Feature Extraction and Prediction of Combined Text and Survey Data using Two-Staged Modeling

Asif Ahmed Neloy
Department of Computer Science
University of Manitoba
Winnipeg, Canada
asif.neloy@umanitoba.ca

Maxime Turgeon
Department of Statistics
University of Manitoba
Winnipeg, Canada
max.turgeon@umanitoba.ca

***Abstract*—Deep learning (DL) based natural language processing (NLP) has recently grown as one the fastest research domain and retained remarkable improvement in many applications. Due to the significant amount of data, the adaptation of feature learning and symmetric data efficiency is a critical underlying task in such applications. However, their ability to extract features is limited due to a lack of proper model formation. Moreover, the use of these methods on smaller datasets is unexplored and underdeveloped compared to more popular research areas. This work introduces a two-stage modeling approach to combine classical statistical analysis with NLP problems in a real-world dataset. We effectively layout a combination of the classical statistical model incorporating a stacked ensemble classifier and a DL framework of convolutional neural network (CNN) and Bidirectional Recurrent Neural Networks (Bi-RNN) to structure a more decomposed architecture with lower computational complexity. Additionally, the experimental results illustrating 96.69% training and 70.56% testing accuracy and hypothesis testing from our DL models followed by an ablation study empirically demonstrate the validation of our proposed combined modeling technique.**

***Index Terms*—convolutional neural network, bidirectional recurrent neural networks, long short-term memory, natural language processing, ensemble learning.**

I. Introduction

High-quality domain-specific DL frameworks in NLP research are in high demand, whereas general-purpose DL models have limited applications. Common NLP tasks are already addressed using deep models, including text preprocessing, embedding/representation, classification, and sentiment analysis. As a consequence, many fields of application have also benefited from these models, such as language analysis, cybersecurity, customer segmentation, and recommender systems. However, perceiving a unique architecture that will perform all classes of classification and feature extraction is complex. The text classification problem, including a question, survey, or topic classification, depends on the target word and the relation between each targeted feature. Although traditional text classification methods have shown reasonable performance, their feature extracting and mapping abilities are seriously limited.

The key to DL frameworks' remarkable performance is that they efficiently learn complex representations of text data that can be combined with other statistical models to perform classification. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are the most utilized DL architectures in many applications. Text data can be construed as a sequence of data points; however, CNN models cannot directly learn sequential correlated latent vectors from such points. Hence, RNNs have been the tool of choice for simplier NLP tasks. However, due to their architecture, the performance tends to decrease for long vector sequences, gradient vanishing and exploding [1]. Long Short-Term Memory (LSTM) [2] is a special type of RNN that effectively mitigates such issues by utilizing long short-term hidden memory units. Recently, LSTMs with bi-directional units, attention mechanism, and label embedding attentive models are applied in various tasks and achieved good performance [3] [4] [5]. Despite showing promising results, DL-based methods are often criticized for lacking reproducibility issues, significant training parameters, complex architecture, and costing enormous computational resources [6] [7].

Most recently proposed DL and combined methods utilize synthetic data, large corpus, extracted or compiled web data [8]. Also, the majority of the samples are augmented in the training and testing phase to remove dependencies. The primary reason behind following a trivial process is that substantial DL architectures require large training samples for better accuracy. Smaller sample sizes often lead to over-fitting, biased and corrupted modeling. Moreover, structured datasets like surveys and collaborative study datasets are not publicly available for the researchers to comply with in different studies. As a result, large research areas of NLP and traditional statistical modeling are still unexplored. To summarize, we identified the following shortcomings:

- Extract correlated features from a survey dataset rather than creating a dictionary or summarized keywords.
- A cross-modal hypothesis testing for deep learning models to test the models' prediction.
- Emphasizing the feature filtering and sampling process rather than designing a DL framework with high computational cost and multiple training parameters.

To mitigate limitations mentioned above, we propose the following framework:

This research project is funded by NSERC CREATE The Visual and Automated Disease Analytics (VADA) program.

2375-9259/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDMW58026.2022.00064

- A novel two-stage modeling approach combining classification and prediction tasks with DL architecture that emphasizes feature extraction and utilizes models with low complexity.
- In our first stage of modeling, we adopted a staging classifier model with k-fold cross-validation to address the concern of effective feature extraction in a smaller dataset with fewer instances.
- Finally, we design a Bi-RNN framework to combine the final task of prediction and classification.

We give empirical evidence that, our proposed two-stage modeling approach demonstrates better results than traditional DL framework utilizing a lower number of training instances. Specifically, for the first-stage modeling, we successfully implemented convergence of hyper-parameter selection in a stacked ensemble learning method that achieves better results despite having weak predictors. Then, for the second-stage modeling, our designed CNN and Bi-RNN architecture exhibit better performance with lower computational complexity and cost.

The rest of the paper is organized as follows: Section II introduces the literature review and overview of this study, Section III establishes the fundamental concepts of our proposed two-stage modeling approach. We review the implementation and training parameters in Section IV. Finally, we summarize our experimental results in the section V, followed by the conclusion.

