Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/68/20241448

BXCNN: A novel depression detection model

Haocheng Xi[1,2], **Haoyang Zhong**[1,3], **Yutong Wang**[1,4]
[1] School of Information Science and Engineering, Yunnan University, Yunnan 650500, China

[2] shaunspike813jelly@gmail.com
[3] aitanxiandeqimiao@gmail.com
[4] eggw81616@gmail.com

Abstract. In response to the mounting societal pressures, an increasing number of individuals grappling with mental health challenges are turning to social media platforms to express their feelings. The utilization of deep learning models for analyzing social media data has become increasingly crucial in detecting early signs of depression. Early intervention through depression detection can significantly enhance patients' quality of life and even save lives. However, many existing deep learning models suffer from low prediction accuracy, exacerbated by the imbalance between positive and negative samples in the collected data. To address these challenges, we propose a novel depression detection model integrated with BERT, XGBoost, and Convolutional Neural Networks (BXCNN). This model harnesses the advantages of ensemble learning and deep learning technologies by integrating XGBoost for feature extraction to alleviate data imbalance and CNN for classification. We transform depression-related textual data into sentence vectors using BERT to capture semantic information effectively. These features are then fed into a CNN classifier to accurately predict the likelihood of individuals exhibiting depressive symptoms. Through empirical evaluations on relevant datasets, our approach excels across various evaluation metrics, including accuracy, precision, recall, F1 score, and the area under the receiver operating characteristic curve (AUC-ROC).

Keywords: Early depression detection, Ensemble model, Convolutional neural network (CNN), Data imbalance

1. Introduction

In the contemporary landscape of social media inundated with emotional expression, an increasing amount of research endeavors to detect signs of depression in individuals through the analysis of their extensive textual interactions. However, conventional text embedding techniques like GloVe and Word2Vec often struggle to sufficiently capture the semantic nuances of text, posing challenges in accurately identifying depressive inclinations. Furthermore, social media datasets frequently exhibit issues of data imbalance, potentially introducing bias and reducing model accuracy. Consequently, a pivotal question emerges: How can existing textual data and advanced technologies be leveraged to develop a precise and efficient model for detecting tendencies toward depression?

To tackle this challenge, we propose an innovative model: BXCNN. This model integrates the robust semantic representation capabilities of the BERT model, enhancing its understanding of emotional nuances within the text through fine-tuning. Concurrently, we employ the XGBoost algorithm to address data imbalance issues, thereby enhancing the model's robustness and generalization capacity. Additionally, we utilize Convolutional Neural Networks (CNNs) to extract local text features, facilitating a more comprehensive detection of potential depressive tendencies within the text. Through this multifaceted approach, our model not only surmounts the limitations of traditional methods but also demonstrates superior performance in depression tendency detection tasks.

Consequently, our study contributes significantly to the advancement of depression tendency detection. Firstly, we introduce an advanced BERT-fine-tuning model that optimally exploits its capacity to comprehend text semantics, thus elevating the model's performance and accuracy. Secondly, we mitigate data imbalance issues through the implementation of the XGBoost algorithm, enhancing the stability and reliability of the model. Finally, we validate the efficacy of our model through experiments, with results indicating its notable advantages in depression tendency detection tasks. These findings offer crucial insights and guidance for future clinical applications and research endeavors.

2. Related Work

Previous studies have delved into various dimensions of depression detection on social media. Early investigations concentrated on employing feature engineering methodologies, exemplified by the work of Stankevich et al.[16], who identified depressive tendencies associated with diminished social activity frequency, heightened negative emotions, increased self-focus, and a pronounced emphasis on religion. Similarly, the study conducted by Park et al.[12] analyzed Twitter discourse of individuals with depressive tendencies, revealing a propensity for negative or reactive posts alongside a notable decline in social interactions. These seminal studies furnish crucial insights and foundational principles for the detection of depression tendencies via social media.

As technology progresses, attention has shifted towards traditional machine learning techniques like Support Vector Machines (SVM). For instance, Xu et al. [19] employed an SVM model to project social media user data into a high-dimensional space, effectively discriminating between depressed and non-depressed individuals. Nonetheless, SVM encounters practical limitations such as reliance on feature selection, challenges in handling high-dimensional data, susceptibility to overfitting, and parameter sensitivity.

Furthermore, alternative approaches leverage ensemble learning algorithms, such as Random Forests (RF). For instance, Reece et al.[14] employed a random forest model to investigate the feasibility of utilizing Twitter data for depression detection, demonstrating the potential for accurate predictions months before formal diagnosis. Nevertheless, the random forest method faces practical challenges, including parameter sensitivity, limited model interpretability, and vulnerability to overfitting.

It is noteworthy that Li et al.[9] introduced a hybrid model merging Convolutional Neural Networks (CNN) and XGBoost to forecast social media content popularity. Nonetheless, their approach encounters limitations in text processing. CNNs can solely accommodate multidimensional word vectors and are unable to leverage advanced techniques like BERT, leading to issues of high-dimensional sparse representation and semantic information loss, thereby constraining model performance and generalization capabilities. Consequently, delving into more intricate feature mapping methodologies and effective feature selection techniques holds the potential to enhance model performance and generalization.

