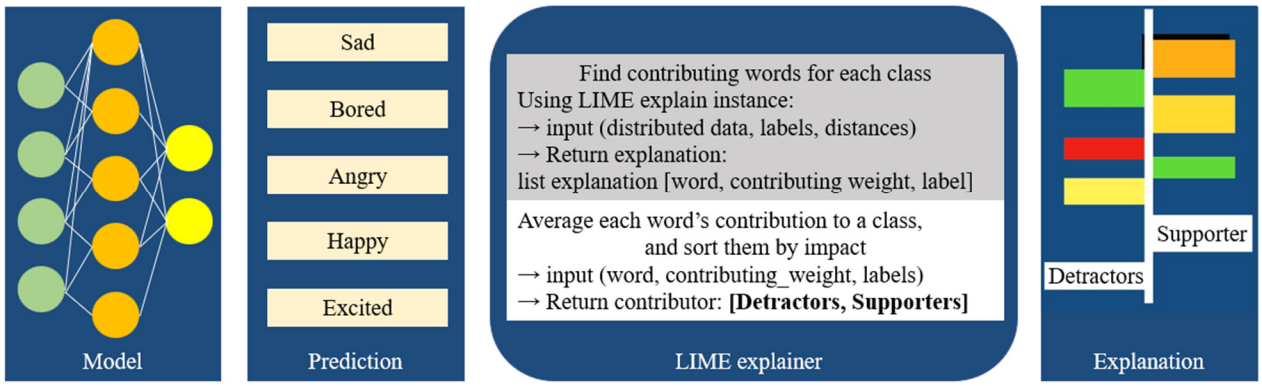

Fig. 4 LIME model process

4. Experiments and Results

This section contains the results of the experiment conducted in this study. It presents the experiment setup and the metric used for evaluation, followed by the classification results with baselines and deep learning models. Additionally, it analyzes the results of the explainability provided by the LIME model.

4.1. Experimental setup

The present experiment was conducted using the Google Colab platform with the aim of training learning models. This platform provides unlimited access to high-performance graphics processing units (GPUs) with minimal configuration requirements.

4.2. Evaluation metric

This study is a comparison of the performance of various baseline models on the EmoHD dataset and the performance of LSTM and BERT for sentiment classification. During the experimentation, the preprocessed data was split using the train_test_split function, so that 20% was allocated for testing and the remaining 80% was used for training.

To evaluate the performance of each classifier, utilize the F1-score, a standard evaluation metric that is more suitable for imbalanced datasets than accuracy. The F1-score is a weighted average of precision and recall and takes into account both false positives and false negatives. The mathematical expression for the F1-score is shown in:

$$\text{F1-score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{3}$$

4.3. Results of classification with baseline methods

A trained of all the machine learning algorithms as described in the "basic modeling" section with the models: logistic regression, NB, and LightGBM, each time using a different feature selection method (CBOW, TF-IDF, Word2vec) and an evaluation of their performance on the EmoHD dataset, including and excluding minority data classes, using five different resampling methods: Naive undersampling, Naive oversampling, oversampling with NearMiss, SMOTE, and ADASYN. The results of these evaluations are presented in Tables 1 and 2.

Table 1 F1-score Implemented Baseline including minority classes

Model	Feature extraction	DC	Naive US	Naive OS	NearMiss	SMOTE	ADASYN
NB	CBOW	0.476	0.574	0.700	0.574	0.579	0.593
LR		0.488	0.608	0.823	0.609	0.617	0.622
LGBM		0.485	0.633	0.603	0.633	0.577	0.613
NB	TF-IDF	0.489	0.638	0.680	0.638	0.599	0.599
LR		0.504	0.617	0.839	0.618	0.629	0.628
LGBM		0.495	0.639	0.613	0.642	0.585	0.592
LR	Word2vec	0.489	0.612	0.631	0.616	0.612	0.613

Table 2 F1-score Implemented Baseline excluding minority classes

Model	Feature extraction	DC	Naive US	Naive OS	NearMiss	SMOTE	ADASYN
NB	CBOW	0.472	0.544	0.737	0.556	0.577	0.551
LR		0.486	0.581	0.803	0.608	0.616	0.633
LGBM		0.465	0.611	0.578	0.604	0.539	0.551
NB	TF-IDF	0.475	0.588	0.667	0.587	0.595	0.608
LR		0.490	0.610	0.831	0.631	0.671	0.676
LGBM		0.487	0.609	0.618	0.601	0.531	0.540
LR	Word2vec	0.483	0.592	0.600	0.569	0.527	0.539