II. Literature Review

Related works and previous literature's for our study can be discussed in several directions. The taxonomy of our related work are divided into the following categories: survey and text data analysis for smaller datasets, text mining and text data analytics, knowledge-intensive models, and finally, sentiment and topic modeling using clustering, classification, and prediction tasks. However, due to the problem definition studied in this research, we opted not to consider powerful methods in Neural Machine Translation [9], [10], Memory Networks and Neural Networks [11], [12], Memory networks and Metric Learning [13]. Recently, such models have been a subject of concern due to rising criticism of using large amounts of training data, over-fitting models, less robustness, and lack of interpretability and reproducibility [14].

One of the critical aspects of this literature review is to set the appropriate methods to compare with. Few-shot learning [15] and zero-shot learning [16] closely match the dataset issues we tackle in this research. Memory-Augmented Neural Networks (MANN) are superior to LSTM's and perform good regression and classification tasks [17]. Similarly, zero-shot learning methods are effective for learning without training instances or in significantly fewer training samples. Conventionally, meta-learning is another efficient task-specific small instance favorable training method that aims to achieve maximal performance on a new task after the parameters are updated through zero or a couple of gradient steps [18]. Promising research outcomes in generalization, topic modeling, text classification, and machine perception can be observed using such models proposed over the years [19]. However, such models are criticized for failing to capture and extract appropriate features from the training instances and showing dependencies on hyperparameter tuning.

To achieve a good representation of words and characterizing an input vector, classical statistical methods have shown promising results, including Support Vector Machines (SVM), Naive Bayes, Bayesian modeling, and Logistic Regression. Another phase of combined modeling is to predict, segment, and classify to obtain the contextual information of the text. However, combined models are criticized due to the lack of extraction of unseen non-linear features, suffering from capturing long-term dependencies among features, and lower accuracy compared to DL methods [20] [21]. From a different angle, eliminating unnecessary features and utilizing proper instances is one of the primary desiderata of NLP models. Yet, what are the "appropriate features" entail and how they should be evaluated are not well understood, nor are there any common standards to evaluate it in a DL framework.

Recently, topic modeling and generalisation stood to be a powerful tool and applied in wide applications in many fields such as linguistics, psychology, clinical methodology, software engineering, and historical science [19]. However, due to the black-box characteristics of the survey or short textual data models, instance-level feature importance and extracted correlations are the key concepts yet to be addressed in such methodology.

III. Methodology

In this section, we describe our methodological approach that is tailored to address the many challenges present in analyzing the Open Sourcing Mental Illness (OSMI) survey dataset. These challenges are also present in many small datasets. The main prediction task is to use the results of the survey to predict the level of comfort of employees to discuss mental health in the workplace. A secondary goal is to provide insights in the factors that influence an employee's level of comfort. These tasks are particularly challenging due to the relatively small number of observations, as well as the mixed data types (numerical, categorical and text data). Our two-stage modeling approach can be summarized as follows:

- The first stage of modeling extracts insights from the dataset and filters the only contributing instances to use for the second stage of modeling. This systematic approach will help retain essential features and utilize the full potential of the dataset by not losing information and data insights from small training sets. A set of the stacked classifier with a second layer of meta-learners followed by hypothesis testing is concludes this stage of modeling.
- The second stage is performed on the modified, filtered set of features extracted in the first stage. Since the second stage provide actionable insights, we proposed CNN and Bi-RNN architectures that is capable of training with minimum text based training samples.

Next, we describe each stage in more details.

A. First stage

The first stage of modeling involves stacking two levels of classifiers for appropriate selecting features. In general, stacking classifiers is also regarded as an ensemble learning technique where multiple classifiers combine modeling with a meta-classifier. Given a training dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{m} (\mathbf{x}_i \in \mathbb{R}^n, y_i \in \mathcal{Y})$, y_i represents the target value, and x_i represents the feature vectors of the n-th instance which randomly split the data into K-fold such that $S_1, S_2, \ldots, S_K (K = n)$. In this paper, we mainly adopted the CV method proposed by Wolpert [22]. However, we modified the original proposed method into a convergence strategy to select the best classifier among the learners. For convergence, each CV $(C_1, C_2, \ldots, C_I)$, each C_i is trained by $D^{(-K)}$ and predict each instance x_i in $D_K \cdot R_K^{(-i)}(x)$. At the end of entire CV process, the prediction on the model P_{kn} is represented by the output $H(\mathbf{x})$ and denoted by:

$$H_{\text{cv}} = (y_n, P_{1n}, \ldots, P_{In}) \, (n = 1, 2, \ldots, N) \tag{1}$$

Algorithm 1 Stacking classifiers with k-fold CV

Input: Training data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{m} (\mathbf{x}_i \in \mathbb{R}^n, y_i \in \mathcal{Y})$
Output: Stacked ensemble Classifier H