When confronted with real-world data imbalance challenges, researchers have proposed various solutions. For instance, Rula Kamil et al.[7], devised a hybrid model amalgamating Bi-LSTM and XGBoost models to predict depression. However, some prior methodologies may exhibit drawbacks such as overfitting during model evaluation. Hence, further refinement of these methods is essential to bolster model performance and robustness.

Finally, we were greatly intrigued by the approach proposed by Cong et al.[6], which integrates XGBoost into their methodology. However, upon closer examination, we identified certain limitations. Their innovative utilization of XGBoost to address the highly imbalanced nature of depression datasets is commendable. By capitalizing on XGBoost's inherent ability to assign greater weight to negative samples while effectively managing a plethora of positive samples, they aim to enhance model performance. Nevertheless, upon scrutinizing their methodology in detail, we uncovered some significant flaws. Specifically, their practice of partitioning the raw dataset and training XGBoost with a segmented training set, followed by the use of the entire raw dataset as a test set for model evaluation, is problematic. This oversight introduces a considerable bias since the model has been exposed to test data during training. Consequently, upon replicating their approach, we noted a disproportionately high number of favorable performance indicators, hinting at potential data leakage issues. Essentially, it resembles a scenario where a student gains prior knowledge of exam questions, resulting in an unusually high score. Therefore, while we drew inspiration from their method, we opted for a more methodical approach to conducting our experiments.

3. Methodology

Our proposed Depression Detection Model adopts a hybrid approach that integrates BERT-fine-tuning, CNN, and XGBoost, with the objective of harnessing the complementary strengths of deep learning and ensemble learning techniques for precise and resilient depression detection from textual data. The model architecture is illustrated in Figure 1 below.

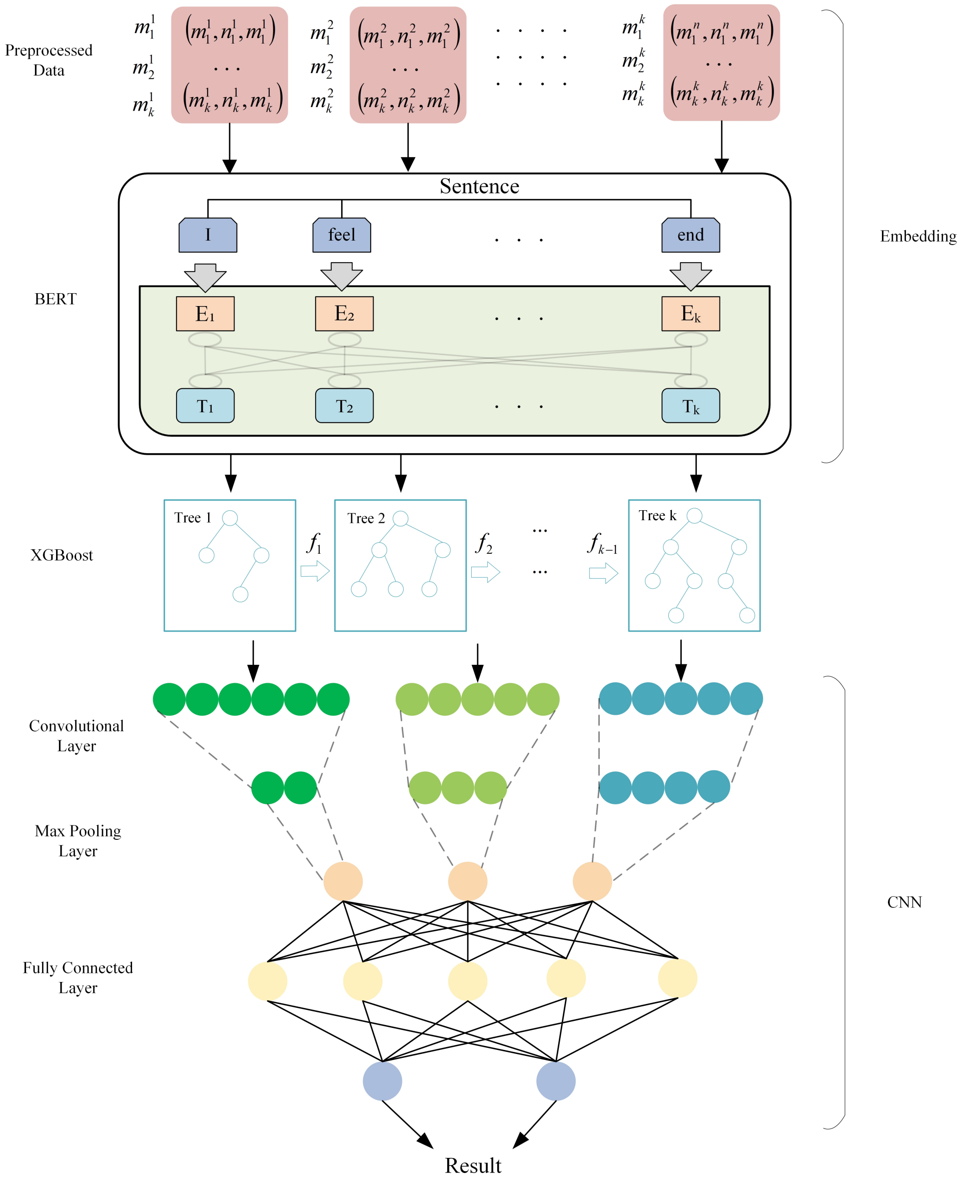

Figure 1. Our Whole Model

3.1. BERT-fine-tuning

In the text processing stage with BERT, the initial step entails tokenization, wherein each input text is decomposed into a sequence of words or subwords[13]. This process, represented by Tokenizer$(text) = \{w_1, w_2, ..., w_n\}$, is subsequently augmented by the addition of special tokens `[CLS]` and `[SEP]` to the tokenized sequence, serving to denote the commencement and conclusion of the sentence, respectively:

$$\text{Tokenizer}(text) = \{[\text{CLS}], w_1, w_2, ..., w_n, [\text{SEP}]\} \tag{1}$$

Subsequently, each token undergoes transformation into a word vector utilizing a pre-trained word embedding model, such as WordPiece or Byte Pair Encoding, denoted as:

$$\text{Word Embeddings} = \{\text{Embed}(w_1), \text{Embed}(w_2), ..., \text{Embed}(w_n)\} \tag{2}$$

For tasks involving sentence pair classification, segment embeddings are allocated to each token to signify its associated sentence, typically denoted as 0 for the first sentence and 1 for the second:

$$\text{Segment Embeddings} = \{s_1, s_2, ..., s_n\} \tag{3}$$

To incorporate positional information, positional embeddings are appended to the representation of each token, thereby informing BERT of its position within the input sequence:

$$\text{Positional Embeddings} = \{\text{PosEmbed}(1), \text{PosEmbed}(2), ..., \text{PosEmbed}(n)\} \tag{4}$$

Finally, the sentence embedding is generated by aggregating the word embeddings, segment embeddings, and positional embeddings:

$$\begin{gathered}\text{Sentence Embedding} = \\ \text{Word Embeddings} + \text{Segment Embeddings} + \text{Positional Embeddings}\end{gathered} \tag{5}$$

During the final fine-tuning phase, we feed the generated Sentence Embedding into the model's classifier for task-specific refinement tailored to the depression detection task. This fine-tuning process entails training a pre-trained BERT model with task-specific labeled data, iteratively updating the model's parameters via back-propagation and gradient descent algorithms. This iterative refinement enables the model to develop a deeper understanding of and distinguish depression-related textual features. This stage is pivotal as it allows the model to adapt more effectively to the nuances of the depression detection task by learning from task-specific data. Consequently, this approach facilitates the enhancement of the model's performance in depression detection, thereby enabling more accurate identification of texts indicative of depression tendencies.

3.2. XGBoost Model

XGBoost[5]: an optimized distributed gradient boosting library that extends the gradient boosting framework, is employed to construct a robust classifier using the extracted features from BERT. XGBoost aggregates predictions from multiple weak learners (decision trees) to achieve accurate predictions while mitigating overfitting. It leverages an ensemble of K classification and regression trees (CART), each comprising T nodes. The final prediction is the summation of prediction scores from each tree:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{6}$$

Here, x_i an instance from the training set, with y_i denoting its corresponding class label. f_k signifies the k-th tree, and F denotes the collection of all scores from the ensemble of CARTs. Regularization techniques are applied to enhance the final outcomes:

$$\text{Obj} = \sum_i l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{7}$$

In this equation, the first term, denoted as l signifies a differentiable loss function employed to quantify the disparity between the target y_i and the predicted value $\hat{y}_i$. The second term serves to prevent overfitting by penalizing the complexity of the model:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|w\|^2 \tag{8}$$

Here, γ represents a constant that regulates the strength of regularization, T denotes the number of leaf nodes in the tree, and w signifies the weight associated with each leaf. Gradient boosting (GB) demonstrates efficacy in both regression and classification tasks. GB, coupled with the loss function, undergoes expansion through a second-order Taylor series, eliminating constant terms to yield a simplified objective at step t, as depicted below:

$$\text{Obj}^{(t)} = \sum_i l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{9}$$

In this equation, $\hat{y}_i^{(t-1)}$ represents the predictions from previous $t - 1$ iterations. The function l computes the first and second-order gradient statistics of the loss function. The optimal weights $w_j^{(q)}$ for a given tree structure, leaf j, and tree structure q are calculated using the gain metric:

$$\text{gain} = \frac{1}{2}\left[\frac{(\sum_{\text{left}} g)^2}{\sum_{\text{left}} h + \lambda} + \frac{\left(\sum_{\text{right}} g\right)^2}{\sum_{\text{right}} h + \lambda} - \frac{(\sum_{\text{total}} g)^2}{\sum_{\text{total}} h + \lambda}\right] - \gamma \tag{10}$$

In practice, the evaluation of candidate splits involves utilizing the scores score $\text{score}_{\text{left}}$ and weights w_{left} of instances in the left instance set, along with the scores $\text{score}_{\text{right}}$ and weights w_{right} of instances in the right instance set. The gain for a split, where ”gain” signifies the reduction in loss after the split, is computed as:

$$\text{gain} = \frac{1}{2}\left[\frac{\text{score}_{\text{left}}^2}{w_{\text{left}} + \lambda} + \frac{\text{score}_{\text{right}}^2}{w_{\text{right}} + \lambda} - \frac{\text{score}_{\text{total}}^2}{w_{\text{total}} + \lambda}\right] - \gamma \tag{11}$$

