

Received August 10, 2021, accepted September 27, 2021, date of publication October 21, 2021, date of current version November 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3121791

A Hybrid Deep Learning Technique for Personality Trait Classification From Text

HUSSAIN AHMAD¹, MUHAMMAD USAMA ASGHAR^{ID1}, MUHAMMAD ZUBAIR ASGHAR^{ID1}, AURANGZEB KHAN², AND AMIR H. MOSAVI^{ID3,4,5}

¹Institute of Computing and Information Technology, Gomal University, Dera Ismail Khan 29220, Pakistan

²Department of Computer Science, University of Science & Technology, Bannu 28100, Pakistan

³Faculty of Civil Engineering, Technische Universität Dresden, 01069 Dresden, Germany

⁴John von Neumann Faculty of Informatics, Óbuda University, 1034 Budapest, Hungary

⁵Department of Informatics, J. Selye University, 94501 Komárno, Slovakia

Corresponding authors: Muhammad Zubair Asghar (zubair@gu.edu.pk) and Amir H. Mosavi (amir.mosavi@mailbox.tu-dresden.de)

ABSTRACT Recently, Cognitive-based Sentiment Analysis with emphasis on automatic detection of user behaviour, such as personality traits, based on online social media text has gained a lot of attention. However, most of the existing works are based on conventional techniques, which are not sufficient to get promising results. In this research work, we propose a hybrid Deep Learning-based model, namely Convolutional Neural Network concatenated with Long Short-Term Memory, to show the effectiveness of the proposed model for 8 important personality traits (Introversion-Extroversion, Intuition-Sensing, Thinking-Feeling, Judging-Perceiving). We implemented our experimental evaluations on the benchmark dataset to accomplish the personality trait classification task. Evaluations of the datasets have shown better results, which demonstrates that the proposed model can effectively classify the user's personality traits as compared to the state-of-the-art techniques. Finally, we evaluate the effectiveness of our approach through statistical analysis. With the knowledge obtained from this research, organizations are capable of making their decisions regarding the recruitment of personals in an efficient way. Moreover, they can implement the information obtained from this research as best practices for the selection, management, and optimization of their policies, services, and products.

INDEX TERMS Personality trait, deep learning, artificial intelligence, convolutional neural network, long short-term memory, social networks, machine learning.

I. INTRODUCTION

Cognitive Science is a multidisciplinary area of research that aims at addressing different cognitive processes and mental states, including learning, thinking, perception, remembering, and emotions. Among the aforementioned types of cognition, personality plays a pivotal role in identifying the social behaviour of humans. Computer-based personality detection and classification have remained an active area of research for a long time. Personality detection can be performed using multiple media, such as text, images, video, and audio [1]. Cognitive-based sentiment classification from social media-text is an area that originates from several challenges for researchers in cognitive computation. In this area, a lot of work has been done and much more can be investigated. Textual cognitive-based sentiment analysis (SA) is not merely a theoretical area, rather it has several applied

fields, such as health [2], [3], education [4], finance [5], and others [6]. Being a merger of cognitive science and human neurology, it can address the gap between the abstractions of cognitive science and the more emerging area of personality detection from a person's textual feedback expressed on social media [7]. Social media platforms, such as Twitter, Facebook, and Instagram, have experienced an unexpected worldwide spread in recent years. For example, by the 3rd quarter of 2019, Twitter had over 330 million active users per month [8]. Progress in natural language processing and text analytics gives researchers an opportunity to use big-data sources for extracting and analysing the textual personality traits expressed by users while using social media, as long as the data scientists working on social media content are able to address the challenging issues specific to such content.

A. RESEARCH STUDY MOTIVATION

Several studies have been conducted by computational linguistic experts to detect and classify personality-traits

The associate editor coordinating the review of this manuscript and approving it for publication was Hiram Ponce^{ID}.

at several levels, such as words, phrases, sentences, and reviews [9], [10]. However, in many studies, personality detection and classification is a challenging area in cognitive computation. The prior work [9] on user personality detection used a single Convolutional Neural Network (CNN) deep learning model. The work by [9] exploited the CNN model that only extracts the local features of sentences without retaining the previous context information. Therefore, further work is required to address the aforementioned issues for efficient detection and classification of a user's personality traits.

We propose a hybrid deep learning technique for user personality detection. The main emphasis is to efficiently classify the user's personality traits by adding a Long Short Term Memory (LSTM) layer for persevering with contextual information in an efficient manner. The proposed system is inspired by the prior work [9] performed on the user's personality detection and classification. Previous studies have used a single deep learning model for personality trait classification. However, we introduced a hybrid user's behaviour detection and classification approach using an LSTM layer. The proposed technique aims to classify the user's personality traits efficiently with a fusion of deep learning models.

B. PROBLEM BACKGROUND

In the last few years, cognitive-based SA applications have become popular among online communities for knowing about the opinions and personality traits of individuals pertaining to different issues, policies, and others. However, due to the diverse nature of social media content, it is tedious to analyse text using existing techniques to detect personality traits from such content. Therefore, extraction and analysis of social media content has become essential through automatic classification of personality traits. A lot of work has been carried out in the fields of text-based SA [11]–[13], lexicon construction [14], [15], cognition [16], aspect-based SA [17], and visual SA [18]. However, more work needs to be done in the context of cognitive-based social media, with an emphasis on extracting and classifying personality traits from social media content. The aforementioned issues often result in incorrect classification of cognitive-based sentiment classification of social media content. Therefore, it is necessary to develop a method to classify personality traits in social media by automatic classification of such content.

C. RESEARCH PROBLEM

Existing studies on personality trait classification are based on models with a limited number of personality clues. Furthermore, such techniques need an advanced feature set for efficient classification of users' behaviour on social media content. The problem of personality trait classification tackled in this research work is formulated as follows: Consider a set of P labelled text reviews: $P = \{p_1, p_2, p_3, \dots, p_n\}$, for ($i = 1$ to P) with labels P_i . The label p_i specifies that i th text review is an I(Introversion)-E(Extroversion), N(Intuition)-S(Sensing),

T(Thinking)-F(Feeling), and J(Judging)-P(Perception) personality trait. We consider personality trait detection as a classification problem in which the focus is on classifying input text into I(Introversion)-E(Extroversion), N(Intuition)-S(Sensing), T(Thinking)-F(Feeling), and J(Judging)-P(Perception) personality traits. The research problem is to build a reliable and powerful personality trait detection model that can detect and classify a text as I(Introversion)-E(Extroversion), N(Intuition)-S(Sensing), T(Thinking)-F(Feeling), and J(Judging)-P(Perception) personality traits. For instance, the input text from the dataset related to the personality trait classification problem is given as: "*I'm finding the lack of me in these posts very alarming.*" Our aim is to identify the correct personality traits I-E, N-S, T-F, and J-P of the user from the given text review. In this research work, our objective is to build a robust hybrid deep learning-based personality detection model that can efficiently classify the given text reviews into the desired personality class.

D. RESEARCH QUESTIONS

RQ.1 How to apply a hybrid Deep Learning technique, namely CNN+LSTM, for classifying personality traits from the input text?

RQ.2 What is the efficiency of the proposed technique in relation to other machine learning and deep learning techniques?

RQ.3 How to estimate the efficiency of the proposed technique regarding personality trait classification with respect to baseline studies?

E. CONTRIBUTION

The major contribution to the research work are as follows:

1. To capture the previous context information from texts, we exploit the LSTM deep learning model to extract word semantics at a deeper level. The LSTM model performs effective learning of long-range dependencies among sequences of words for large texts.
2. By combining CNN with LSTM, we propose a hybrid deep neural network, namely CNN+LSTM, to extract local features and learn long-range dependencies that assist in capturing context information in an effective way.
3. We conducted experiments with different machine learning classifiers like Support Vector Machine (SVM), K Nearest Neighbors (KNN), Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), and XGBoost, and also a variant of deep learning models like CNN, LSTM, Recurrent Neural Network (RNN), and Gated Recurrent Network (GRU).
4. Our proposed technique produced efficient results over state-of-the-art techniques.
5. In this work, the intention is to contribute to the scientific literature by developing an enhanced deep learning technique for efficient classification of social

- media-based textual content into different personality-related traits.
6. The proposed system can assist organizations in detecting and analyzing the attitudes of their customers about a service, policy, or product, which can assist in better decision making.

The rest of the paper is structured as follows. (i) Section 2 demonstrates the related work on personality trait classification by discussing existing techniques and systems. (ii) Section 3 introduces the proposed methodology for personality trait classification. (iii) Experimental setup is presented in section 4 and the final section concludes the proposed system with its limitations and possible future guidelines.

II. REVIEW OF LITERATURE

This section reports a detailed review of selected literature pertaining to personality trait classification. In the literature, numerous methods are suggested for exploiting different approaches to resolve the prediction issue, and techniques are implemented to present personality prediction. The different studies relevant to machine learning approaches to personality recognition are recorded as follows:

A. SUPERVISED MACHINE LEARNING FOR PERSONALITY RECOGNITION

The Supervised machine learning approach, also called the Corpus based approach (CBA), is one of the most widely used approaches among researchers to explore written and spoken text. It has several advantages, for example, exploring word usage, frequency, collocation, and concordance [19]. However, the Supervised or Corpus-Based approach (CBA) requires an annotated corpus for classifier training and testing [20] which is one of the major drawbacks of such techniques. Ref. [19] addressed the problem of database-independent emotion recognition by applying a neuro-fuzzy-based inference system. The proposed system can effectively learn and generalize the training samples. The experimental results show the performance improvement of the proposed system with respect to SVM.

B. UNSUPERVISED MACHINE LEARNING FOR PERSONALITY RECOGNITION

This approach is exploited when the annotated data credibility is complicated. The classification of input text is performed based on keyword lists regarding the individual classes. The unsupervised technique is easy to use in the case of investigating domain-dependent data [21]. The following sections present an overview of the selected studies related to machine learning approaches. Ref. [22] proposed a personality recognition system using an unsupervised approach and considered five personality traits (Big Five) to extract and classify personality traits of the user from different social media sites like Friend Feed. They exploited different linguistic features related to personality and built a personality model. The system computes personality scores for a given text

and obtains satisfactory results. However, their system didn't address features related to inter-user interaction. Ref. [23] proposed a Twitter Personality based Influential Community Extraction (T-PICE) System that generates the most influential communities in a Twitter network graph. They identified users' personality traits by extending the existing approaches by aggregating data that shows more aspects of user behaviour based on machine learning techniques. For this purpose, an existing modularity-based community detection algorithm is used by including a pre-processing step that removes graph edges based on users' personalities. However, scalability problems need to be addressed while considering a large graph.

C. SEMI-SUPERVISED AND HYBRID APPROACHES FOR PERSONALITY RECOGNITION

In semi-supervised and hybrid techniques, the features regarding supervised approaches and lexicon-based approaches with labeled data records are included. The related literature about hybrid techniques is as follows: [24] developed a personality comparison system between humans and chimpanzees using static images with an emphasis on neutral expressions between them. The results obtained show that humans can perceive more accurate characteristics as compared to chimpanzees. Further experiments are required to investigate more features between humans and chimpanzees.