4.4. Results of classification with recurrent neuronal networks (*LSTM and BERT*)

In this experiment, an LSTM and a BERT model (all parameters are explained in previous sections) are trained with three different approaches: without re-sampling, with oversampling including minor classes, and with oversampling excluding minor classes. The obtained F1-scores are in Table 3:

Table 3 F1-score of LSTM and BERT models

Re-sampling approach	LSTM	BERT
Without re-sampling	49%	57%
With oversampling including minor classes	82%	82%
With oversampling excluding minor	89%	89%

4.5. Analysis of results of classification

By closely analyzing Tables 1, 2, and 3: unbalanced data significantly affects the classification and it is necessary to resample the data to obtain better results. The inclusion or exclusion of minority classes has no significant impact on the classification results according to the different basic models. The scores obtained in Tables 1 and 2 are almost identical which is not the case in Table 3 which clearly shows the improvement in the classification rate after removing the minority classes. This leads to the deduction that oversampling has more impact on the classification using machine learning models which is not the case for deep learning algorithms, which are already more efficient. In this specific case, it is clear that the vectorization step and the choice of model have a significant impact on the final results. By analyzing Tables 1 and 2:

(1) Choosing a good resampling method is important to improve classification. In this study, naive oversampling gave the best results on average for the baseline algorithms and even the highest F1-score was obtained using logistic regression with TF-IDF and naive oversampling.

(2) Table 1 clearly shows that the logistic regression learning model using feature selection with TF-IDF outperformed others with a satisfactory F1-score of 83.9% when including the minority class in the naive oversampling case and 83.1% in the case of the exclusion of minority classes. These results demonstrate the choice of using TF-IDF is decisive for the improvement of the classification because even by including the minority classes the variation between the two F1-scores obtained is significant and the result remains satisfactory.

(3) Furthermore, the use of the Word2vec model does not give satisfaction, especially in the case of the exclusion of minority classes. It seems obvious that the model does not manage to select the right characteristics of words which leads to wrong classification.

(4) The logistic regression model based on Word2vec has obtained the worst result in terms of F1-score, an explanation of these predictions is in the following section.

Looking to further, improve the classification score has been obtained using LSTM and BERT. As observed in Table 3, the F1-scores obtained by the LSTM and BERT models after oversampling the data, with or without minor classes were similar. The best F1-score of 89% was obtained with BERT and LSTM models with oversampling and exclusion of minority classes. Overall, both the LSTM and BERT models gave better performance than the baseline models. However, the BERT model outperformed other models even when classifying imbalanced data without resampling, achieving an F1-score of 57%.

The negative point observed about the execution of deep learning models is that they are very time-consuming. In particular, the BERT model took an average of 6 hours to execute compared to the LSTM model, which took 1 hour on average. In contrast, the base models were the fastest; with an average execution time of 40 seconds. Despite the Despite the time-consuming nature of the execution, the classification models used in this study particularly LSTM and BERT, achieved satisfactory results in terms of F1-score, comparing them to the models proposed in the literature. Table 4 shows us the different F1-score measurements obtained by other models on the EmoHD database, but also on other works using recurrent deep learning models on other textual databases.

Table 4 F1-score of related works

Database	Approach	F1-score
EmoHD	This approach with the LSTM and BERT model	89%
	Multi-layered perceptron [1]	87%
	Back-propagation neural network (BPNN) based intelligent water drop algorithm [14]	96%
Another textual database using recurrent deep learning	BI-LSTM [10]	94%
	CNN + LSTM [11]	95%
	BERT + NBSVM [13]	73%

Table 4 highlights that the classification results obtained by this BERT and LSTM models outperformed the MultiLayer perceptron implemented by Azam et al. [4]. Although these results are satisfactory, they remain lower than those of the intelligent water drop algorithm based on the back-propagation neural network (BPNN) model [14]. It seems obvious that using the intelligent water drop algorithm for feature selection optimization made a difference and significantly improved the classification result. Compared to the results of related works on different text databases these transformer models were more efficient than those in Umair et al. [13], which only obtained an F1-score of 73%, also, recurrent LSTM models used by Tan et al. [10], and Meena et al. [11] were very successful with F1-scores of 94% and 95%, these results which exceed this study are encouraging regarding the usability and performance of recursive neural networks in classification feelings from textual data.