Building upon the advancements made by Cong et al.[6], we implement the following training steps: Firstly, we employ the XGBoost classifier to filter the sentence vectors already processed by BERT, which have been partitioned into an 80% training set and a 20% test set. Initially, the XGBoost model is trained with 80% of the training set, while reserving 20% for testing purposes. Subsequently, in the second step, all sentence vectors are utilized to train the XGBoost model. Samples predicted as negative by XGBoost are discarded, whereas positive predictions are further split into an 80% training set and a 20% test set for training the CNN model, with the test set being used for CNN model evaluation. At this stage, both positive and negative examples are outputted based on predictions. The third step involves utilizing the first 20% of the test set for input into XGBoost for prediction. If the output is positive, it is discarded; otherwise, it is considered as output. The original imbalance ratio of positive to negative samples in the erisk2017depression dataset was 9.87:1 (S positive:S negative). Following XGBoost classification, the ratio of true positive (TP) to false positive (FP) was reduced to 3.85:1, indicating a partial alleviation of the imbalance between positive and negative samples.

A comprehensive overview of how XGBoost addresses the data imbalance issue and our enhancements are illustrated in Figure 2.

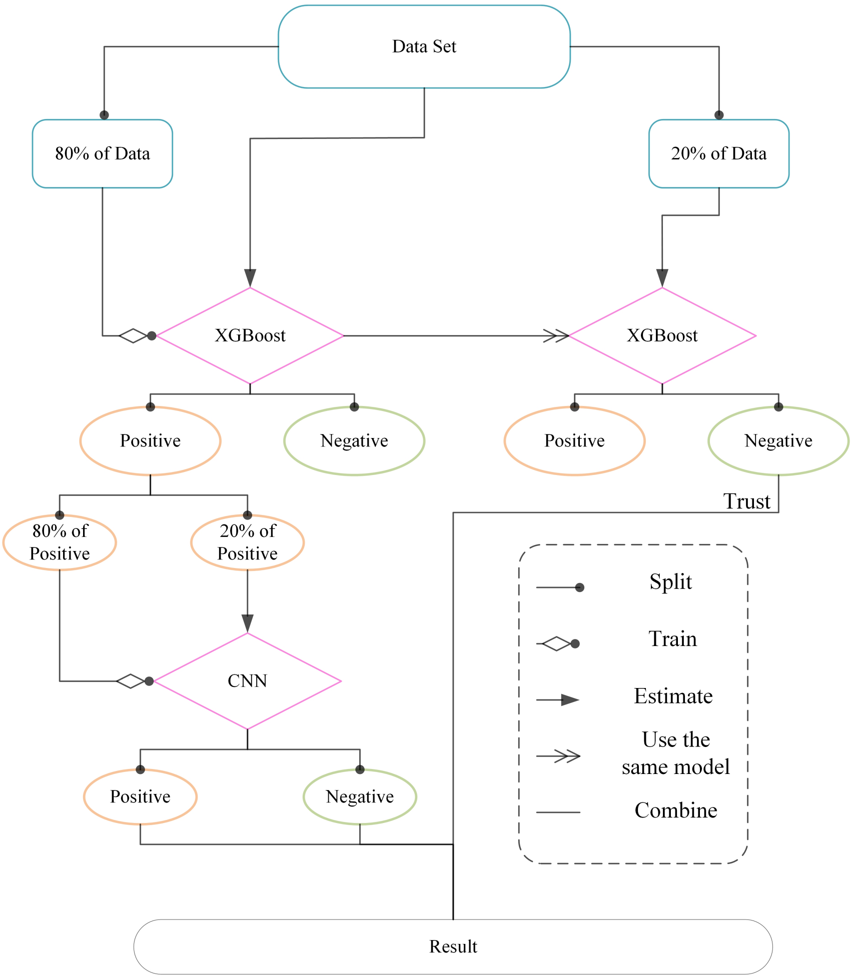

Figure 2. Our XGBoost

We employ the XGBoost algorithm to augment the base model. Given a training dataset $T = \{(x_i, y_i)\}, i = 1, 2, ..., N$, the steps of the ensemble learning model XGBoost can be outlined as follows:

(i) Initialize the weight distribution of the training data. Each training sample is initially given the same weight $\frac{1}{N}$. The first weight distribution D_1 is calculated as follows:

$$D_1 = (w_{11}, w_{12}, ..., w_{1i}, ..., w_{1N}), \quad w_{1i} = \frac{1}{N}, \quad (i = 1, 2, ..., N) \tag{12}$$

where N is the number of training samples.