D. DEEP LEARNING APPROACHES FOR PERSONALITY RECOGNITION

Deep learning is comprised of a group of algorithms that mimic the functionality and structure of the human brain. In simple terms, it contains a set of neurons to receive input and also a set of neurons to transmit output signals. The deep learning-based models can assist with different tasks like speech recognition, computer vision, natural language processing, and handwriting generation [25]. A model has been proposed for the human Big Five personality traits prediction, which needs 8 times less data [26]. An embedding layer used for word extraction underlying user tweets is named the GloVe model. The model's training and testing are performed using the provided Twitter data. Moving forward, the testing of data is performed on three fusions (i) LIWC along with GP, (ii) 3-Gram along with GP, and (iii) GloVe along with RR. The proposed model outperformed the state-of-the-artwork with a mean correlation of 0.33 across the Big Five traits. The present work exploited only English Twitter content that needs to be extended to further languages. Furthermore, the proposed model's efficiency can be estimated by an extended number of tweets. A personality identification approach is applied to text data by exploiting a deep neural network, proposed by [10]. The present work used a hierarchical scheme called AttRCNN that is capable of retaining semantic features at a deeper level. The results reveal that the proposed features effectively perform better than the compared features.

E. RESEARCH GAP IDENTIFIED FROM THE EXISTING STUDIES

The aforementioned studies on predicting personality traits have applied different ML, DL, and other techniques. However, ML approaches have used a classical machine bag of words approach as a feature representation technique. This limitation has successfully been overcome by the state-of-the-art word embedding used in DL models. However, the issue of context information handling empowered by rich feature extraction using deep neural networks needs to be addressed using hybrid DL models supported by feature selection for a more efficient classification of personality traits.

III. OVERVIEW OF THE PROPOSED METHODOLOGY

A. MOTIVATION

Different deep learning models have been applied for personality classification, including CNN, RNN, GRU, and LSTM. All these individual deep learning models are incapable of capturing semantic information in a better way. However, applying the fusion of deep learning models, such as CNN+LSTM, allows taking advantage of the two models, namely CNN and LSTM, to assist in capturing context information in a better way. Furthermore, the exploitation of the LSTM model facilitates understanding the context in an effective way by saving information in one direction. The current work is aimed at performing the task of *personality trait classification* by extracting the user's personality traits "I-E," "N-S," "T-F," and "J-P" from textual data. For example, the input text: "*I am finding the lack of me in these posts very alarming,*" is tagged with one of the desired personality traits. For this purpose, we proposed a deep neural network model, namely the Convolutional Neural Network including Long Short-Term Memory (CNN+LSTM). An overview of the proposed technique is presented as follows:

The proposed method for personality trait classification from social media text is comprised of different modules: (i) acquiring data; (ii) pre-processing of data; and (iii) implementing a deep neural network (see Fig. 1). The acquisition of data is the initial step in the proposed method, after which certain pre-processing steps are applied to the social media textual reviews. The pre-processed reviews are input into the third module, where the deep neural network is applied to transform the reviews into a machine-readable form, i.e., a real-valued vector [27]. The numeric representation of words, also called word embedding, is then made input to the hidden layer. The models used in the hidden layers are CNN and LSTM. The CNN model acts as a feature extractor that extracts the salient features from the input data [28], and the LSTM model learns long-term information [29] to efficiently classify the user reviews. Lastly, input review text is classified into different personality classes, namely "I-E," "N-S," "T-F," and "J-P." A detailed description of the three modules is presented in the subsequent sections.

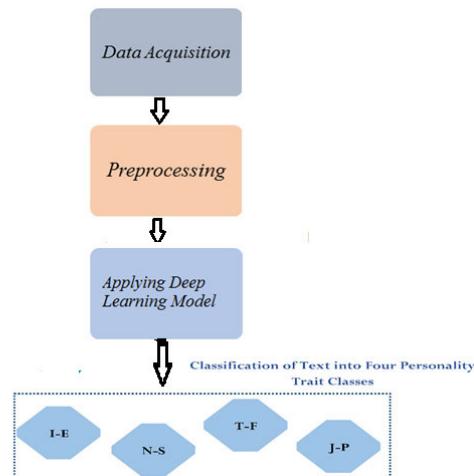


FIGURE 1. Flowchart of the proposed method.

B. ACQUIRING DATA

We used the *MBTI* dataset for conducting experiments on personality trait classification [30] that consists of 8675 rows. Each review in the data is tagged with a unique class with respect to the four sets of personality trait classes: "I-E," "N-S," "T-F," and "J-P" (see Table 1). In this work, experiments are implemented using the Python programming language, based on the Anaconda Jupyter notebook [31]. The partitioning of the original dataset is performed with ternary sets that are a set of validation data, a set of test data, and a set of train data. (see Fig. 2).

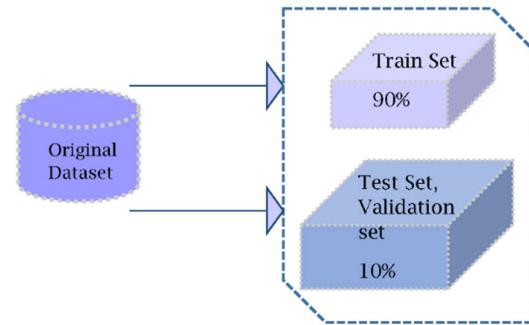


FIGURE 2. Original dataset partitioning.

TABLE 1. Detail of MBTI dataset.

Dataset	URL	Description
MBTI	https://www.kaggle.com/datasnaek/mbti-type	The dataset is created by Mitchell J and comprised of 8676 instances of tweets along with personality types. The size of the dataset is 25 MB.

1) SET OF TRAIN DATA

The set of train data is exploited for model training by providing the input and the associated outcomes [12]. The

preparation of the proposed model is performed by utilizing 90% of the information as a training set, which can change for various experiments. To fit the model, training data is used for learning the model [15]. A train set example is given in Table 2. The trainset is saved in a CSV file.

TABLE 2. Train set examples related to personality trait classification.

Review Text ID	Review Text	Label
1.	I fully believe in the power of being a protector, to give a voice to the voiceless. So in that spirit I present this film, and hope it it received in the spirit of compassion.	T(Thinking)-F(Feeling)
2.	Sometimes I wonder that too.. the reason being that INFJ's seem so abstractly deep that even being one myself I don't understand them all the time	T(Thinking)-F(Feeling)
3.	he ISFPs I know are charming, reflective people. They have a soul..	J(Judging)-P(Perception)
4.	I got 593. From what I've read about the Enneagram I'm a 953, though. I read somewhere that a lot of 9's mistype themselves as 5's.	N(Intuition)-S(Sensing)
6.	Sorry I still can't determine the order, right now I have you as xNFP leaning INFP. Maybe you can tell us about your childhood. What were you like as a child? Were you always looking for new...	I(Introversion)-E(Extroversion)

2) SET OF VALIDATION DATA

The model usually shows high accuracy in the training phase, but during the testing phase, the model's efficacy declines. Hence, to overcome the model performance error with respect to under-fitting and overfitting, it is needed to exploit the validation set [12]. We used a 10% validation set. Keras supports two approaches regarding parameter tuning to verify model best functioning [32], namely (i) manual data validation, and (ii) automatic data validation. In the present work, we have implemented manual data validation.

3) SET OF TEST DATA

The set of test data is exploited to estimate the model's efficacy based upon new/unseen examples. It is applied after the model gets fully trained through the training as well as the validation set. The test set performs the model's final estimation [12]. We used 10% of test data, which is independent of the train set. The train set is used once when the model is completely trained. The model is finally evaluated using test data. A test set example is provided in Table 3. Fig. 2 depicts that the dataset is split into 90:10 proportions using the train-test split method of scikit-learn, in which 10%

TABLE 3. Test set examples related to personality trait classification.

Review Text ID	Review Text	Label
5.	I was not the most clear with my example, you're right. Hmm... how about: I appreciate my trainer, but I wouldn't shed a tear if he got hit by a bus. vs I love my trainer, but I...	J(Judging)-P(Perception)
7.	Your post really makes a lot of sense to me because I also feel like I need some sort of real meaning that I can touch and feel within me. I want my actions in this life to matter in the long run and...	T(Thinking)-F(Feeling)

is the validation set. We used a validation set to tune the hyperparameters and to configure and evaluate the model.

Table 4 provides detail of the dataset used in this study.

TABLE 4. Dataset description.

Task	Dataset with title	Number of Review Text	Labels
<i>Personality trait classification</i>	DS: MBTI	8675	I(Introversion)-E(Extroversion) (6676-1999), J(Judging)-P(Perception) (5241-3434), N(Intuition)-S(Sensing) (7478-1197), T(Thinking)-F(Feeling) (4694-3981)

After the acquisition of data, it is transferred to the pre-processing module.

C. PRE-PROCESSING OF DATA

The second module of the proposed method is used to apply some basic pre-processing steps to the acquired dataset. described as follows: (i) *Applying Lower Casing*: In this approach, the entire reviews are modified towards the lower case by applying a python-based code that helps to remove duplicated words within the data file. (ii) *Eliminating Stop Words*: The data file contains different stop words like 'the,' 'is,' and 'and,' which are eliminated at this stage [33]. and (iii) *Tokenization*: During tokenization, a review is divided into a sequence of words, called tokens, by applying the Keras tokenizer. The tokenizer enables one to shift the tokens into an integer value [34].

D. IMPLEMENTING DEEP NEURAL NETWORK

In the final step, a deep neural network for *personality trait classification* is implemented on the input text. The CNN+LSTM model consists of distinct layers, namely: (i) Input Layer, (ii) Hidden Layer, and (iii) Output Layer, described as follows:

1) INPUT LAYER

The deep neural network input layer contains input data [35]. We used the embedding layer [36], with the Keras library layer to perform encoding of words into real-valued vectors, which is a numerical representation of the words. It captures semantic information [37].

2) HIDDEN LAYER

The hidden layer is comprised of different layers of CNN and LSTM. The layers/components of the CNN model are described as under:

a: CONVOLUTIONAL LAYER

It is a basic layer and acts as a feature extractor layer because it extracts features from the input data. The feature extraction is performed using a linear operation, namely convolution, in which the filter is moved along the input data to obtain the feature map. After convolution, the feature map is then moved across a nonlinear activation function: ‘Relu,’ to remove the negative values [33].

b: POOLING LAYER

In this layer, a downsampling operation, namely Maxpooling, is applied to the input received from the prior layer, with the aim of reducing the volume of the feature map after convolution [38]. To learn long-term information, we introduced the LSTM layer, described as follows:

c: LSTM LAYER

The LSTM layer takes input from the CNN model and retains both the current and prior information. It provides the facility of memorizing long-term memory by keeping the information for an extensive time period besides information decay [39]. Finally, the LSTM layer outcome is inserted into the output layer.