Step-1: initialize cross-val for classifiers $S_1, S_2, \ldots, S_K (K = n)$
Step-2: Randomly split $\mathcal{D}$ into K equal-size subsets: $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K\}$
for $k \leftarrow 1$, to k **do**
    Step 2.1: first-level classifiers
    for $t \leftarrow i$ to T **do**
        Learn first stacked classifiers
    end for
    Second-level classifier
    for $\mathbf{x}_i \in \mathcal{D}_k$ **do**
        Get $\{\mathbf{x}'_i, y_i\}$
    end for
end for
Step-3: check max value of h'
if $max = True$ **then**
    go to Step-4
else
    go to Step 2.1
end if
Step-4: initialize level-two classifiers
for $t \leftarrow 1$, to T **do**
    Re-learn h' on split $\mathcal{D}$
end for
return $H(\mathbf{x}) = h' \left(h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_T(\mathbf{x})\right)$

Based on the primary prediction task and considering the nature of our imbalanced training dataset, first we chose widely adopted logistic regression, random forest, KNN, and GaussianNB as our base pipeline classifier. Secondly, we optimized the gradient boosting as a meta-classifier. Finally, the meta-classifier accuracy and mitigation of the over-fitting is optimized by best parameters converged from the stacked classifier.

1) Hypothesis: A prescriptive hypothesis testing and analysis is applied for actionable recommendations and filtering relevant factors of Mental Health. The setup of hypothesis testing steps is pursued as follows:

- First, a simple logistic regression determines the factors with statistical significance. Statistical significance (p-value, confidence interval) sets the testing parameter, and Pseudo R-squared evaluates the hypothesis testing.
- Secondly, a random forest and XGB-classifier generate the important features together and rank the features by feature importance and odds ratio. Subsequently, the feature relevance may select both un-correlated and irrelevant features for a certain degree of confidence due to erroneous information gained in tree nodes. The odds ratio that includes Fisher's exact probability statistic and the maximum-likelihood ratio mitigates such issue of determining irrelevant features [23].

B. Second stage

For stage-two modeling, we propose modified architectures of CNN and Bi-RNN.

1) CNN: For text classification problems, CNN works as an optimized neural network. The leading optimization lies between the text input matrix and convolution kernels that operate the separation or classification. However, unlike other text classification problems, where input sequence is $[x_0, x_1, \ldots, x_{T-1}]$, where $x_t \in \mathcal{R}^d (t = 0, 1, \ldots, T-1)$, $\left[w^1, w^2, \ldots, w^m\right]$ is the m convolutional filters of length l, this particular architecture uses a single pooling layer over the whole input text and results in a representation of the sequence within a single vector. The general structure of such architecture is presented in Figure 1.

a) Input Layer: A word vector matrix consisting of $N \times d$ is the input layer for the model where N is the no of input, and d is the dimension. Depending on the max features of the word vector matrix, N can be optimized.

b) Hidden Layer: Since one of the primary objective of our proposed model is to make it less complex and computationally efficient, the hidden layer is optimized very carefully. In general, CNN hidden layer includes a convolutional layer followed by a pooling layer. The conv layer is proposed as follows::

$$c_i = f\left(\sum W_1 \cdot X_{i:i+h-1} + b_1\right) \tag{2}$$

Where c_i is the Conv operation result, h is the no of words in Conv kernel, d is the vector dimension with weight matrix W. Finally, no flatten is used as a global pooling layer is utilized to reduce the dimensionality from 3D to 1D.

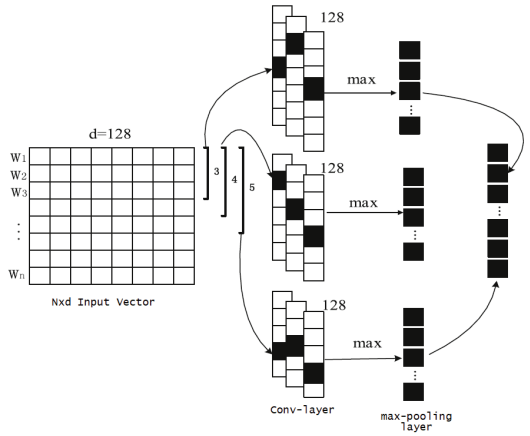

Fig. 1. Proposed CNN Architecture.

c) Word Embedding Layer: Classical NLP problem requires converting One-hot representation to distributed representation. However, the traditional One-hot representation suffers from oversize and losing words [24]. A separate text-level embedding layer on top of the Conv layer is positioned to mitigate such operation. Furthermore, instead of using the weight matrix w (Equation 2), wr_m replaced as the embedding matrix of $m \in [1, M]$-dimensional vectors. As a result, x_m vector representation having w_{re} matrix-vector product is reproduced from each $m - th$ word. Equation 4 updates the previously defined W_1 with wr_m.