(ii) Update the weight distribution and conduct multiple rounds of iteration. $m = 1, 2, ..., M$ denotes the round of iteration. $G_m(x)$ represents the base model trained by the weight distribution D_m. The error rate of $G_m(x)$ on the training data is calculated as:

$$e_m = \frac{\sum_{i=1}^{N} w_{mi} \cdot I(G_m(x_i) \neq y_i)}{\sum_{i=1}^{N} w_{mi}} \tag{13}$$

where e_m is the sum of the weights of the samples misclassified by $G_m(x)$. The importance of $G_m(x)$ in the final classifier is determined by:

$$\alpha_m = \frac{1}{2} \ln \frac{1 - e_m}{e_m}$$

When $e_m \leq \frac{1}{2}$, $\alpha_m \geq 0$. α_m increases with the decrease of e_m. The weight is updated as follows:

$$w_{m+1,i} = \frac{w_{mi} \cdot \exp\left(-\alpha_m \cdot y_i \cdot G_m(x_i)\right)}{Z_m}, \quad (i = 1, 2, ..., N) \tag{14}$$

where $Z_m = \sum_{i=1}^{N} w_{mi} \cdot \exp\left(-\alpha_m \cdot y_i \cdot G_m(x_i)\right)$ is a normalization factor.
(iii) The ensemble learning model is formed by combining all base models:

$$G(x) = \text{sign}(f(x)) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m \cdot G_m(x)\right) \tag{15}$$

3.3. CNN Model

We assume the input is the processed result of sentence vectors from XGBoost output, represented as:

$$X = \{x_1, x_2, ..., x_n\} \tag{16}$$

where x_i represents the feature vector of the i-th sentence.

The convolution[1] operation with a filter (or kernel) W on the input X produces a feature map Z from the sentences by applying the filter. The filter moves over the sentence vectors with a certain stride, padding with zeros. Discrete convolution can be defined as:

$$Z_i^{(l)} = \phi\left(\sum_j \sum_k x_{j,k}^{(l-1)} * w_{i,j,k}^{(l)} + b_i^{(l)}\right) \tag{17}$$

where $Z_i^{(l)}$ represents the elements of the feature map in layer l. The Rectified Linear Unit (ReLU) activation function ϕ, bias matrix $b_i^{(l)}$, and filter $w_{i,j,k}^{(l)}$ of size K are involved.

Thus, the elements of the output of layer l, for feature map i at position (p, q), are:

$$Z_{i,p,q}^{(l)} = \phi\left(\sum_j \sum_k x_{j,p+k-1,q+k-1}^{(l-1)} \cdot w_{i,j,k}^{(l)} + b_i^{(l)}\right) \tag{18}$$

The pooling layer further modifies the output of the layer by downsampling and helps prevent overfitting. It substitutes the output with the maximum or average value within a rectangular neighborhood. For instance, if Y is the output of the previous layer with ReLU activation, then Y' is a pooling function acting on Y through a pool window with a stride and pool size. Typically, the pooling operation positions the window at non-overlapping positions in each feature map and retains one value per window to subsample the feature map. Two common types of pooling are average pooling and max pooling. The output of the max-pooling function becomes:

$$Y'_{i,p,q} = \max\left(Y_{i,p+(k-1)s,q+(k-1)s}\right) \tag{19}$$

where the function is applied across the dimensions of the max-pooling window.

The fully connected (FC) layer represents the final layer of the CNN architecture. The output of the preceding pooling layer is flattened into a single column vector and serves as the input to this layer. In the FC layer, each neuron from the previous layer is connected to every neuron in the subsequent layer. Typically, the equations used in multilayer perceptrons are employed in the FC layer. Let L denote the number of FC layers, K denote the number of feature maps of size N (following the notation used by Stutz), and M denote the number of neurons in layer L. The computation of feature maps in layer L is expressed as:

$$Y_m^{(L)} = \phi\left(\sum_i \sum_{p,q} w_{m,i,p,q}^{(L)} \cdot Y_{i,p,q}^{(L-1)} + b_m^{(L)}\right) \tag{20}$$

where $w_{m,i,p,q}^{(L)}$ represents the weight connecting neuron i in feature map m of layer L to position (p, q) in feature map i of layer $L - 1$. In practice, ϕ is a transformation function utilized for multi-class prediction, and dropout regularization is implemented in the FC layer to reduce the number of neurons, thereby mitigating overfitting.

By amalgamating the feature extraction capabilities of XGBoost with the ensemble learning prowess of CNN[17], our proposed model presents a comprehensive approach to depression detection, effectively harnessing textual information to identify individuals at risk of depression. Through empirical evaluation and fine-tuning on relevant datasets, we demonstrate the effectiveness and practical applicability of our hybrid model for real-world deployment in mental health care settings and digital platforms.

4. Experiments and Results

4.1. Dataset

Our study utilized datasets from the eRisk 2017 evaluation tasks (Losada et al., 2017)[11] for depression detection. These datasets comprised posts from various users, with those explicitly stating a diagnosis of depression classified as positive cases. Expressions such as ”I think I have depression” were disregarded, and the remaining posts were labeled as negative instances. The dataset encompasses sequential posts and comments from Reddit, encompassing users diagnosed with depression and users in a random control group. Users with depression were identified through posts explicitly mentioning their diagnosis. Usernames were anonymized using IDs such as ”train subject 1”. Each message in the dataset may consist of a title, text, or both, depending on its type. Users can post content based on images or URLs (included in the title only), textual content (title and optional text), or comments on other messages (text only). Several messages were discarded due to being entirely empty. Additionally, each message includes a date attribute indicating the timestamp of the user message posting that is accurate to the second. However, since control group users were selected based on recent posting activity, timestamp information was disregarded in our analysis. Simultaneously, we transform all data in URLs to text format to enable the BERT Model to process the data.