3) OUTPUT LAYER

After feature extraction through a convolutional layer, down-sampling through the pooling layer, and learning long-term information through the LSTM layer, the classification of the learned features is performed at the output layer [37]. In this layer, the function “softmax” is applied, which assists in classifying the given input text “*I am finding the lack of me in these posts very alarming*” into four personality trait classes, namely: “I-E,” “N-S,” “T-F,” and “J-P.” The softmax function generates probabilities related to the target label (class), and the label (class) with the maximum likelihood is assigned to the input review text.

E. METHOD ARCHITECTURE FOR PERSONALITY TRAIT CLASSIFICATION

The proposed method uses four main modules, namely (i) Representation of Words using the Embedding Layer, (ii) Feature Extraction using the CNN model, (iii) Learning long-term information using the LSTM model, and

(iv) Classification using the Softmax layer. The first module is used to obtain the numeric representation of words, which is then made input to the second module to perform feature extraction. After that, the long, short-term memory takes the output from the previous module to preserve the long-term information, and finally, the last module applies the classification (see Fig. 3). A detailed description of each module is presented as follows:

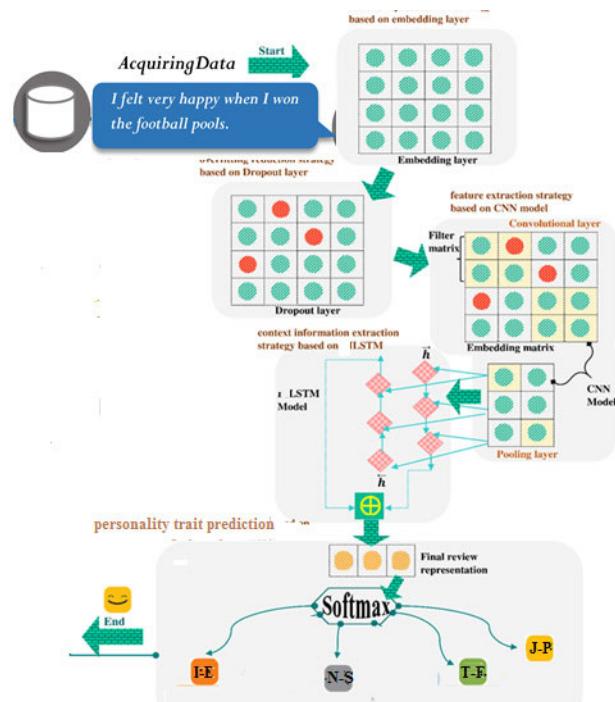


FIGURE 3. Detailed Structure of personality trait classification using CNN+LSTM model.

1) REPRESENTATION OF WORDS USING EMBEDDING LAYER

In the first module, the words are transformed into numerical form, i.e. real-valued vectors [40]. In this layer, the individual review: “*I am finding the lack of me in these posts very alarming*” is treated as a word sequence, while the individual word is expressed as a fixed size vector [41]. The values of vectors are randomly initialized, i.e. a fine-tuned word embedding scheme is used in the embedding layer [40]. Suppose a word w_j is randomly initialized with a v dimensional vector. If the maximum number of words in a review is x then a review is defined as:

$$R = w_1 : x = w_1 \otimes w_2 \otimes w_3 \dots \otimes w_j \otimes \dots \otimes w_x \quad (1)$$

In Eq. (1), the concatenation operator \otimes generates a single review matrix. The review matrix $I \in R^{(x \times v)}$ is created for each review I , while individual row j shows a word embedding w_j related to the j th word in the review [41]. The word vectors, also named as word embedding, are passed to the next module.

Mitigating Overfitting Issue Using Dropout Rate: When a neural network is trained on a limited training set too much, it causes an overfitting issue, which affects the network performance on the test set negatively. In order to mitigate this issue, we apply the dropout layer “*rate*” parameter. The task of the dropout layer is to deactivate a fraction of neurons (units) by assigning neurons a zero value. The dropout rate falls in the range of [0, 1] [41].

2) FEATURE EXTRACTION USING CNN MODEL

In this module, we exploit the CNN model that consists of three different submodules: (i) Filtering using a convolutional layer, (ii) Adding a Bias and Presenting Nonlinearity using an activation function, and (iii) Downsampling using a pooling layer. The detailed illustration of three submodules of the CNN model is shown in Fig. 3. Each of the submodules is described as follows:

a: FILTERING USING CONVOLUTIONAL LAYER

This submodule aims at extracting local (best) features through the review matrix [28]. Mathematically, the feature map generated during the filtering process is defined as:

$$L[p, q] = a(F[p, q] + b) \quad (2)$$

Eq. (2), shows that ‘*a*’ is an activation function, ‘*b*’ is a bias term, while ‘*F* [p, q]’ represents the feature map elements, that are produced using the filtering operation. During this operation, an overlay, also named as the kernel, is applied to the input review text. This operation is completed in different steps: (i) the overlay is set at the top left side region of the input review matrix, (ii) the multiplication operation is performed between the overlay and the underlay, that is, the input review matrix portion is enclosed by the overlay, (iii) sums all the elements obtained after the multiplication operation. Finally, a single element of the feature map is generated, (iv) The overlay then shifts to the right, and the prior steps (i), (ii), and (iii) are performed again, giving another element to the feature map. In this way, the first row is completed, then we shift to the downside, and replicate the same process, which includes all the prior steps, going on all the way to the last overlay in the bottom right-side region f4. Furthermore, the process of filtering is defined mathematically as below:

$$\begin{aligned} F[p, q] &= i \odot k[p, q] = \sum_m \sum_n k[m, n] \\ &\odot i[p + m - 1, q + n - 1] \end{aligned} \quad (3)$$

In Eq. (3), opening up $F[p, q]$ $F[p, q]$ depicts that ‘*i*’ is an individual element of the input review matrix, whereas ‘*k*’ is also an individual element of the kernel matrix $K \in R^m \times n$. Finally, ‘ \odot ’ represents the multiplication operator, where ‘ Σ ’ states the summation operator.

b: ADDING A BIAS AND PRESENTING NONLINEARITY USING ACTIVATION FUNCTION

Firstly, a bias term ‘*b*’ is added to the feature map elements, and then an activation function is applied. The main objective

of the activation function is to declare the nonlinear decision boundary [43]. Here, the activation function, namely “*ReLU (Rectified Linear Unit)*” is exploited, which is an element-wise operation, and it is numerically defined as $\text{Relu}(z) = \max(0, z)$. The formula states that the output will be 0 if the input is 0, otherwise *z* will be output. Finally, after introducing the bias term and activation function in Eq. (2), we obtained $L[p, q]$ that shows the elements of the feature map where $L \in R^c \times d$.

c: DOWNSAMPLING USING POOLING LAYER

The last submodule is the pooling layer used for downsampling the output, taken from the prior submodule. In this layer, only the utmost salient features/information (maximum value in the feature map) are selected by erasing the low-activation features/information from the feature map. Mathematically, it is written as $W = \max(L)$, which states that when the maximum pooling operation is applied to the feature map *L*, then it cuts the feature map into different window sizes and selects the maximum value from each window. The final feature vector is then passed on to the next module.

3) LEARNING LONG TERM INFORMATION USING LSTM MODEL

The third module applies the LSTM model to the input data taken from the CNN model. The main task of LSTM is that it operates on the current data with the prior information. In other words, it memorizes and moves the prior data (information) in a string like neural network structure. The long-time storage of information without decay is due to long-term memory existence, also known as cell states within the LSTM units. The LSTM structure is comprised of three gates, and in order to put the weights between the neural network and the memory cells, these gates are exploited. Furthermore, the memory cell has a connection with itself. If the first forget gate, also known as the ‘*remember vector*,’ contains value 1 with a self-connection, also contains weight value 1, then the information is remembered by the memory cell. Otherwise, if the forget gate contains value 0, then the information is forgotten by the memory cell. The equation for the forget gate is given as:

FORGET GATE

$$\text{forget}_t = \text{sigmoid}(\text{weight}_r [\text{wgm}_{t-1}, \text{x}_t] + \text{bias}_r) \quad (4)$$

If the second input gate, also known as ‘*save vector*’ contain value 1, then it allows the network to write the values into the memory cells. The equation for the input gate is given as:

INPUT GATE

$$\text{save}_t = \text{sigmoid}(\text{weight}_s [\text{wgm}_{t-1}, \text{x}_t] + \text{bias}_s) \quad (5)$$

If the third output gate is known as ‘*focus vector*,’ contains value 1, then it allows the network to read the values from the

memory cell. The equation for the output gate is given as:

OUTPUT GATE

$$\text{focus}_t = \text{sigmoid}(\text{weight}_f [\text{wgm}_{t-1}, \mathbf{x}_t] + \text{bias}_f) \quad (6)$$

In Eq. (4), (5), (6), remember_t , save_t , and focus_t represents the forget, input, and output gates, weight_r , weight_s , weight_f represents weight matrices, while \mathbf{x}^t depicts the current input; wgm_{t-1} shows the previous hidden state, also named as ‘working memory’; and bias_r , bias_s , bias_f represents the bias values for the forget, input, and output gates. Lastly, sigmoid is an activation function.

After that, a new candidate value is added to the cell state (*long term memory*) [44]. In order to compute the candidate values, the following equation is used [45].

Candidate Value

$$\text{lgtm}'_t = \tanh(\text{weight}_l [\text{wgm}_{t-1}, \mathbf{x}_t] + \text{bias}_l) \quad (7)$$

In Eq. (7), lgtm'_t represents the candidate value, and \tanh depicts a tangent hyperbolic activation function.

Each LSTM unit generates two outputs. The first is the new/updated cell state, as well as the other one, is the new hidden state. Now, equations for the two outputs are given as follows:

New Cell State

$$\text{lgtm}_t = \text{remember}_t * \text{lgtm}_{t-1} + (\text{save}_t * \text{lgtm}'_t) \quad (8)$$

New hidden State

$$\text{wgm}_t = \text{focus}_t * \tanh(\text{lgtm}_t) \quad (9)$$

In Eq. (8), lgtm'_t represents the new/update cell state, and lgtm_{t-1} represents the previous cell state, where in Eq. (9), wgm_t represents the new hidden state.

The existence of the three gates secures the LSTM units from the problem of vanishing and exploding gradient. The technique by which LSTM operates the data is sequential. Hence, to effectively perform emotion classification, we applied the LSTM model.

4) FEATURE CLASSIFICATION USING SOFTMAX LAYER

This layer’s objective is to classify input features (final depiction) it receives from the prior module [46]. Initially, we calculate the net input using Eq. 10.

$$A_f = \sum_n^m f_n x_n + b \quad (10)$$

where x represents the input vector, f represents weight vector, also b shows bias term. We selected the softmax activation function for the feature classification. The goal of the softmax function is to give decimal probabilities to individual classes, and the addition of these decimal probabilities should be equal to 1. Each class output is normalized between 0 and 1 by the softmax function using Eq. (11) [47]. Finally, the target class comes with the highest probability [48]. Mathematically, the softmax activation function is defined

as (see Eq. 11):

$$\text{softmax}(A_f) = \frac{e(A_f)}{\sum_{g=0}^n e(A_g)} \quad (11)$$

Applying an Example Case: We take an example of a user review, which is passed through each step of the proposed CNN+LSTM model by classifying into different personality traits: I-E, N-S, T-F, and J-P.