4.6. Classification diagnostic

As shown in Table 1, the results obtained by the basic logistic regression machine learning model with the Word2vec method are not satisfactory, the F1-score obtained was the worst of all. To explain these classification results, it is relevant to

understand how these classifications are made, so, the LIME model is used to see how the predictions of the logistic regression model with Word2vec on the EmoHD database were influenced. For this evaluation, a selected of two classes representing opposite feelings: “Sad” and “Happy”. Fig. 5 shows the most important words deemed relevant or not for these two classes.

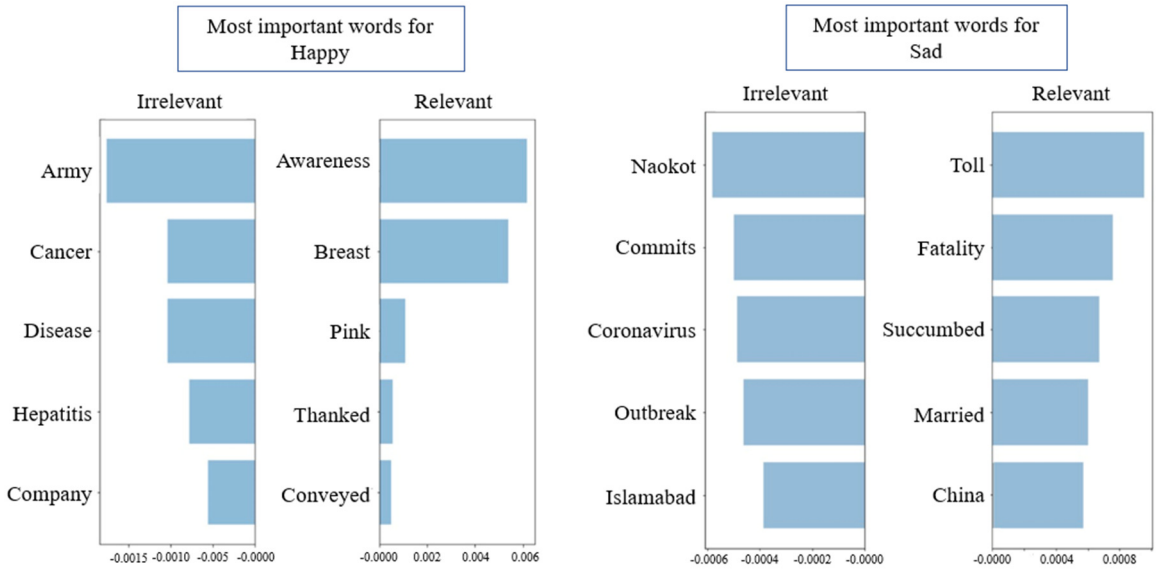

Fig. 5 Most important words for Happy and Sad class

It can be noted that the category of “Happy” feelings should be expressed through words that evoke feelings of joy and happiness. Relevant terms in this class include “Pink”, “Thanked”, and “Awareness”, which inspire positive feelings. Terms that do not belong to the “Happy” class include “Cancer”, “Army”, and “Hepatitis”, which only inspire negative feelings of sadness.

In the “Sad” feeling class, among the important terms for the classification considered favorable, found “Toll”, “Married”, and “China”, which are not relevant for identifying a feeling of sadness. Detractor words include “Outbreak” and “Coronavirus”, which are words expressing illness and sad events, but they appear in the list of detractors in the “Sad” class.

Using LIME, the results of the classification model are presented with some details. A perception of the Word2vec model has difficulty identifying the appropriate lexical representations to support good classification in certain classes, such as the “Sad” class. This is not the case for the “Happy” class, where the predictions are consistent. It is clear that a feature of the “Sad” feeling class belongs to the minority classes, made up of 395 negative feelings. Additionally, it is important to note that the feeling “Sad” can easily be confused with emotions expressed in other classes such as “Angry”, “Bored”, and “Fear”, potentially leading to misclassifications and confusion. This classification provided by LIME is very interesting and can be used to improve the performance of the model, thanks to the addition of preprocessing which removes ambiguous words, or to the parameterization of the Word2vec model that thus varies the contextual size and verifies the change in obtained results.