The detailed statistics are shown in Table 1.

Table 1. The depression dataset.

Item	Depressed users	Non-depressed users
Training	674	3214
Testing	156	816
Total	830	4030

4.2. Experiment Setting

Before feeding the data into the models, we conducted preprocessing steps, which included tokenization, lowercasing, and removing stop words and punctuation marks.

We employed BERT (Bidirectional Encoder Representations from Transformers) to convert each textual input into sentence vectors. Fine-tuning was utilized to adapt BERT to the specific task of depression detection, enabling it to capture contextual information relevant to our classification task.

The processed sentence vectors were then input into a hybrid model consisting of XGBoost and CNN. XGBoost served as a feature extractor to effectively handle data imbalance and capture complex patterns in the data. After meticulously tuning the hyperparameters of the XGBoost model component using cross-validation, the output was subsequently passed to a CNN architecture, comprising a convolutional layer and max-pooling operation. Hyperparameters such as epochs, embedding size, filter size, and learning rate were adjusted to optimize performance.

In the convolutional layer, we utilized a filter size of 3 to capture trigram features, with a stride of 1 to maximize feature extraction while retaining spatial information. A ReLU activation function was introduced to introduce non-linearity, facilitating complex pattern detection.

By selecting the maximum activation within each window, we emphasized salient features, achieving down-sampling of feature maps through max-pooling and reducing computational complexity. The flattened feature maps were then fed into fully connected layers, promoting hierarchical feature learning.

Dropout regularization in the fully connected layers prevented overfitting by randomly deactivating neurons during training, enhancing robustness and facilitating generalized feature learning.

4.3. Evaluation Metrics

Evaluation metrics include accuracy, precision, recall, F1 score, area under the ROC curve (AUC), and Kolmogorov-Smirnov (KS) statistic, providing a comprehensive assessment of classifier performance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{21}$$

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{24}$$

$$AUC = \int_0^1 ROC(x)\, dx \tag{25}$$

$$KS = \max(TPR - FPR) \tag{26}$$

4.4. Baseline

We designed three sets of controls: The first group utilized the detection results of our team and several other teams on the eRisk 2017 depression dataset as a comparison. The second group compared the combination order of our model with other models designed by us. For instance, instead of using BERT embedding, we employed GloVe as word embedding to convert each word of each sentence into a word vector, resulting in a two-dimensional word vector for each sentence. This vectorization allowed for training with CNN or CNN+XGBoost. The third group employed BERT in combination with other traditional machine learning models to make predictions about this depression dataset. The comparison results of the three groups are presented in Table 2. By comparing with the other baselines, it can be observed that our model demonstrates superior performance.

Our overall experimental process is depicted in Figure 3:

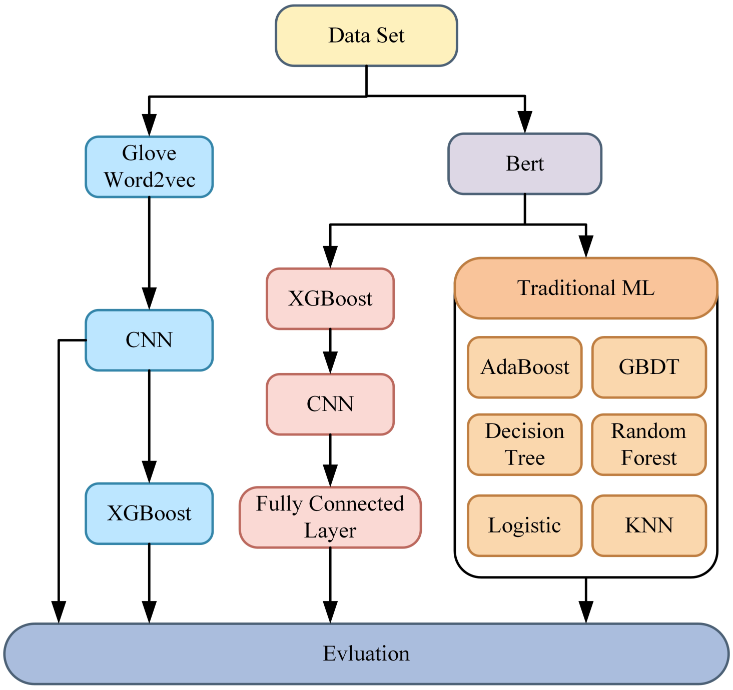

Figure 3. Our Flow Path

4.4.1. Results from other Teams 1)**FHDO-BCSGA**[15]: The UArizona team leveraged external information beyond the small training set, incorporating a preexisting depression lexicon and concepts from the Unified Medical Language System as features. For prediction, they utilized an SVM model trained using Weka with the depression lexicon as features. For convenience, we refer to this model as WSLexicon.

4.4.2. Results from other Traditional ML Models We employ traditional machine learning approaches following the BERT preprocessing step. Specifically, we implement the following baseline models:

1)**KNN**[4]: A method that classifies by finding the K nearest training samples to the input, deciding based on their outputs through voting or averaging. It's intuitive and performs well on small datasets or simple classification problems.