5) EMBEDDING LAYER

Consider a sample review: “I am finding the lack of me in these posts very alarming,” which is passed through the proposed deep learning model to classify it into different personality traits: I-E, N-S, T-F, and J-P. Initially, the tokenization of input review is performed that represents individual words (tokens) for the given review, such as “I,” “am,” “finding,” “lack,” “of,” “me,” “in,” “these,” “posts,” “very,” “alarming.” After generating the tokens, we create an index for particular tokens, known as the vocabulary index. Since the number of individual tokens is 12, the size of the vocabulary is 12. “I:1,” “am:2,” “finding:3,” “the:4,” “lack:5,” “of:6,” “in:8,” “these:9,” “posts:10,” “very:11,” “alarming:12.” Next, the input review is turned into a sequence of integers shown as follows: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. We used an embedding layer to generate the embedding vector for the individual words. For example, the embedding vector for the token “I” is represented as: [0.1 0.3 0.6] and [0.5 0.7 0.3] represents the embedding vector for the token “am.” A similar process is repeated for other words (tokens). As a result, an embedding/review matrix is produced, which is represented as [[0.1 0.3 0.6], [0.5 0.7 0.3], [0.1 0.9 0.1], [0.7 0.8 0.8], [0.4 0.3 0.2], [0.9 0.2 0.6], [0.3 0.7 0.1], [0.9 0.3 0.6], [0.6 0.6 0.3], [0.2 0.3 0.1], [0.3 0.9 0.6], [0.2 0.8 0.9]].

6) DROPOUT LAYER

The dropout layer was introduced in the neural network to mitigate the overfitting issue by using the “rate” parameter with a value of 0.8. The dropout layer randomly deactivates the neurons from the input it received from the embedding layer. Now the output of dropout layer is presented as: [[0.1 0 0.6], [0.5 0.7 0.3], [0 0.9 0.1], [0.7 0.8 0.8], [0 0.3 0.2], [0 0.2 0.6], [0.3 0 0.1], [0.9 0.3 0.6], [0.6 0.6 0.3], [0.2 0 0.1], [0.3 0.9 0.6], [0.2 0.8 0.9]].

7) CNN MODEL

The output of the dropout layer is received by the CNN model, which is then moved through different layers of the CNN model to perform the task of extracting local (low-level) features from the input text. The working of the CNN model is given in the following way:

a: LEVEL-1 APPLYING FILTERING

During the filtering process, the filter matrix(overlay) is implemented in the review text. The steps included in the

filtering process are described in section 3.4.2. The filtering process for the example sentence is illustrated as follows:

Embedding/Review matrix elements regarding filter matrix window size: 0.1 0 0.5 0.7

Filter matrix (overlay) elements: 0.3 0.4 0.1 0.5

1st Element of feature map: $0.1 \times 0.3 + 0 \times 0.4 + 0.5 \times 0.1 + 0.7 \times 0.5 = 0.43$

Summation of Bias term with feature map elements: $0.43 + 1 = 1.43$

Presenting Nonlinearity using Relu function: Relu(1.43) = max(0, 1.43), where $1.43 > 0$, so output = 1.43.

b: LEVEL-2 APPLYING DOWNSAMPLING

The rectified feature map obtained after passing through the Relu activation function is now used for the downsampling operation. This operation reduces the size of the rectified feature map using $W = \max(L)$. There are three steps involved in down sampling, described as: (i) Selection of a suitable window size (usually 2 or 3), (ii) Selecting a value for a stride that is 2, in our case, where the stride assists in controlling the flow of the window over the rectified feature map, and (iii) Selection of the highest value for the window chosen in the prior step. *Maximum pooling operation regarding 1st element:* In a given pool of numbers, i.e. max (1.43, 1.46, 1.85, 1.47), the maximum number (1.85) is selected as the 1st element of the pooled feature map. Similarly, other elements of the pooled feature map are generated. Finally, when the pooled feature map is generated, it is transferred to the LSTM model for further processing.

8) LSTM MODEL

To solve the issue of long term dependencies associated with earlier models like RNN, LSTM is used. It adds gates, and cell state (long term memory) to RNN. It helps in preserving the context information of the past, which makes it feasible to exploit the context information at the sentence beginning. Therefore, we selected the LSTM model for learning the sequence (time series) data [49]. The LSTM model received input from the previous layer of the CNN model. The calculation of the input text involves 4 major items, named as the forget gate (remember_t), input gate (save_t), output gate (focus_t), and a new cell state (lgtm_t). *Output of 1st LSTM cell:* The LSTM model is comprised of different cells and each cell needs the current input (x_t), prior hidden state (wgm_{t-1}), implements some calculation (see Eqs. 4-9), and finally, the hidden state of the 1st cell (see Fig. 4) is obtained as follows:

Output of Last LSTM Cell: The last cell of LSTM needs the current input (x_t), prior hidden state (wgm_{t-1}), implements some calculation (see Eqs. 4-9), and finally, the hidden state of the last cell (see Fig. 5) is obtained as follows:

Now, the hidden state(wgm_{t-1}) of the last LSTM cell is taken as a final depiction of the input text that is further moved forward to the Softmax(output) layer.

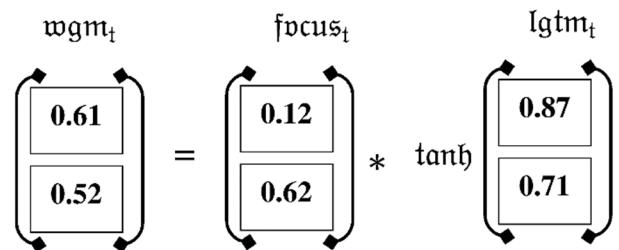


FIGURE 4. Output of 1st LSTM cell.

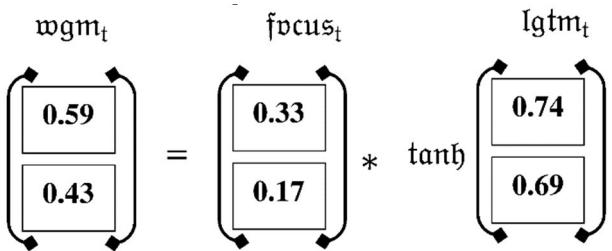


FIGURE 5. Output of last LSTM cell.

9) SOFTMAX LAYER

The softmax layer acts as the last layer of the proposed model by exploiting the softmax activation function for the classification of feature vector(final depiction) into the desired personality classes I-E, N-S, T-F, and J-P. The computation using Eq. 10 is depicted as follows.

$$\begin{aligned} A_1 &= f_1 \times x_1 + f_2 \times x_1 + b \\ A_1 &= 0.8 \times 0.59 + 0.7 \times 0.59 + 0.9 \\ A_1 &= 0.236 + 0.9 = 1.78 \\ A_2 &= f_1 \times x_2 + f_2 \times x_2 + b \\ A_2 &= 0.5 \times 0.43 + 0.9 \times 0.43 + 0.5 \\ A_2 &= 0.0602 + 0.5 = 1.102 \end{aligned}$$

After obtaining the value of A_1 and A_2 , we will apply the softmax activation function using Eq. 11 as follows:

$$\begin{aligned} \text{softmax}(A_1) &= \frac{e(A_1)}{e(A_1) + e(A_2)} \\ \text{softmax}(A_1) &= \frac{e(1.78)}{e(1.78) + e(1.102)} \\ \text{softmax}(A_1) &= \frac{5.92}{8.93} = 0.66 \\ \text{softmax}(A_2) &= \frac{e(A_2)}{\sum_{g=0}^n e(A_1 + A_2)} \\ \text{softmax}(A_2) &= \frac{e(1.102)}{e(1.102) + e(1.78)} \\ \text{softmax}(A_2) &= \frac{3.01}{8.93} = 0.33 \end{aligned}$$

The above-mentioned calculation shows that the highest probability is achieved by the I(Introversion) personality trait. So, the given input text “I am finding the lack of me in

these posts very alarming" is classified as an Introversion personality trait among all the other classes.

Algorithm 1 shows the steps of the CNN+LSTM model pseudo code for personality trait classification.

Algorithm 1 Pseudocode of Proposed Personality Classification Model

```

Step1: Input dataset as csv file.
Step II: Spilt into train ( $R_{train}$ ,  $NR_{train}$ )-test ( $R_{test}$ ,  $NR_{test}$ ) using scikit learn.
Step II: Break tweets in words as tokens using keras tokenizer.
Step III: Build the vocabulary to map integer to each word.
Step IV: Transform each tweet into sequence of integers.
Step V: procedure CNN- LSTM MODEL (Rtrain, NRtrain)
Parameter Initialization: i.max_features = 10000, input_length = 40, embed_dim = 128, classes = 2,
    NEpoch = 3, batch_size = 32
    ii.LSTM units = 280
    iii.CNN: - filters = 8; pool_size = 2, Kernel size = 2,
Build Deep learning model
model = Sequential()
Embedding Layer
model.add(Embedding(max_features, embed_dim, input_length))
Dropout Layer
model.add(Dropout(0.5))
Convolutional Layer
model.add(Conv1D(filters, kernel_size, padding = 'valid', activation = 'relu'))
Maxpooling Layer
model.add(MaxPooling1D(pool_size))
LSTM layer
model.add(LSTM(280, return_sequences = True))
Softmax Layer
model.add(Dense(classes, activation = 'softmax'))
Compile Function
model.compile(loss = binary_crossentropy, optimizer = adamax, metrics = [accuracy])
Model Summary
print(model.summary())
for all epochs in (1: NEpoch) do
    Fit model on train data
    model.fit(X_train_seq, y_train_le, epochs, batch_size = batch_size, validation_data = (X_test_seq, y_test_le))
End for
Evaluate model on test data
acc = model.evaluate(X_test_seq, y_test_le, verbose = 2, batch_size = batch_size)
Output: return acc
End Procedure

```

To insert images in *Word*, position the cursor at the insertion point and either use Insert | Picture | From File or copy the image to the Windows clipboard and then Edit | Paste Special | Picture (with "float over text" unchecked).

IEEE will do the final formatting of your paper. If your paper is intended for a conference, please observe the conference page limits.

IV. RESULTS AND DISCUSSION

The section presents the answers to the posed research questions as follows:

A. ANSWER TO RQ1

To answer RQ1 (How to apply the Deep Learning technique, namely CNN+LSTM, to classifying personality traits from the input text?), the implementation of different CNN+LSTM models with varying parameters is exploited for the classification of input text over distinct personality traits: I-E, N-S, T-F, and J-P. To build various CNN+LSTM models, we used an individual layer with a variant of parameters. Consequently, the efficiency of the classifier is enhanced by fine-tuning the parameters of CNN+LSTM. We applied different values of the "units" parameter of an LSTM layer. Moreover, several, additional parameter values, like pool_size, number of filters, epochs, kernel_size, and padding are listed in Table 5.