$$x_m = W_e wr_m \tag{3}$$

$$c_i = f\left(\sum W_e wr_m \cdot X_{i:i+h-1} + b_1\right) \tag{4}$$

In this proposed CNN, the maximum dimensionality of each word vector is 800 and the lowest is 7.

d) Regularization and Dropout: A classical and widely used regularization method is dropout. Often dropout is placed in large complex CNN models to tackle over-fitting [25]. Unlike the other regularization methods, dropout is a radically different technique that randomly deletes some neurons in the network while training with the same parameters. In our proposed CNN model, we introduced dropout in both word embedding and pooling layers to heavily optimize the training examples. A dropout value of 0.5 is uniformly used in all layers of the CNN.

e) Output: The final dense output layer takes the global pooling layer as input, places dropout and performs classification through the Softmax function. The classification formula is defined as follows:

$$f(x)_{\varnothing} = \frac{1}{1 + \exp\left(-\varnothing^T x\right)} \tag{5}$$

here, exp is the exponential function with base e, $\varnothing$ is the evaluation parameter, and the base value is estimated by the minimum cost function $J(\varnothing)$.

2) Bi-RNN: In general, text classification is a sequential classification problem, and the most commonly used RNN for this particular issue is LSTMs. Given an inputs $X = \{x_1, x_2, \cdots, x_{n_X}\}$, a LSTM having input, memory and output gate, respectively denoted as i_t, f_t and o_t, at time step t, captures both the current h_t and previous sequence h_{t-1}. We can denote a single LSTM network as:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ l_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(W \cdot \begin{bmatrix} h_{t-1} \\ e_t \end{bmatrix} + b \right) \tag{6}$$

where $W \in \mathbb{R}^{4K \times 2K}, b \in \mathbb{R}^{4K \times 1}$. However, a single layer LSTM may or may not retain information from the previous layer and is computationally not efficient. LSTMs as a Bi-RNN is just connecting two independent RNNs. At every time step t, this structure allows the framework to compute backward $h_t^{\leftarrow}$ and forward $h_t^{\rightarrow}$ embeddings.

$$\begin{aligned} h_t^{\rightarrow} &= f\left(W^{\rightarrow} \cdot \left[\overrightarrow{h_{t-1}^{-1}}, e_t\right] + b^{\rightarrow}\right) \\ h_t^{\leftarrow} &= f\left(W^{\leftarrow} \cdot \left[h_{t+1}^{\leftarrow}, e_t\right] + b^{\leftarrow}\right) \end{aligned} \tag{7}$$

At any time t, the output from Bi-RNN is $h_t = [h_t^{\rightarrow}, h_t^{\leftarrow}]$

a) Proposed Bi-RNN: Our proposed Bi-RNN only incorporates embeddings, Conv, and dropout layer to reduce unnecessary complexion. No attention or language-based approach, e.g., BiLSTM, XLNet, Transformers, are utilized for our model. Generally, attention-based LSTMs (Multi-level attention, Deep BiLSTM, Multi-pass BiLSTM) efficiently reduce the impact of non-keywords and improve accuracy. In our proposed approach, we resolve the feature-vectorization issue with our step-one modeling and data pre-processing that builds the text-level word vector representation. Figure 2 represents the general architecture of our proposed Bi-RNN.

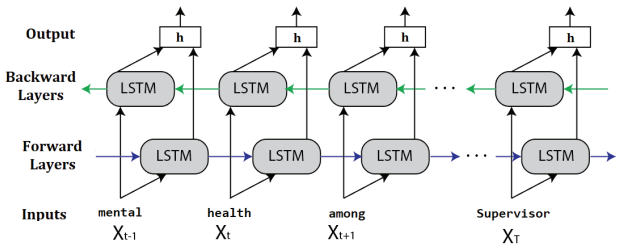

Fig. 2. Proposed Bi-RNN Architecture.

b) Input Layer: Since the stage-one modeling involves transforming the word or sentence into vector tokenization, the first layer of the Bi-RNN directly incorporates the feature sequences L_c as $[Lc_1, Lc_2, \ldots, Lc_{100}]$. Like the CNN, the text-level weighted embedding layer $\mathbf{W_c^T} \in R^{md \times k}$($R$ is a real number and d is the dimension) defines the input operations:

$$Lc_n = f\left(\mathbf{W_c}^{\mathrm{T}} x_{i:i+m-1} + b\right) \in R^k \tag{8}$$

A mixture of spatial dropout (0.35) and 1D dropout (0.50) regularizes the output of the embedding layer (size 30000) before conveying it towards the LSTM layer (size 128).