2)**Random Forest (RF)**[3]: An ensemble learning model built using multiple decision trees, which improves accuracy and robustness by voting or averaging predictions.

3)**Adaboost**[20]: Sequentially trains a series of weak classifiers, where each classifier attempts to correct the errors of its predecessor, ultimately forming a strong classifier through weighted combination.

4)**Logistic Regression**[18]: A binary classification algorithm that estimates the probability of a binary outcome based on one or more predictor variables. It models the relationship between the independent variables and the categorical dependent variable using a logistic function.

5)**Decision Tree (DT)**[10]: A tree-like model that classifies or predicts by recursively partitioning the dataset.

6)**Gradient Boosting Decision Trees (GBDT)**[2][8]: An ensemble learning model that sequentially trains decision trees to correct errors made by the previous trees, gradually improving overall performance. GBDT minimizes the gradient of the loss function at each step. It excels in various machine learning tasks including classification and regression.

4.4.3. Results from Our Models We rearrange the sequence of operations in our proposed model, opting not to use BERT for sentence vector conversion of depression text. Instead, we employ glove as Word embedding to convert each word of each sentence into a word vector, resulting in a two-dimensional word vector for each sentence. For an $n \times v$ word vector matrix $V(d_i)$, where v represents the word vector dimension: $V(\mathbf{d}_i) = \begin{bmatrix} t_{11,1} & t_{12,1} & \cdots & t_{1v,1} \\ t_{21,2} & t_{22,2} & \cdots & t_{2v,2} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1,v} & t_{n2,v} & \cdots & t_{nv,v} \end{bmatrix}$

The weight of each word vector in document d_i is computed, capturing the relationship between individual words and the entire document. This process yielded an $n \times v$ weight matrix $t_i^{tf\_idf}(d_i) = \begin{bmatrix} t_i^{tfidf}(t_1) & \cdots & t_i^{tfidf}(t_n) \\ t_i^{tfidf}(t_2) & \cdots & t_i^{tfidf}(t_n) \\ \vdots & \ddots & \vdots \\ t_i^{tfidf}(t_v) & \cdots & t_i^{tfidf}(t_n) \end{bmatrix}$ Then, the final vector-matrix $\text{Vec}(d_i)$ of depression text document d_i is expressed as the dot product of the two matrices in the aforementioned and the following Formula:

$$\text{Vec}(d_i) = V(d_i) \cdot t_i^{tf\_idf}(d_i) \tag{27}$$

After preprocessing, we apply CNN for reduction and feature extraction, and finally use XGBoost for prediction. This sequential approach aims to compare against the proposed model's original configuration. The model of this sequence is called Glove+CNN+XGBoost (or GCX). Simultaneously, we attempted to use only BERT+XGBoost (or BX) and only Glove+CNN+FCL (or GCF) as well, which respectively serve to evaluate the effect of CNN + Fully connected layer and XGBoost.

These models serve as benchmarks to evaluate the performance of our proposed BERT+XGBoost+CNN model in terms of accuracy, recall, precision, F1 score, area under the ROC curve (AUC), and Kolmogorov-Smirnov (KS) statistic.

Table 2. Performance of Different Methods for Depression Text Classification Prediction

Group	Method	P%	R%	F1%	ACC%	AUC%	KS%
Ours	**BXCNN**	**85.56**	33.77	**48.43**	84.32	**66.09**	**32.18**
Group 1	WSLexicon	33.00	27.00	30.00	-	-	-
Group 2	KNN	14.93	**38.46**	21.51	54.93	48.27	3.45
	RF	19.98	7.56	11.23	83.95	50.00	2.36
	Adaboost	23.81	6.41	10.10	81.69	51.24	2.49
	Logistic	16.67	13.46	14.89	75.31	50.30	0.59
	DT	11.76	14.10	12.83	69.24	46.94	6.12
	GBDT	20.16	16.65	18.56	83.74	49.88	0.25
Group 3	GCX	75.63	28.16	39.56	83.95	50.00	6.23
	BX	13.33	1.28	2.34	82.81	49.84	0.31
	GCF	71.03	30.01	47.52	**85.60**	64.24	30.47

(In this modified table, the best performance is highlighted in bold, and the second-best performance is underlined.)

4.5. Results

Our groundbreaking BERT-fine-tuning+XGBoost+CNN+FCL model surpasses conventional machine learning techniques and ensemble methods in depression detection, as demonstrated in Table 2. Through comprehensive evaluation, including metrics such as accuracy, recall, precision, F1 score, AUC, and KS statistic, our model exhibits exceptional performance. While the Accuracy value

(84.32%) is slightly lower than that of the Glove+CNN+FCL Model (85.60%), and the Recall value (33.77%) trails behind the BERT+KNN Model (38.46%), when considering all metrics collectively, our model demonstrates outstanding performance with Precision=85.56%, F1=48.43%, Acc=84.32%, AUC=66.09%, and KS=32.18%. Notably, the key performance index F1 of our model is 1.91% higher than that of the optimal F1 in the control group, and the Precision index is 20.46% higher than that of the optimal F1 in the control group. This highlights its innovative capability, offering a promising approach for precise and effective identification of depressive tendencies in textual data.