TABLE 5. Parameters with values for the CNN+LSTM proposed method regarding classes N-S, I-E, J-P, T-F.

CNN+LSTM Layers	Parameters and their values
CNN Layer:	Convolutional layer: kernel_size=3x3 filters number:32 padding=same Layer of Pooling: pool_size=2x2
Layer of LSTM:	units=110,120,130,140,150,160, 170,180,190,200
Dropout Layer:	rate=0.8
Dense Layer:	nb_classes=2 activation=softmax
Further parameters:	max_features = 2000 length of input sentence = 1703 epochs=7 embed_dim=128 batch_size=32

Table 5 depicts values of different parameters, where values of a number of filters, padding, pool_size, kernel_size, and epochs are set as fixed, while the value of LSTM units varies from 110 to 200.

The parameter arrangements for the individual 10 CNN+LSTM models are shown in Table 6. The values for the LSTM parameter 'units' represent an increasing trend, ranging

TABLE 6. CNN+LSTM model parameter arrangements regarding classes N-S, I-E, J-P, T-F.

Model Name	Batch Size	LSTM units	Convolutional Kernel size
CNN+LSTM1	32	110	3x3
CNN+LSTM2	32	120	3x3
CNN+LSTM3	32	130	3x3
CNN+LSTM4	32	140	3x3
CNN+LSTM5	32	150	3x3
CNN+LSTM6	32	160	3x3
CNN+LSTM7	32	170	3x3
CNN+LSTM8	32	180	3x3
CNN+LSTM9	32	190	3x3
CNN+LSTM10	32	200	3x3

from 110 to 200, whereas the batch size = 32, and convolutional kernel size = 3×3 .

Table 7 illustrates outcomes of different evaluation metrics, namely: f1-score, recall, precision, and accuracy (see Table 7) of the proposed CNN+LSTM model regarding different personality traits I-E, N-S, T(Thinking)-F(Feeling), and J(Judging)-P(Perception). We conducted different experiments to obtain the optimal values of parameters, yielding the best performance of the CNN+LSTM model. In the case of I-E's binary classifier, it was observed that CNN+LSTM2 outperformed other models in terms of f1-score (88%), recall (88%), precision (88%), and accuracy (88%) with parameter settings of batch size = 32, lstm units = 120, and convolutional kernel size = 33. For the N-S binary classifier, it is observed that CNN+LSTM4 outperformed other models in terms of f1-score (91%), recall (91%), precision (91%), and accuracy (91%) with parameter setting batch size = 32, lstm units = 140, and convolutional kernel size = 33. In the case of the T-F binary classifier, it is noted that CNN+LSTM3 outperformed other models in terms of f1-score (85%), recall (85%), precision (85%), and accuracy (85%) with parameter settings of batch size = 32, lstm units=130, and convolutional kernel size = 33. Finally, for the J (Judging)-P (Perception) binary classifier, it is observed that CNN+LSTM3 outperformed other models in terms of f1-score (80%), recall (80%), precision (80%), and accuracy (80%) with parameter settings of batch size = 32, lstm units = 130, and convolutional kernel size = 33.

The performance of CNN+LSTM in terms of accuracy is presented in Fig. 6. The horizontal side of the graph

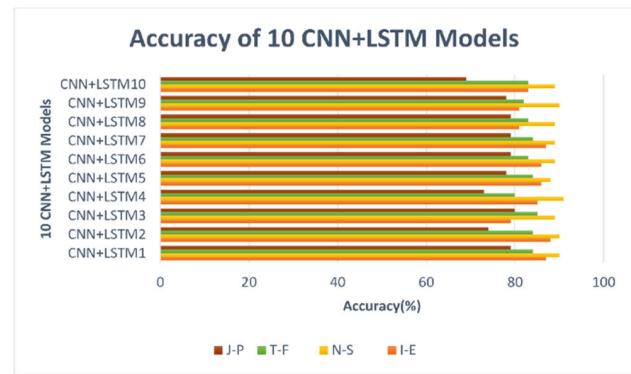


FIGURE 6. Accuracy regarding 10 CNN+LSTM Models regarding personality trait classification.

shows the accuracy measure, and the vertical side shows 10 CNN+LSTM models.

Table 8 presents the scores of loss, accuracy, and training time related to the classes I-E, N-S, T-F, and J-P. In the case of the I-E binary classifier, it is found that the rise in lstm unit value results in the decline in accuracy for a CNN+LSTM model. The model's accuracy tends to increase at the lstm unit of 120, which is considered as an optimal value for the lstm unit parameter, and it remains the same at the lstm unit of 170. Additionally, the score loss grows as the decline in accuracy occurs. For the N (Intuition)-S (Sensing) binary classifier, it is noted that the rise in lstm unit value results in a decline in the CNN+LSTM model's accuracy. The model's accuracy tends to increase at the lstm unit of 140, which is considered as an optimal value for the lstm unit parameter, and it remains the same at lstm unit = 190. In addition, the rise in scores losses occurred due to the decline in accuracy. In the case of the T-F binary classifier, it is noted that the rise in the LSTM unit's value results in a decline in the CNN+LSTM model's accuracy. The model's accuracy tends to increase at the lstm unit 130, which is considered as an optimal value for the lstm unit parameter, and it remains the same at the lstm units 120, 150, and 170. Furthermore, the rise in scores losses occurred due to the decline in accuracy.

Finally, for the J-P binary classifier, it is noted that the rise in lstm unit value results in a decline in the CNN+LSTM model's accuracy. The model's accuracy tends to increase at the lstm unit of 130, which is considered as an optimal value for the lstm unit parameter, and it remains the same at the lstm units of 160, 170, and 180. Also, a rise in the loss score occurred due to the decline in accuracy.

1) PERFORMANCE EVALUATION OF PROPOSED CNN+LSTM MODEL FOR DIFFERENT PERSONALITY CUES

In this section, we report the findings of the experiment conducted to evaluate the efficiency of the proposed deep learning model with respect to different personality traits like (i) Extroversion (E), (ii) Introversion (I), (iii) Sensing (S), (iv) Intuition (N), (v) Feeling (F), (vi) Thinking (T),

TABLE 7. CNN+LSTM evaluation metrics results regarding classes N-S, I-E, J-P, and T-F.

Model Name	F1 Score (%)				Recall (%)				Precision (%)			
	I-E	N-S	T-F	J-P	I-E	N-S	T-F	J-P	I-E	N-S	T-F	J-P
CNN+LSTM1	87	90	84	79	87	90	84	79	87	90	84	79
CNN+LSTM2	88	90	84	74	88	90	84	74	88	90	84	74
CNN+LSTM3	79	89	85	80	79	89	85	80	79	89	85	80
CNN+LSTM4	85	91	80	73	85	91	80	73	85	91	80	73
CNN+LSTM5	86	88	84	78	86	88	84	78	86	88	84	78
CNN+LSTM6	86	89	83	79	86	89	83	79	86	89	83	79
CNN+LSTM7	87	89	84	79	87	89	84	79	87	89	84	79
CNN+LSTM8	81	89	83	79	81	89	83	79	81	89	83	79
CNN+LSTM9	81	90	82	78	81	90	82	78	81	90	82	78
CNN+LSTM10	83	89	83	69	83	89	83	69	83	89	83	69

(vii) Perceiving (P), and (viii) Judging (J). For this purpose, we designed 4 binary classifiers, namely I(Introversion)-E (Extroversion), N (Intuition)-S (Sensing), T(Thinking)-F (Feeling), and J(Judging)-P (Perception).

a: EXPERIMENT#1-I-E

We conducted experiment #1 to evaluate the proposed CNN+LSTM model's efficiency with respect to I (Introversion)-E (Extroversion) personality traits, and the results are reported in Table 9. It is evident that the CNN+LSTM model attained better performance for the “I (Introversion)” personality trait in terms of recall (0.93), precision (0.92), f1-score (0.92) and total accuracy of 0.88%.

b: EXPERIMENT#2-N-S

We conducted experiment #2, to evaluate the efficiency of the proposed CNN+LSTM model with respect to N (Intuition)-S (Sensing) personality traits. The results presented in Table 10 show that the CNN+LSTM model attained better performance for the “N” personality trait with recall (0.95), precision (0.94), and f1-score (0.95), and. The total accuracy is 0.91%.

c: EXPERIMENT#3-T-F

We conducted experiment #3 to evaluate the efficiency of the proposed model (CNN+LSTM) for T-F personality traits.

The results reported in Table 11 show that the proposed CNN+LSTM model attained better performance for the “F (Feeling)” personality trait in the cases of recall (0.87), precision (0.87), and f1-score (0.87). The total accuracy is 0.85%.

d: EXPERIMENT#4-J-P

We conducted experiment # 4 to evaluate the efficiency of the proposed CNN+LSTM model with respect to J (Judging)-P (Perception) personality traits. The results given in Table 12 show that the CNN+LSTM model attained better performance for the “P” personality trait with recall (0.92), f1-score (0.85) and overall accuracy of 0.80%. Furthermore, the model attained a maximum precision (0.82) for the “J (Judging)” personality trait.

B. ANSWER TO RQ2

To answer RQ2 (What is the efficiency of the proposed technique w.r.t other machine learning and deep learning techniques?), experimentation was performed to evaluate the efficiency of the proposed CNN+LSTM model that exploits word embedding, with respect to comparing machine learning classifiers that utilize the classical feature representation scheme, i.e., Bag-of-Words (Countvectorizer and TF-IDF). We implemented different ML and DL techniques for

TABLE 8. Score of loss with accuracy and time of training regarding classes N-S, I-E, J-P, and T-F.

Name Model	of	Score of Loss		Accuracy(%)		LSTM units (N-S, I-E, J-P, and T-F)
		I-E T-F	N-S J-P	I-E T-F	N-S J-P	
CNN+LSTM1		0.66 0.72	0.60 0.80	87 84	90 79	110
CNN+LSTM2		0.66 0.72	0.62 0.87	88 84	90 74	120
CNN+LSTM3		0.78 0.70	0.63 0.81	79 85	89 80	130
CNN+LSTM4		0.73 0.75	0.59 0.89	85 80	91 73	140
CNN+LSTM5		0.67 0.71	0.64 0.83	86 84	88 78	150
CNN+LSTM6		0.69 0.72	0.62 0.81	86 83	89 79	160
CNN+LSTM7		0.65 0.73	0.89 0.83	87 84	89 79	170
CNN+LSTM8		0.78 0.73	0.66 0.82	81 83	89 79	180
CNN+LSTM9		0.77 0.76	0.64 0.83	81 82	90 78	190
CNN+LSTM10		0.71 0.73	0.62 0.93	83 83	89 69	200

TABLE 9. Performance evaluation results of proposed (CNN+LSTM) model.