Algorithm 2 Pseudo-code for Bi-RNN

procedure BI-RNN(0,1)
  Construct word embedding text-embeddings
  Employ Bi-RNN
  if $maxlength > n$ **then**
    preceding contextual features $\overrightarrow{h_f}$ and contextual features $\overleftarrow{h_b}$.
    if $dropout = True$ **then**
      initialize spatialdropout
    else if $dropoutin(lower\_bound, upper\_bound)$ **then**
      initialize conv dropout
    else
      calculate $h_t = \left[h_f^{\rightarrow}, h_b^{\leftarrow}\right]$
    end if
  end if
  while $maxlength \neq 1$ **do**
    optimize loss function
    transform classification
  end while
end procedure

c) Bi-RNN Layer: Bi-RNN obtains the vector by summarizing the feature sequences from both directions (forward $\overrightarrow{h_f}$ and backward $\overleftarrow{h_b}$). The outputs of Bi-RNN are stated as follows:

$$\begin{aligned} \overrightarrow{h_f} &= \overrightarrow{\mathrm{LSTM}}\left(Lc_n\right), n \in [1, maxlength] \\ \overleftarrow{h_b} &= \overleftarrow{\mathrm{LSTM}}\left(Lc_n\right), n \in [maxlength, 1] \end{aligned} \tag{9}$$

d) Output Layer: The final dense is an output layer with a softmax activation function. Softmax helps determine the probability of inclination of a text towards either positivity or negativity. Moreover, binary cross-entropy with an Adam optimizer is incorporated as a loss function along with an accuracy score to evaluate the classification performance.

IV. Experiments

In this section, we briefly describe the experiments on the OSMI dataset to evaluate the performance of the proposed two-stage modeling approach. We perceived and discussed models with comparatively better performance on different evaluation strategies and compare trade-offs among the baseline models.

A. Data Description

OSMI dataset contains cross-sectional survey (2014-2021) information on Mental Health (MH) in Tech individuals working within the technology industry. The dataset is anonymous and openly available from Kaggle at https://osmihelp.org/research, released under a Creative Commons Attribution-ShareAlike 4.0 International license. We only conducted our experiment in minimum training instances choosing 1200 data samples from year 2018 and 2019. The original dataset was unprocessed and unclean and primarily contains survey response. The features are mainly surveyed questions related to Demographic Information (age, sex, race, employment), Geographical Information (Country, State/Prov, territory), and Keywords (mental health, leave, workplace). After combining 2018 and 2019, the processed dataset contained 1170 training samples with 72 features. With 2.33% missing data, 30 of them are numerical features and 41 categorical features along with textual response column.

An extensive data engineering process is regulated in such a way that the processed dataset can be used in both NLP modeling and statistical analysis. Moreover, based on the research statements, the feature columns are modified and grouped together, transformed into a text response to aid modeling. Therefore, the final target columns contains transformed textual response from each participants. Additionally, removing HTML tags, reforming column values and names to avoid duplicate responses, similar grouping questions, handling empty responses, imputing missing values by Decision Tree Classifier, initializing vector tokenizer, and removing punctuation's or stop-words are a few of the data cleaning processes executed to transform the raw dataset.

B. Experimental Setup

In this subsection, we first introduce the machine learning models for the two-stage modeling approach and then outlines the measures used for the model's performance to evaluation. Finally, we describe the implementation details [1].

a) Baseline methods: Based on our literature review and the nature of our research objective, We chose the following baseline models to compare the performance of our proposed CNN and Bi-RNN method: NGrams and MAML MLP, SepCNN, and base LSTM/GRU [26].

b) Evaluation metrics: The choice of appropriate performance metrics depends strongly on the NLP task. Such methods are most effective in prediction and classification tasks when they optimize an appropriate performance measure. Therefore, considering the research objective addressed in this study, accuracy and f1-score are uniformly chosen to compare stage-one and stage-two models' performance. F1-score is widely regarded for text classification and feature extraction tasks; however, to compare different approaches with general models with DL framework, we assessed accuracy and F1-score together as a base evaluating index [27].

c) Implementation Details: We randomly split the data into a training set and test set to validate the robustness against insufficient data, we vary the size of the training set from 60% to 75% and used the remaining part as the test set. Table states the training parameters for all models. Table I, II and III

[1]Our implementation can be found here: https://github.com/UMDimReduction/survey-data-analysis

consists all training parameters and model architecture used in this implementation stage.

TABLE I
TRAINING PARAMETERS

Model	Parameters	Value
CNN	epochs	100
	embedding dim	200
	Batch size	128
	learning rate	0.001 and 0.0023
	dropout	0.5
RNN	epochs	90
	embedding dim	30000
	batch size	128
	learning rate	0.001
	max features	4000
	dropout	0.35 to 0.50

TABLE II
MODEL PARAMETERS: CNN

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 800, 200)	445400
dropout (Dropout)	(None, 800, 200)	0
conv1d (Conv1D)	(None, 796, 128)	128128
global_max_pooling1d (Global)	(None, 128)	0
1MaxPooling1D)		
dense (Dense)	(None, 15)	1935
dense_1 (Dense)	(None, 50)	800
dropout_1 (Dropout)	(None, 50)	0
dense_2 (Dense)	(None, 1)	51