5. Conclusion and Perspectives

In this paper, an approach was presented to classify sentiments from imbalanced health text data. The classification scores obtained confirmed the impact of imbalanced data on classification and the importance of selecting an appropriate resampling method. For feature selection in this case, TF-IDF performed better than the Word2vec model. The results also demonstrated the superiority of the LSTM and BERT deep learning models, compared to the reference models. It has also been found that models like LIME can help identify words that influence the classification of each class depending on the learning model used. The comparative study of related research works demonstrated the performance of deep learning models such as LSTM, BI-LSTM, and hybrid models, which is encouraging for the usability of such models in the field of sentiment classification from textual data.

Even though the EmoHD database explored in this study is interesting in terms of application in cognitive psychology, in particular, to help identify the mental state of patients suffering from serious illnesses and to promote a healing process by addressing negative thoughts of patients thanks to a dedicated diagnosis providing an automatic system based on robust learning models for sentiment analysis; nevertheless, EmoHD presents many limitations in terms of unbalanced data, but also in terms of ambiguity of classes like the sad, angry, and bored classes are very close and can lead to confusion a term representing a negative thought: for example, the word war can be classified into these 3 classes.

In this context, some limitations were observed for future research, it is wise to explore other sentiment databases in the health domain using recursive models and transformers. This is a relevant area for analyzing the impact of positive or negative feelings toward patients. This can be very beneficial in the field of health, both in the healing process of patients and in the field of psychology.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Y. Madani, M. Erritali, and B. Bouikhalene, "A New Sentiment Analysis Method to Detect and Analyse Sentiments of Covid-19 Moroccan Tweets Using a Recommender Approach," Multimedia Tools and Applications, vol. 82, no. 18, pp. 27819-27838, July 2023.

[2] K. Chakraborty, S. Bhatia, S. Bhattacharyya, J. Platos, R. Bag, and A. E. Hassanien, "Sentiment Analysis of COVID-19 Tweets by Deep Learning Classifiers—A Study to Show How Popularity is Affecting Accuracy in Social Media," Applied Soft Computing, vol. 97, part A, article no. 106754, December 2020.

[3] H. Xu, R. Liu, Z. Luo, and M. Xu, "COVID-19 Vaccine Sensing: Sentiment Analysis and Subject Distillation from Twitter Data," Telematics and Informatics Reports, vol. 8, article no. 100016, December 2022.

[4] N. Azam, T. Ahmad, and N. Ul Haq, "Automatic Emotion Recognition in Healthcare Data Using Supervised Machine Learning," PeerJ Computer Science, vol. 7, article no. e751, 2021.

[5] S. M. Srinivasan, R. S. Sangwan, C. J. Neill, and T. Zu, "Twitter Data for Predicting Election Results: Insights from Emotion Classification," IEEE Technology and Society Magazine, vol. 38, no. 1, pp. 58-63, March 2019.

[6] W. Lin and L. C. Liao, "Lexicon-Based Prompt for Financial Dimensional Sentiment Analysis," Expert Systems with Applications, vol. 244, article no. 122936, June 2024.

[7] R. Catelli, S. Pelosi, C. Comito, C. Pizzuti, and M. Esposito, "Lexicon-Based Sentiment Analysis to Detect Opinions and Attitude Towards COVID-19 Vaccines on Twitter in Italy," Computers in Biology and Medicine, vol. 158, article no. 106876, May 2023.

[8] D. C. J. W. Wise, S. Ambareesh, P. Babu, D. Sugumar, J. P. Bhimavarapu, and A. S. Kumar, "Latent Semantic Analysis Based Sentimental Analysis of Tweets in Social Media for the Classification of Cyberbullying Text," International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 7s, pp. 26-35, 2024.

[9] R. Bhaskaran, S. Saravanan, M. Kavitha, C. Jeyalakshmi, S. Kadry, H. T. Rauf, et al., "Intelligent Machine Learning with Metaheuristics Based Sentiment Analysis and Classification," Computer Systems Science and Engineering, vol. 44, no. 1, pp. 235-247, 2023.