Personality Cues	F1- score	Recall	Precision	Accuracy(%)
I(Introversion)	0.92	0.93	0.92	
E(Extroversion)	0.72	0.70	0.73	0.88

TABLE 10. Performance evaluation results for proposed method (CNN+LSTM).

Personality Cues	F1- score	Recall	Precision	Accuracy (%)
N(Intuition)	0.95	0.95	0.94	
S(Sensing)	0.62	0.58	0.66	0.91

personality traits, and their results are presented in Table 13 and Table 14.

1) COMPARISON OF PROPOSED CNN+LSTM WITH MACHINE LEARNING TECHNIQUES EXPLOIT-ING CLASSICAL FEATURES

To compare the proposed CNN+LSTM method with classical machine learning methods, we performed experiments, reported in the rest of this section.

TABLE 11. Performance evaluation results for proposed CNN+LSTM method.

Personality Cues	F1- score	Recall	Precision	Accuracy (%)
T(Thinking)	0.84	0.84	0.84	0.85
F(Feeling)	0.87	0.87	0.87	

TABLE 12. Performance evaluation results of proposed (CNN+LSTM) method.

Personality Cues	F1- score	Recall	Precision	Accuracy(%)
J(Judging)	0.70	0.61	0.82	
P(Perception)	0.85	0.92	0.79	0.80

TABLE 13. Efficiency comparison between machine learning classifiers and proposed technique.

Classifier/	Accuracy (%)		Precision (%)		Recall (%)		F1-score (%)	
Model	I-E T-F	N-S J-P	I-E T-F	N-S J-P	I-E T-F	N-S J-P	I-E T-F	N-S J-P
SVM	78.69 81.19	88.48 68.66	0.78 0.81	0.88 0.68	0.78 0.81	0.88 0.68	0.78 0.81	0.88 0.68
Sengupta and Ghosh, [50]								
KNN	78.46 56.8	86.41 58.18	0.78 0.56	0.86 0.58	0.78 0.56	0.86 0.58	0.78 0.56	0.86 0.58
LR	79.03 78.46	87.1 74.42	0.79 0.78	0.87 0.74	0.79 0.78	0.87 0.74	0.79 0.78	0.87 0.74
Sengupta and Ghosh, [50]								
RF	75.69 60.71	86.06 56.91	0.75 0.60	0.86 0.56	0.75 0.60	0.86 0.56	0.75 0.60	0.86 0.56
DT	77.07 70.16	81.8 68.89	0.77 0.70	0.81 0.68	0.77 0.70	0.81 0.68	0.77 0.70	0.81 0.68
XGBoost	83.18 75.58	88.71 73.85	0.83 0.75	0.88 0.73	0.83 0.75	0.88 0.73	0.83 0.75	0.88 0.73
Proposed (CNN+ LSTM)	88 85	91 80	88 85	91 80	88 85	91 80	88 85	91 80

Machine Learning Techniques Exploiting Classical Features: Different classical feature representation schemes, such as Countvectorizer and TF-IDF, are evaluated. The Countvectorizer scheme inspects the presence of words within a vocabulary by exploiting the bag-of-words technique, and TF-IDF is a numeric estimator that calculates

TABLE 14. Performance comparison between variants of deep learning and proposed technique.

Classifier/ Model		Accuracy (%)		Precision (%)		Recall (%)		F1-score (%)	
		I-E		N-S		I-E		N-S	
		T-F	J-P	T-F	J-P	T-F	J-P	T-F	J-P
Individual Iqbal Siddiqui,[62]	CNN and	81 84	88 75	81 84	88 75	81 84	88 75	81 84	88 75
Individual Hernandez Scott, [63]	LSTM and	79 57	87 66	79 57	87 66	79 57	87 66	79 57	87 66
Individual Hernandez Scott, [63]	BILSTM and	78 66	87 65	78 66	87 65	78 66	87 65	78 66	87 65
Individual Hernandez Scott, [63]	RNN and	78 54	87 61	78 54	87 61	78 54	87 61	78 54	87 61
Proposed (CNN+LSTM)		88 85	91 80	88 85	91 80	88 85	91 80	88 85	91 80

the score of words by computing the frequency of words within a document and a group of documents. Experiments are conducted using various ML algorithms, such as SVM, KNN, LR, RF, DT, and XGBoost, that exploit classical features like countvectorizer and tf-idf. From the experimental comparison, it is found that among different machine learning classifiers, XGBoost attained a maximum accuracy of 83.18%, whereas KNN attained a minimum accuracy of 78.6% for the I-E personality trait. In the case of the N-S class, XGBoost attained the highest accuracy (88.71%), and the minimum accuracy (81.8%) was attained by DT. For class T-F, SVM attained a maximum accuracy of 81.19%, and a minimum accuracy of 56.8% was attained by KNN. Finally, for class T-F, LR attained a maximum accuracy of 74.42%, and a minimum accuracy of 56.91% was attained by RF.

Proposed CNN+LSTM Model: We conducted experiments for personality trait classification by applying the proposed model (CNN+LSTM) that uses word embedding as a feature representation scheme. It is observed that the proposed CNN+LSTM model yields improved performance in relation to the compared machine learning classifiers, such as SVM, KNN, LR, RF, DT, and XGBoost with the classical Bag of words representation approach (Table 13), and a variant of deep neural network techniques like Individual LSTM, Individual CNN, Individual BILSTM, Individual RNN (see Table 14).

The results presented in Table 13 show that the proposed model outperformed the machine learning methods due to the following limitations of ML classifiers: (i) In a bag of words approach, an individual word is considered as a one-hot vector, and because the data is sparse, the approach is incapable of capturing different linguistic word aspects. When the data is sparse, then the length of the one-hot vector is similar

to the size of the vocabulary that contains zero values, excluding a dimension that contains 1 value. For instance, in a bag of words approach, the word “Sunday” in sparse depiction is given as [0, 0, 0, 0, 0, 1] [27], (ii) The BOW technique does not consider the word context within the given text, i.e. it recognizes a single word as an independent unit [51]. Furthermore, due to the following benefits of word embedding over the traditional bag of words approach, the CNN+LSTM performed efficiently with respect to the compared machine learning models. These are listed as follows: (i) In word embedding, the model is learned to represent an individual word in a real-valued vector (continuous) through training of each sentence. Therefore, word embedding efficiently performs information semantic encoding with respect to the classical feature set, namely bag of words [51], (ii) Using similar words, the classification of the current word, that is unavailable within the training set, becomes achievable since the real-valued feature depiction scheme enables us to map close words towards one another having a similar meaning [51], (iii) Semantic sparsity occurring within a short text is managed by the dense feature representation scheme, i.e. word embedding [52]. Word embedding-based techniques perform effectively over BOW-based methods due to the fact that the information encoding related to the rare words in the training dataset is performed efficiently by word embedding. Furthermore, the reason for the better performance of the CNN+LSTM neural network method over the compared machine learning models is that the hierarchical feature abstraction of the user-created content towards the individual layer [53]. At the convolutional layer, the learning of best (local) features is conducted [54], while at the pooling layer, it maintains the salient local features [55], and finally, at the LSTM layer, it captures the global features [56]. On the other hand, machine learning deals with a less efficient feature engineering process, which is manual [53]. The self-learning ability of deep learning models enables them to achieve significant accuracy and fast processing speed with respect to compared machine learning classifiers exploiting classical features [57]. The reason behind the better performance of the CNN+LSTM model for personality prediction with respect to the machine learning algorithms is that it exploits word embedding feature representation approach, executes automated feature understanding, and works at a deep level of layers.

2) COMPARISON OF PROPOSED CNN+LSTM WITH VARIANTS OF DEEP NEURAL NETWORK METHODS

To compare the proposed CNN+LSTM model with variants of deep neural network methods, we performed different experiments and evaluated the results as follows:

a: VARIANTS OF DEEP LEARNING MODELS

We conducted experiments on different deep neural network models, specifically Individual LSTM, Individual CNN, Individual BILSTM, and Individual RNN, each of which exploits

an advanced feature representation scheme, namely word embedding [58].

The experimental results (Table 14) show that, with respect to other deep learning models, CNN attained a maximum accuracy of 81% for I-E, 88% for N-S, 84% for T-F, and 75% for J-P. It is due to the fact that during feature extraction, CNN provides the service of getting local association between connecting words. RNN and BILSTM attained a minimum accuracy of 78% for I-E. In the case of N-S, a minimum accuracy of 87% was achieved by three models, namely LSTM, BILSTM, and RNN, because the LSTM and BILSTM models are not good at capturing important features [59], [60]. For T-F, RNN performed poorly with an accuracy of 54%, and finally, for J-P, the lowest performance of 61% was attained by RNN. The reason for the low performance of RNN is that its memory storage capacity is small, i.e. short-term memory [61].

In the following sections, we present a comparison of the proposed model (CNN+LSTM) with different deep neural network models, namely Individual LSTM, Individual CNN, and Individual BILSTM.

b: COMPARING PROPOSED CNN+LSTM WITH INDIVIDUAL CNN

In this experiment, we evaluated the performance of the proposed model (CNN+LSTM) with respect to individual CNN for personality trait classification. The results presented in Table 14 show that CNN gives a poor performance with precision (81%), recall (81%), f1-score (81%), and accuracy (81%) for class I-E. In the case of N-S, the performance of CNN's model is given as precision (84%), recall (84%), f1-score (84%), and accuracy (84%). For class T-F, the model's Precision = 84%, Accuracy = 84%, F1-score = 84%, and Recall = 84%. Finally, in the case of J-P, CNN gives Accuracy = 75%, Precision = 75%, Recall = 75%, and F1-score = 75%. CNN is capable of extracting important features belonging to the input text, while it lacks the ability to hold long-term dependencies [39].

c: COMPARING PROPOSED CNN+LSTM WITH INDIVIDUAL LSTM

In this experiment, the performance of the proposed model is evaluated with respect to individual LSTM. Table 14 shows that the individual LSTM yields the worst performance with Accuracy = 79%, Precision = 79%, Recall = 79%, and F1-score = 79% for class I-E. For N-S, LSTM's performance is as follows: Accuracy = 87%, Precision = 87%, Recall = 87%, and F1-score = 87%. For class T-F, the LSTM model's performance is: Accuracy = 57%, Precision = 57%, Recall = 57%, and F1-score = 57%. Lastly, in the case of J (Judging)-P (Perception), the LSTM provides an Accuracy = 66%, Precision = 66%, Recall = 66%, and F1-score = 66%. The LSTM is efficient in such tasks that require a long time lag [64], whereas it is not good for capturing important features [59].

d: COMPARING THE PROPOSED CNN+LSTM METHOD WITH INDIVIDUAL BILSTM

In this experiment, we compared the performance of the proposed model (CNN+LSTM) with the individual BILSTM model. The results reported in Table 14 show that the individual BILSTM attained low performance with a Precision = 78%, Accuracy = 78%, F1-score = 78% and Recall = 78% for class I-E. In the case of N-S, BILSTM's performance is given as: Precision = 87%, Recall = 87%, F1-score = 87%, and Accuracy = 87%. For class T-F, the model attained Accuracy = 66%, Precision = 66%, Recall = 66%, and F1-score = 66%. Lastly, in the case of J-P, the BILSTM yields an Accuracy = 65%, Precision = 65%, Recall = 65%, and F1-score = 65%. The BILSTM has the attribute of exploring the bidirectional long-term dependencies regarding the input text. However, it is not efficient for the task of feature extraction [59].

e: COMPARING PROPOSED CNN+LSTM WITH INDIVIDUAL RNN

In this experiment, we evaluated the performance of individual RNNs with the proposed CNN+LSTM. The results presented in Table 14 show that individual RNNs have shown the worst efficiency with regard to our proposed CNN+LSTM model with a recall = 78%, precision = 78%, accuracy = 78%, and F1-score = 78% for class I-E. In the case of N-S, RNN attained: Precision = 87%, Recall = 87%, Accuracy = 87%, and F1-score = 87%. For class T (Thinking)-F (Feeling), the model's Accuracy = 54%, Precision = 54%, Recall = 54%, and F1-score = 54%. Lastly, in the case of J-P, the RNN provides an Accuracy = 61%, Precision = 61%, F1-score = 61%, and Recall = 61%. The RNN is efficient in passing learned information to the upcoming layer, but it suffers from the issue of vanishing gradient, as the two-time step gap increases [39].