Total params: 576,314
Trainable params: 576,314
Non-trainable params: 0

TABLE III
MODEL PARAMETERS: BI-RNN

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 800, 128)	3840000
spatial_dropout1d (Spatial_Dropout)	(None, 800, 128)	0
Dropout1D	(None, 800, 128)	0
bidirectional (Bidirectional)	(None, 256)	263168
dense_3 (Dense)	(None, 1)	257
Total params: 4,103,425	(None, 1)	51
Trainable params: 4,103,425		
Non-trainable params: 0		

1) Stage-one Modeling Results: Stage-one modeling involves extracting and filtering features through predictive modeling and hypothesis testing. The columns of comfort level of discussing mental health in the workplace (Table IV) and ”comfort level of discussing mental health with supervisor (Table V)” are the dependent variables, grouped into two broad categories (Hesitant/Comfortable) geared for two different analyses. The chosen stacked classifiers predict and determine if the qualitative responses are viable predictors of one's comfort level in discussing MH at the workplace.

The features in Table V and V are ranked by the feature importance from the XGB Classifier model, and the odds ratios are generated from the Logistic Regression model. For better understanding, the odds ratio is converted into a percentage for interpretation. The stacked models illustrated training, test, and validation accuracy of 60.33%, 54.42%, 51.72%, and 62.78%, 57.15%, 53.56%, respectively, for the dependent variable of ”workplace” and ”supervisor.” On the other hand, the f1-score for the models is 52.33% and 55.63%. There are limitations in optimizing the f1-score, especially for the survey dataset, due to the quality of the responses and feature variables being transformed for modeling [27].

TABLE IV
COMFORT LEVEL DISCUSSING MH AT THE WORKPLACE

Question	Odds Ratio	Percentage
Very easy access to medical leave	1.327434	32.743382
Somewhat easy access to medical leave	1.206848	20.684780
Willingness to share MH illness to friends and...	1.313076	31.307603
Company size more than 1000	1.059055	5.905540
Employer formally discussed MH	1.195558	19.555759
Personality Disorder	1.022615	2.261485
Sought treatment from MH professional	0.986758	-1.324217
Neutral difficulty in accessing to medical leave	0.889130	-11.086952
Less likely to reveal MH issue due to observat...	1.703502	70.350174
Europe	0.834415	-16.558473
Age	0.816591	-18.340851
Asia	0.948887	-5.111348

TABLE V
COMFORT LEVEL DISCUSSING MH WITH SUPERVISOR

Question	Odds Ratio	Percentage
Willingness to share MH illness to friends and...	1.238665	23.866512
Employers' emphasis on MH	1.838305	83.830509
Very easy access to medical leave	1.378787	37.878736
Overall MH rating of the industry	1.504067	50.406734
Mood Disorder	1.019414	1.941436
Somewhat easy access to medical leave	1.328322	32.832231
Sought treatment from MH professional	0.820555	-17.944509
Neutral difficulty in accessing to medical leave	1.105449	10.544944
Post-Traumatic Stress Disorder	0.934394	-6.560568
African American	1.040608	4.060828
Adjustment disorder	1.142043	14.204316
Asia	0.948887	-5.111348

Besides the accuracy analysis of the model, the odds ratio and feature importance exhibit some surprising findings. Features related to “medical leave”, “discussing issues”, “workplace ratings”, “employee size”, “medical treatment”, and “age” contributes most to the feature ranking. This refers to a typical pattern or component of personal and professional

attributes related to demographic information affecting MH issues. Based on the odds ratio, feature ranking, and a threshold of 0.512, both positively and negatively correlated 31 features are selected to convey into the second modeling stage.

Finally, for the hypothesis testing, the α is set to 0.05 and the confidence interval as 95%. The proposed null hypothesis can be rejected if the $p-$value is less than 0.05 and the confidence interval does not cross zero.

TABLE VI
HYPOTHESIS TESTING

Model	R-squared	p-value
OLS	0.30	$4.26e^{-62}$
SMOTE	0.362	$3.99e^{-61}$

Table VI implies the pseudo-R-squared of the model is 0.30 along with an LLR p-value $4.26e^{-62}$. Both results indicate that the model is a good fit with the features derived from predictive modeling. We can conclude from stage-one modeling is the survey questionnaires related to mental health or extracted demographic features impact the overall persuasive analysis.

2) Stage-two Modeling Results: Figure 3 and 4 illustrate the training and testing accuracy for CNN and Bi-RNN, respectively. Both graphs show the train and test score crossing ar epoch points 18 to 20. This trend essentially clarifies the small number of training examples with a large feature size. Moreover, features or responses are mostly correlated with other samples. Once the model learns a particular feature vector, it ignores the duplication on the embedding layer. As a result, the training score steadily goes up, and the testing score flutters for newly unseen testing examples. Overall, the f1-score of 0.46 and 0.45 and validation treats confirm the better performance of Bi-RNN compared to CNN. Bidirectional LSTM layer with embedding vectors plays a significant role in feature filtering compared to the CNN method.