5. Conclusion

In conclusion, we introduce a novel Depression Detection Model that integrates BERT-fine-tuning, XGBoost, and CNN to predict depressive tendencies from text data. Through rigorous experimentation and comparison, we have demonstrated the effectiveness and superiority of our proposed model over traditional machine learning techniques and ensemble methods. Our innovative approach leverages XGBoost for feature extraction and adopts more rational training and testing methods, effectively addressing the challenge of data imbalance. Furthermore, the subsequent CNN contributes to more accurate depression prediction. Through a series of comparative experiments, we have verified the robustness and versatility of our model. The comprehensive evaluation metrics, including accuracy, recall, precision, F1 score, AUC, and KS statistic, underscore the reliability and efficiency of our model in detecting depressive tendencies. Overall, our research contributes to the advancement of depression detection techniques, providing valuable insights for potential clinical applications and mental health interventions. Additionally, our research paves the way for further exploration and development in the field of mental health informatics. Future work could focus on refining and optimizing the model architecture, exploring additional data sources and features, and conducting large-scale validation studies in diverse populations.

References

[1] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.

[2] Vanishri Arun, V Prajwal, Murali Krishna, BV Arunkumar, SK Padma, and V Shyam. A boosted machine learning approach for detection of depression. In *2018 IEEE symposium series on computational intelligence (SSCI)*, pages 41–47. IEEE, 2018.

[3] Fidel Cacheda, Diego Fernandez, Francisco J Novoa, and Victor Carneiro. Early detection of depression: social network analysis and random forest techniques. *Journal of medical Internet research*, 21(6):e12554, 2019.

[4] Fidel Cacheda, Diego Fernández Iglesias, Francisco Javier Nóvoa, and Victor Carneiro. Analysis and experiments on early detection of depression. *CLEF (Working Notes)*, 2125:43, 2018.

[5] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[6] Qing Cong, Zhiyong Feng, Fang Li, Yang Xiang, Guozheng Rao, and Cui Tao. Xa-bilstm: a deep learning approach for depression detection in imbalanced data. In *2018 IEEE international conference on bioinformatics and biomedicine (BIBM)*, pages 1624–1627. IEEE, 2018.

[7] Rula Kamil and Ayad R Abbas. Predicating depression on twitter using hybrid model bilstm-xgboost. *Bulletin of Electrical Engineering and Informatics*, 12(6):3620–3627, 2023.

[8] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[9] Liuwu Li, Runwei Situ, Junyan Gao, Zhenguo Yang, and Wenyin Liu. A hybrid model combining convolutional neural network with xgboost for predicting social media popularity. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1912–1917, 2017.

[10] Zhenyu Liu, Dongyu Wang, Lan Zhang, and Bin Hu. A novel decision tree for depression recognition in speech. *arXiv preprint arXiv:2002.12759*, 2020.

[11] David E Losada, Fabio Crestani, and Javier Parapar. erisk 2017: Clef lab on early risk prediction on the internet: experimental foundations. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11–14, 2017, Proceedings 8*, pages 346–360. Springer, 2017.

[12] Minsu Park, Chiyoung Cha, and Meeyoung Cha. Depressive moods of users portrayed in twitter. In *Proceedings of the 18th ACM International Conference on Knowledge Discovery and Data Mining, SIGKDD 2012*, pages 1–8, 2012.

[13] Giovanni Puccetti, Alessio Miaschi, and Felice Dell'Orletta. How do bert embeddings organize linguistic knowledge? In *Proceedings of deep learning inside out (DeeLIO): the 2nd workshop on knowledge extraction and integration for deep learning architectures*, pages 48–57, 2021.

[14] Andrew G Reece, Andrew J Reagan, Katharina LM Lix, Peter Sheridan Dodds, Christopher M Danforth, and Ellen J Langer. Forecasting the onset and course of mental illness with twitter data. *Scientific reports*, 7(1):13006, 2017.

[15] Farig Sadeque, Dongfang Xu, and Steven Bethard. Uarizona at the clef erisk 2017 pilot task: linear and recurrent models for early depression detection. In *CEUR workshop proceedings*, volume 1866. NIH Public Access, 2017.

[16] Maxim Stankevich, Vadim Isakov, Dmitry Devyatkin, and Ivan V Smirnov. Feature engineering for depression detection in social media. In *ICPRAM*, pages 426–431, 2018.

[17] Setthanun Thongsuwan, Saichon Jaiyen, Anantachai Padcharoen, and Praveen Agarwal. Convxgb: A new deep learning model for classification problems based on cnn and xgboost. *Nuclear Engineering and Technology*, 53(2):522–531, 2021.

[18] Christiana Sri Wahyuningsih, Achmad Arman Subijanto, and Bhisma Murti. Logistic regression on factors affecting depression among the elderly. *Journal of Epidemiology and Public Health*, 4(3):171–179, 2019.

[19] Ronghua Xu and Qingpeng Zhang. Understanding online health groups for depression: social network and linguistic perspectives. *Journal of medical Internet research*, 18(3):e63, 2016.

[20] Jingwen Zhang, Haochen Yin, Jinfang Wang, Shuxin Luan, and Chang Liu. Severe major depression disorders detection using adaboost-collaborative representation classification method. In *2018 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, pages 584–588. IEEE, 2018.