The abovementioned results indicate better results for our proposed CNN+LSTM method when a comparison is performed with different DL models, such as individual LSTM, individual CNN, individual RNN, and individual BILSTM. The proposed CNN+LSTM model attained better performance, i.e. 88% for I-E, 91% for N-S, 85% for T-F, and 80% for J-P, with respect to the compared DL models. It is due to the fact that the proposed method attains the benefits of the two deep learning models, i.e. CNN and LSTM. In the case of CNN, it efficiently extracts significant information related to the input text and LSTM is efficient at holding information for a long time period without information degeneration [39]. Hence, it is noted that for personality traits, our proposed approach has shown improved performance with respect to other variants of Deep Learning models, such as the individual LSTM model, individual CNN model, individual BILSTM, and individual RNN.

C. ANSWER TO RQ3

To address the research question: “How to estimate the efficiency of the proposed technique regarding personality

TABLE 15. Comparing metrics result for the proposed approach with baseline studies for a personality trait classification.

Study	Technique	Dataset	Classifier	Obtained Results				
				Metrics	I/E	S/N	F/T	J/P
Majumder et al. [9]	Deep Learning	MBTI	NB	Accuracy	77	86.2	77.9	62.3
				Recall	77.0	83.3	54.1	61.8
				Precision	79.5	81.3	78.6	68.2
				F1-score	81.9%	80.4%	69.0%	64.8%
Khan et al. [65]	Machine Learning	MBTI	XGBoost	Accuracy	77.35	78.93	73.89	73.71
				Recall	76.44	79.06	73.30	75.75
				Precision	78.19	78.78	75.32	73.37
				F1-Score	77.30	78.92	74.30	74.54
Proposed	Deep Learning	MBTI	CNN+LSTM	Accuracy	88	91	85	80
				Recall	88	91	85	80
				Precision	88	91	85	80
				F1-Score	88	91	85	80

trait classification w.r.t baseline studies?,” the experiment was conducted for performance comparison of the proposed model with respect to the baseline study conducted by [9]. In Table 15, experimental results are presented.

In this experiment, we compared the performance of our model with the baseline models.

Majumder et al. [9] used the Convolutional Neural Network method (CNN) to detect personality traits. Experimental results recorded in Table 15 indicate that the proposed approach performed better than the baseline study with an accuracy of 88% for I (Introversion)-E (Extroversion), 91% for N (Intuition)-S (Sensing), and 80% for T (Thinking)-F (Feeling), and 80% for J (Judging)-P (Perception).

Khan et al. [65] applied the XGBoost classifier to classify personality traits using the MBTI dataset. They achieved maximum accuracy (83%). However, the use of conventional feature sets associated with ML classifiers can further be improved by applying deep learning techniques with automated feature engineering.

Proposed (Our) Model’s Performance: The justification behind our model’s performance enhancement is that we have exploited an “LSTM” model which is efficient at saving information for a long time span among the word sequences. So, the information maintained for a large period of time is

very beneficial for text classification/prediction tasks. In contrast, an individual CNN layer is not capable of retaining the sequential information underlying text, which is needed for the problems regarding text classification. Moreover, for CNN to produce effective results, it is required to use a large dataset.

D. SIGNIFICANCE TEST

In this section, experimentation was conducted to examine whether the CNN+LSTM (proposed model) exploiting advanced features (word embedding) is statistically significant with respect to the classical machine learning model using traditional features (BOW features) and did not happen by chance.

A random extraction of 868 input reviews from the corpus is performed, where each individual input review is classified by the CNN+BILSTM vs XGBoost for I (Introversion)-E (Extroversion) class. During the experimental setting, the null and alternate hypothesis are presented.

H₀: Identical rate of error for both classifiers.

H_A: Significantly different rate of error for both classifiers.

The mathematical representation of McNemar’s statistics is expressed as follows (Eq. 12):

$$\chi^2 = (|m - n| - 1)/(m + n) \quad (12)$$

While χ^2 shows the chi-squared statistics, variables n and m indicate the discordant, and the degree of freedom is represented by 1 value in Eq. 12.

1) DISCUSSION

In the first experiment, we examined and reported (see Table 13) the performance of the XGBoost classifier using the classical feature scheme for I(Introversion)-E(Extroversion) personality class. The XGBoost classifier yielded low performance in terms of each evaluation metric like F1-score, recall, accuracy, and precision. The experimental results revealed the poor performance of the XGBoost classifier exploiting the classical feature set (BOW features) for personality trait detection from the textual content. In the next experiment, the CNN+LSTM model with word embedding features produced considerably improved results (Table 14). The classifier performed a prediction of personality traits underlying text data with an accuracy of 88%.

The significance test was conducted to validate that there exists a significant difference between CNN+LSTM (DL) exploiting word embedding based features, and XGBoost (ML) exploiting traditional features. The input text on which the two models differed was 122 (models having distinct feature types exhibit distinct misclassification performance) as shown in Table 13. McNemar’s test is applied to compute the p-value. A value of 1.4 is attained by Chi-Squared, and 1 is the degree of freedom, and 0.239 (two-tailed p-value). Finally, the null hypothesis is rejected due to a low p-value of 0.2390.5, and the alternative hypothesis is accepted: the proposed CNN+LSTM method, exploiting word embedding, is statistically significantly more

significant than that of the XGBoost model, exploiting traditional BOW features.

The statistical analysis mentioned above revealed that the features based upon word embedding have significantly enhanced the effectiveness of the proposed model, namely CNN+LSTM regarding personality trait classification (see Table 16).

TABLE 16. Computation of performance difference using significance test between proposed CNN+LSTM(DL) and XGBoost(ML) for I-E class.

	CNN+LSTM (proposed) Correct Classification	CNN+LSTM (proposed) Misclassification	Total
XGBoost Correct Classification	676	54	730
XGBoost Misclassification	68	70	138
Total	744	124	868
McNemar's Test: From mcnemar's test we found that a chi-squared statistic = 1.4, with 1 degree of freedom, and the 2-tailed p-value is 0.239. Thus, an alternative hypothesis is accepted and the null hypothesis is rejected.			

We also conducted experiments to inspect whether our proposed (CNN+LSTM) model, exploiting word embedding feature, is statistically significant than the XGBoost for different combinations of personality traits, such as N(Intuition)-S(Sensing), LR for J-P, and SVM for T-F. The results depict that in case of J-P, T-F, and N-S, the advanced word embedding features significantly enhanced the effectiveness of the proposed CNN+LSTM model regarding personality trait classification problem.

V. CONCLUSION

In this work, we investigated the task of personality trait classification from textual content. To accomplish the research task, we proposed applying a deep learning model, namely CNN+LSTM. The proposed study includes the following modules: (i) acquiring data, (ii) pre-processing of data, and (iii) implementing the deep neural network.

The proposed CNN+LSTM model for personality trait classification is a merger of CNN and LSTM that assists in classifying the input text into different personality traits like I-E, N-S, T-F, and J-P. The main emphasis of the CNN model is to extract and retain the local features using a convolutional and max-pooling layer. CNN acts as a robust tool for choosing the best features that enhance the prediction accuracy. The LSTM model preserves the prior information regarding context, which helps to exploit significant context information at the start of a sentence. Its benefit is that it takes sequential information through the examination of prior data. After receiving the final representation of an input sentence, it is classified among the different personality traits. The experiments with different machine learning and deep learning models are also conducted and their results are recorded on the personality trait dataset. The results show that

the proposed CNN+LSTM model for personality trait classification produced improved results in terms of improved accuracy (88% for I-E, 91% for N-S, 85% for T-F, and 80% for J-P, precision (88% for I-E, 91% for N-S, 85% for T-F, and 80% for J-P, and f1-score (88% for I-E, 91% for N-S, 85% for T-F, and 80% for J-P, and the proposed CNN+LSTM model for personality trait classification (88% for The information obtained from this research acts as best practices for the selection, management, and optimization of their policies, services, and products.

A. LIMITATIONS

The possible limitations of the proposed work are given as follows: (i) In this study, we performed a personality trait classification on limited personality traits pertaining to only textual content in the English language. (ii) The work is limited to random word embedding without the exploitation of several word representation models like Glove, Fasttext, and word2vec, (iii) The attention mechanism was not introduced with CNN+LSTM that assists in extracting relative significant features, (iv) Other combinations of deep learning models like CNN+Bi-LSTM, CNN+GRU, Bi-LSTM+CNN, and CNN+RNN are not applied for personality trait classification. (v) The focus of the current research content is only on the MBTI dataset for personality trait classification. (vi) The limited number of machine learning classifiers is used for experimentation that needs to be further extended with other machine learning classifiers.

B. FUTURE DIRECTIONS

The possible future directions of the work are as follows: (i) The work can be extended by conducting experiments in other languages on an extended set of personality traits using non-textual content such as images and videos, (ii) In future work, we can exploit other pre-trained word representation schemes like Glove, word2vec, and Fasttext regarding word embedding layer, (iii) Introducing an attention mechanism regarding personality classification may enhance the system's performance, (iv) We will explore different combinations of deep neural networks like CNN+Bi-LSTM, CNN+GRU, Bi-LSTM+CNN, and CNN+RNN for personality trait classification, and CNN+Bi-LSTM+CNN for facial recognition. (v) In addition to the MBTI dataset, other different datasets regarding personality trait classification tasks can be exploited.

It is planned to collect additional data and apply the proposed methods to some larger corpora in order to test the effectiveness of the proposed approaches on massive data records, (vi) The combination of other deep neural network frameworks will better handle the problem of personality trait classification tasks. Thus, in the future, we will apply other neural networks, (vii) In the future, we will focus on applying ensemble methods to enhance system performance, and (viii) We will work on including further base models and by searching for other parameters that may assist in enhancing the overall accuracy of the proposed work.