[10] Y. Y. Tan, C. O. Chow, J. Kanesan, J. H. Chuah, and Y. L. Lim, "Sentiment Analysis and Sarcasm Detection Using Deep Multi-Task Learning," Wireless Personal Communications, vol. 129, no. 3, pp. 2213-2237, April 2023.

[11] G. Meena, K. K. Mohbey, S. Kumar, and K. Lokesh, "A Hybrid Deep Learning Approach for Detecting Sentiment Polarities and Knowledge Graph Representation on Monkeypox Tweets," Decision Analytics Journal, vol. 7, article no. 100243, June 2023.

[12] R. K. Das, M. Islam, M. M. Hasan, S. Razia, M. Hassan, and S. A. Khushbu, "Sentiment Analysis in Multilingual Context: Comparative Analysis of Machine Learning and Hybrid Deep Learning Models," Heliyon, vol. 9, no. 9, article no. e20281, September 2023.

[13] A. Umair, E. Masciari, and M. H. Ullah, "Vaccine Sentiment Analysis Using BERT + NBSVM and Geo-Spatial Approaches," The Journal of Supercomputing, vol. 79, no. 15, pp. 17355-17385, October 2023.

[14] G. B. Mohammad, S. Potluri, A. Kumar, R. Kumar, P. Dileep, R. Tiwari, et al., "An Artificial Intelligence-Based Reactive Health Care System for Emotion Detections," Computational Intelligence and Neuroscience, vol. 2022, article no. 8787023, 2022,
[15] K. Denecke and D. Reichenpfader, "Sentiment Analysis of Clinical Narratives: A Scoping Review," Journal of Biomedical Informatics, vol. 140, article no. 104336, April 2023.
[16] S. Gohil, S. Vuik, and A. Darzi, "Sentiment Analysis of Health Care Tweets: Review of the Methods Used," JMIR Public Health and Surveillance, vol. 4, no. 2, article no. e43, April-June 2018.
[17] P. Padmavathy and S. Pakkir Mohideen, "An Efficient Two-Pass Classifier System for Patient Opinion Mining to Analyze Drugs Satisfaction," Biomedical Signal Processing and Control, vol. 57, article no. 101755, March 2020.
[18] Y. Bhangdia, R. Bhansali, N. Chaudhari, D. Chandnani, and M. L. Dhore, "Speech Emotion Recognition and Sentiment Analysis based Therapist Bot," Third International Conference on Inventive Research in Computing Applications, pp. 96-101, September 2021.
[19] A. Saranya and R. Subhashini, "A Systematic Review of Explainable Artificial Intelligence Models and Applications: Recent Developments and Future Trends," Decision Analytics Journal, vol. 7, article no. 100230, June 2023.
[20] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135-1144, August 2016.
[21] I. Mani and J. Zhang, "kNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction," Proceedings of Workshop on Learning From Imbalanced Datasets, pp.1-7, August 2003.
[22] A. Fernandez, S. Garcia, F. Herrera, and N.V. Chawla, "Smote for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," Journal of Artificial Intelligence Research, vol. 61, pp. 863-905, 2018.
[23] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive Synthetic Sampling Approach for Imbalanced Learning," IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322-1328, June 2008.
[24] Y. Zhang, R. Jin, and Z. H. Zhou, "Understanding Bag-of-Words Model: A Statistical Framework," International Journal of Machine Learning and Cybernetics, vol. 1, no. 1-4, pp. 43-52, December 2010.
[25] G. Salton and M. J. McGill, Introduction to Modern Information Retrieval, International Student ed., Auckland: McGraw-Hill International,1983.
[26] Y. R. Chao and G. K. Zipf, "Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology," Language, vol. 26, no. 3, pp. 394-401, July-September 1950.
[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," https://arxiv.org/pdf/1301.3781.pdf, January 16, 2013.
[28] C. Y. J. Peng, K. L. Lee, and G. M. Ingersoll, "An Introduction to Logistic Regression Analysis and Reporting," The Journal of Educational Research, vol. 96, no. 1, pp. 3-14, 2002.
[29] C. Sammut and G. Webb, Encyclopedia of Machine Learning and Data Mining, Living ed. Boston: Springer, 2016.
[30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," Advances in Neural Information Processing Systems 30 (NIPS 2017), pp. 1-9, December 2017.