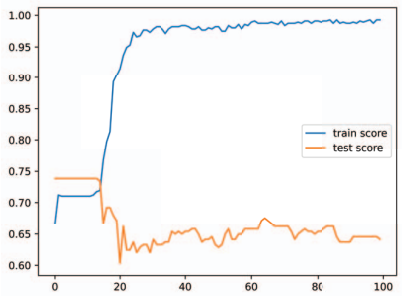

Fig. 3. Train-Test Accuracy: CNN

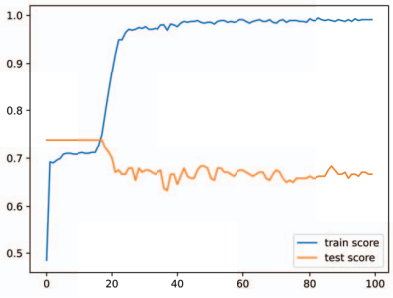

Fig. 4. Train-Test Accuracy: Bi-RNN

3) Sufficiency Analysis: In this subsection, we compare the results of baseline methods with our proposed CNN and Bi-RNN model.

TABLE VII
BASELINE COMPARISON

Model	Train Accuracy	Test Accuracy	f1-score
CNN	98.30	49.12	0.46
Bi-RNN	96.69	70.56	0.45
MLP	99.89	52.33	0.23
SepCNN	98.88	44.91	0.31
LSTM/GRU	99.01	56.66	0.51

It is evident from Table VII that all the other baseline DL models over-fit the data and are not ideal for survey-like dataset features. Our proposed CNN model over-fits after certain training epochs, but the Bi-RNN model illustrates notable performance compared to other baseline models. An ablation study analysis is conducted in the next step to examine the performance more accurately.

4) Ablation Study: The ablation study conducted in this paper is modified from the original concept proposed by Xiong et al. [25]: instead of removing or modifying certain parts of the neural network or introducing additional training parameters, we removed the stage-one modeling phase to run the complete training on unselected feature training instances. As a result, now the input vector for each model is each feature cleaned from the raw dataset after the data cleaning phase. No features are removed from the cleaned dataset, and all are fed into the training set for this experiment.

TABLE VIII
ABLATION STUDY: BASELINE COMPARISON

Model	Train Accuracy	Test Accuracy	f1-score
CNN	99.21	53.12	0.40
Bi-RNN	68.66	59.56	0.31
MLP	99.91	49.33	0.19
SepCNN	99.67	48.91	0.22
LSTM/GRU	99.52	41.01	0.27

Table!VIII illustrates the ablation study results from proposed CNN, Bi-RNN, and baseline methods. The results reveal that all methods except Bi-RNN stills show over-fitting trends over similar training parameters. Since the feature extraction before tokenization is not applied in this experiment, the

input Lc_n effectiveness of the lower embeddings effectiveness of the proposed Bi-RNN fades away due to small training samples. Moreover, for all models, the f1-score shows steady trends, indicating there are not much deviations between input word vectors and features. In general, the different methods of generating word embedding vectors have different effects on classification performance. The overall results indicate the effectiveness and importance of filtering features through stage-one modeling in such small training set.

V. Conclusion

The classification task in a combined NLP and feature extraction task is substantially more complex with a small training dataset. Traditional statistical and Bayesian models show promising results; however, they are erroneous in large feature vectors. Our two-stage modeling approach successfully resolves the performance issues mentioned above. Stage-one modeling effectively filters the feature from a large feature vector and embedding layer with dropout and bidirectional LSTM enhances semantic understanding and improves classification accuracy in the second stage. It is evident from the experimental results that our proposed approach contributes to the state-of-art models where baseline DL models fail to gain noticeable performance. The future work includes:

- Improving the accuracy of the stacked classifiers by introducing a multi-layer stacked ensemble classifier.
- Investigating the effect of embedding layer on the performance of Bi-RNN.
- Applying our current approach to more real world-dataset in different applications.

Acknowledgment

We would like to acknowledge support from the NSERC CREATE grant on Visual and Automated Disease Analytics program. MT also acknowledges funding via a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (NSERC), RGPIN-2021-04073.

References

[1] Ribeiro, A. H., Tiels, K., Aguirre, L. A., & Schön, T. (2020, June). Beyond exploding and vanishing gradients: analysing RNN training using attractors and smoothness. In International Conference on Artificial Intelligence and Statistics (pp. 2370-2380). PMLR. 1

[2] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780. 1

[3] Onan, A., & Toçoğlu, M. A. (2021). A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification. IEEE Access, 9, 7701-7722. 1

[4] Trueman, T. E., & Cambria, E. (2021). A convolutional stacked bidirectional lstm with a multiplicative attention mechanism for aspect category and sentiment detection. Cognitive Computation, 13(6), 1423-1432. 1

[5] Khan, M., Wang, H., Riaz, A., Elfatyany, A., & Karim, S. (2021). Bidirectional LSTM-RNN-based hybrid deep learning frameworks for univariate time series classification. The Journal of Supercomputing, 77(7), 7021-7045. 1

[6] Kaur, D., Islam, S. N., & Mahmud, M. (2021). A VAE-Based Bayesian Bidirectional LSTM for Renewable Energy Forecasting. arXiv preprint arXiv:2103.12969. 1

[7] Mao, K., Zhang, W., Wang, D. B., Li, A., Jiao, R., Zhu, Y., ... & Chen, J. (2022). Prediction of Depression Severity Based on the Prosodic and Semantic Features with Bidirectional LSTM and Time Distributed CNN. IEEE Transactions on Affective Computing. 1

[8] Semberecki, P., & Maciejewski, H. (2017, September). Deep learning methods for subject text classification of articles. In 2017 Federated Conference on Computer Science and Information Systems (FedCSIS) (pp. 357-360). IEEE. 1

[9] Chen, S., Liu, C., Haque, M., Song, Z., & Yang, W. (2022). NMTSloth: Understanding and Testing Efficiency Degradation of Neural Machine Translation Systems. arXiv preprint arXiv:2210.03696. 2

[10] Cherukuri, H., Ferrari, A., & Spoletini, P. (2022, March). Towards Explainable Formal Methods: From LTL to Natural Language with Neural Machine Translation. In International Working Conference on Requirements Engineering: Foundation for Software Quality (pp. 79-86). Springer, Cham. 2

[11] Ruggeri, F., Lagioia, F., Lippi, M., & Torroni, P. (2022). Detecting and explaining unfairness in consumer contracts through memory networks. Artificial Intelligence and Law, 30(1), 59-92. 2

[12] Xing, Y., Qian, X., Guan, Y., Yang, B., & Zhang, Y. (2022). Cross-Project Defect Prediction Based on NLP Methods. Pattern Recognition Letters. 2

[13] Künas, C. A., Serpa, M. S., Padoin, E. L., & Navaux, P. O. (2022). Improving Performance of Long Short-Term Memory Networks for Sentiment Analysis Using Multicore and GPU Architectures. In Latin American High Performance Computing Conference (pp. 34-47). Springer, Cham. 2

[14] Zini, J. E., & Awad, M. (2022). On the Explainability of Natural Language Processing Deep Models. ACM Computing Surveys (CSUR). 2

[15] Gaikwad, M., & Doke, A. (2022, May). Survey on Meta Learning Algorithms for Few Shot Learning. In 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1876-1879). IEEE. 2

[16] Alhoshan, W., Zhao, L., Ferrari, A., & Letsholo, K. J. (2022, March). A Zero-Shot Learning Approach to Classifying Requirements: A Preliminary Study. In International Working Conference on Requirements Engineering: Foundation for Software Quality (pp. 52-59). Springer, Cham. 2

[17] Becattini, F., & Uricchio, T. (2022, October). Memory Networks. In Proceedings of the 30th ACM International Conference on Multimedia (pp. 7380-7382). 2

[18] Lee, H. Y., Li, S. W., & Vu, N. T. (2022). Meta Learning for Natural Language Processing: A Survey. arXiv preprint arXiv:2205.01500. 2

[19] Sandhiya, R., Boopika, A. M., Akshatha, M., Swetha, S. V., & Hariharan, N. M. (2022). A Review of Topic Modeling and Its Application. Handbook of Intelligent Computing and Optimization for Sustainable Development, 305-322. 2

[20] Zeimpekis, D., & Gallopoulos, E. (2006, December). Linear and nonlinear dimensional reduction via class representatives for text classification. In Sixth International Conference on Data Mining (ICDM'06) (pp. 1172-1177). IEEE. 2

[21] Li, Y., Liu, L., & Yao, K. (2021). Neural Sequence Segmentation as Determining the Leftmost Segments. arXiv preprint arXiv:2104.07217. 2

[22] Wolpert, D. H. (1992). Stacked generalization. Neural networks, 5(2), 241-259. 3

[23] McHugh, M. L. (2009). The odds ratio: calculation, usage, and interpretation. Biochemia medica, 19(2), 120-126. 3

[24] Wang, Y., Zhou, Z., Jin, S., Liu, D., & Lu, M. (2017, October). Comparisons and selections of features and classifiers for short text classification. In Iop conference series: Materials science and engineering (Vol. 261, No. 1, p. 012018). IOP Publishing. 4

[25] Xiong, J., Zhang, K., & Zhang, H. (2019, June). A vibrating mechanism to prevent neural networks from overfitting. In 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC) (pp. 1737-1742). IEEE. 4, 7

[26] Pronko, R. (2019). Simple bidirectional LSTM solution for text classification. In Proceedings of the Pol Eval 2019 Workshop (p. 111). 5

[27] Zhang, D., Wang, J., & Zhao, X. (2015, September). Estimating the uncertainty of average F1 scores. In Proceedings of the 2015 International conference on the theory of information retrieval (pp. 317-320). 5, 6