ACKNOWLEDGMENT

The open access funding is by the Publication Fund of TU Dresden.

REFERENCES

- [1] H. Ahmad, M. Z. Asghar, A. S. Khan, and A. Habib, "A systematic literature review of personality trait classification from textual content," *Open Comput. Sci.*, vol. 10, no. 1, pp. 175–193, Jul. 2020.
- [2] H. Ahmad, M. Z. Asghar, F. M. Alotaibi, and I. A. Hameed, "Applying deep learning technique for depression classification in social media text," *J. Med. Imag. Health Informat.*, vol. 10, no. 10, pp. 2446–2451, Oct. 2020, doi: [10.1166/jmihi.2020.3169](https://doi.org/10.1166/jmihi.2020.3169).
- [3] R. Katarya and P. Srinivas, "Predicting heart disease at early stages using machine learning: A survey," in *Proc. Int. Conf. Electron. Sustain. Commun. Syst. (ICESC)*, Jul. 2020, pp. 302–305.
- [4] Rahul and R. Katarya, "A review: Predicting the performance of students using machine learning classification techniques," in *Proc. 3rd Int. Conf. I-SMAC*, Dec. 2019, pp. 36–41.
- [5] G. Kou, Y. Xu, Y. Peng, F. Shen, Y. Chen, K. Chang, and S. Kou, "Bankruptcy prediction for SMEs using transactional data and two-stage multiobjective feature selection," *Decis. Support Syst.*, vol. 140, Jan. 2021, Art. no. 113429.
- [6] A. Kazameini, S. Fatehi, Y. Mehta, S. Etemadi, and E. Cambria, "Personality trait detection using bagged SVM over BERT word embedding ensembles," 2020, *arXiv:2010.01309*.
- [7] L. Liu, D. Preotiuc-Pietro, Z. R. Samani, and M. E. Ungar, "Analyzing personality through social media profile picture choice," in *Proc. Int. AAAI Conf. Social Media (ICWSM)*, 2016, pp. 211–220.
- [8] All things Statista. *Number of Monthly Active Twitter Users Worldwide From 1st Quarter 2010 to 1st Quarter 2019*. Retrieved From. Accessed: Jan. 18, 2020. [Online]. Available: <https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/>
- [9] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria, "Deep learning-based document modeling for personality detection from text," *IEEE Intell. Syst.*, vol. 32, no. 2, pp. 74–79, Mar. 2017.
- [10] D. Xue, L. Wu, Z. Hong, S. Guo, L. Gao, Z. Wu, X. Zhong, and J. Sun, "Deep learning-based personality recognition from text posts of online social networks," *Int. J. Speech Technol.*, vol. 48, no. 11, pp. 4232–4246, Nov. 2018.
- [11] M. Osama and S. R. El-Beltagy, "A transfer learning approach for emotion intensity prediction in microblog text," in *Proc. Int. Conf. Adv. Intell. Syst. Inform.* Cham, Switzerland: Springer, Oct. 2019 pp. 512–522.
- [12] A. Khattak, M. Z. Asghar, Z. Ishaq, W. H. Bangyal, and I. A. Hameed, "Enhanced concept-level sentiment analysis system with expanded ontological relations for efficient classification of user reviews," *Egyptian Informat. J.*, vol. 3, pp. 1–17, Apr. 2021, doi: [10.1016/j.eij.2021.03.001](https://doi.org/10.1016/j.eij.2021.03.001).
- [13] S. Ahmad, M. Z. Asghar, F. M. Alotaibi, and S. Khan, "Classification of poetry text into the emotional states using deep learning technique," *IEEE Access*, vol. 8, pp. 73865–73878, 2020.
- [14] M. Z. Asghar, A. Sattar, A. Khan, A. Ali, F. Masud Kundi, and S. Ahmad, "Creating sentiment lexicon for sentiment analysis in Urdu: The case of a resource-poor language," *Expert Syst.*, vol. 36, no. 3, Jun. 2019, Art. no. e12397.
- [15] F. M. Alotaibi, M. Z. Asghar, and S. Ahmad, "A hybrid CNN-LSTM model for psychopathic class detection from tweeter users," *Cognit. Comput.*, vol. 13, no. 3, pp. 709–723, Mar. 2021.
- [16] A. Khattak, A. Habib, M. Z. Asghar, F. Subhan, and I. A. R. Habib, "Applying deep neural networks for user intention identification," *Soft Comput.*, vol. 25, no. 3, pp. 2191–2220, Feb. 2021.
- [17] M. Z. Asghar, A. Khan, S. R. Zahra, S. Ahmad, and F. M. Kundi, "Aspect-based opinion mining framework using heuristic patterns," *Cluster Comput.*, vol. 22, no. S3, pp. 7181–7199, May 2019.
- [18] K. Ilimini and T. G. I. Fernando, "Persons' personality traits recognition using machine learning algorithms and image processing techniques," *Adv. Comput. Sci., Int. J.*, vol. 5, no. 1, pp. 40–44, 2016.
- [19] K. Subramanian, V. B. Radhakrishnan, and S. Ramasamy, "Database independent human emotion recognition with meta-cognitive neuro-fuzzy inference system," in *Proc. 9th Int. Conf. Intell. Sensors Netw. Inf. Process. (ISSNIP)*, Apr. 2014, pp. 1–6.
- [20] G. Shivakumar and P. A. Vijaya, "Facial expression based human emotion recognition with live computer response," *Int. J. Comput. Sci. Inf. Technol.*, vol. 4, pp. 81–84, Jan. 2011.
- [21] D. Sharma, M. Sabharwal, V. Goyal, and M. Vij, "Sentiment analysis techniques for social media data: A review," in *Proc. 1st Int. Conf. Sustain. Technol. Comput. Intell.* Singapore: Springer, 2020, pp. 75–90.
- [22] F. Celli, "Unsupervised personality recognition for social network sites," in *Proc. 6th Int. Conf. Digit. Soc.*, 2012, pp. 59–62.
- [23] E. Kafeza, A. Kanavos, C. Makris, and P. Vikatos, "T-PICE: Twitter personality based influential communities extraction system," in *Proc. IEEE Int. Congr. Big Data*, Jun. 2014, pp. 212–219.
- [24] R. S. Kramer, J. E. King, and R. Ward, "Identifying personality from the static, nonexpressive face in humans and chimpanzees: Evidence of a shared system for signaling personality," *Evol. Hum. Behav.*, vol. 32, no. 3, pp. 179–185, 2011.
- [25] J. Asghar, S. Akbar, M. Z. Asghar, B. Ahmad, M. S. Al-Rakhami, and A. Gumaei, "Detection and classification of psychopathic personality trait from social media text using deep learning model," *Comput. Math. Methods Med.*, vol. 2021, pp. 1–10, Apr. 2021.
- [26] P. H. Arnoux, A. Xu, N. Boyette, J. Mahmud, R. Akkiraju, and V. Sinha, "25 tweets to know you: A new model to predict personality with social media," in *Proc. 11th Int. AAAI Conf. Social Media (ICWSM)*, 2017, pp. 472–475.
- [27] H. Christian, D. Suhartono, A. Chowanda, and K. Z. Zamli, "Text based personality prediction from multiple social media data sources using pre-trained language model and model averaging," *J. Big Data*, vol. 8, no. 1, p. 68, Dec. 2021, doi: [10.1186/s40537-021-00459-1](https://doi.org/10.1186/s40537-021-00459-1).
- [28] A. Kumar and A. Jaiswal, "Image sentiment analysis using convolutional neural network," in *Proc. Int. Conf. Intell. Syst. Design Appl.* Cham, Switzerland: Springer, Dec. 2017, pp. 464–473.
- [29] G. Loyer, (2019). *Long Short-Term Memory: From Zero to Hero with PyTorch*. [Online]. Available: <https://blog.floydhub.com/long-short-term-memory-from-zero-to-hero-with-pytorch/>
- [30] M. Jolly, (2017). *(MBTI) Myers-Briggs Personality Type Dataset*. [Online]. Available: <https://www.kaggle.com/datasnaek/mbti-type>
- [31] A. Khattak, M. U. Asghar, U. Batool, M. Z. Asghar, H. Ullah, M. Al-Rakhami, and A. Gumaei, "Automatic detection of citrus fruit and leaves diseases using deep neural network model," *IEEE Access*, vol. 9, pp. 112942–112954, 2021, doi: [10.1109/ACCESS.2021.309685](https://doi.org/10.1109/ACCESS.2021.309685).
- [32] W. A. H. W. Azizan, A. A. A. Rahim, S. L. M. Hassan, I. S. A. Halim, and N. E. Abdullah, "A comparative study of two machine learning algorithms for heart disease prediction system," in *Proc. IEEE 12th Control Syst. Graduate Res. Colloq. (ICSGRC)*, Shah Alam, Malaysia, Aug. 2021, pp. 132–137, doi: [10.1109/ICSGRC5186.2021.9515250](https://doi.org/10.1109/ICSGRC5186.2021.9515250).
- [33] Y. Li, A. Kazameini, Y. Mehta, and E. Cambria, "Multitask learning for emotion and personality detection," 2021, *arXiv:2101.02346*.
- [34] All things Tokenizer. *Text Preprocessing*. Accessed: Sep. 6, 2019. [Online]. Available: <https://keras.io/preprocessing/text/>
- [35] M. Gjurković and J. Šnajder, "Reddit: A gold mine for personality prediction," in *Proc. 2nd Workshop Comput. Modeling People's Opinions, Personality, Emotions Social Media*, New Orleans, LO, USA, 2018, pp. 87–97, doi: [10.18653/v1/W18-1112](https://doi.org/10.18653/v1/W18-1112).
- [36] All things embeddings. *Embedding*. Accessed: Jul. 21, 2019. [Online]. Available: <https://keras.io/layers/embeddings/>
- [37] A. Hassan and A. Mahmood, "Convolutional recurrent deep learning model for sentence classification," *IEEE Access*, vol. 6, pp. 13949–13957, 2018.
- [38] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Imag.*, vol. 9, pp. 611–629, Aug. 2018.
- [39] G. Jain, M. Sharma, and B. Agarwal, "Spam detection in social media using convolutional and long short term memory neural network," *Ann. Math. Artif. Intell.*, vol. 85, no. 1, pp. 21–44, Jan. 2019.
- [40] X. Zhao, C. Wang, Z. Yang, Y. Zhang, and X. Yuan, "Online news emotion prediction with bidirectional LSTM," in *Proc. Int. Conf. Web-Age Inf. Manage.* Cham, Switzerland: Springer, Jun. 2016, pp. 238–250.
- [41] M. A. Rahman and E. K. Dey, "Aspect extraction from Bangla reviews using convolutional neural network," in *Proc. Joint 7th Int. Conf. Informat., Electron. Vis. (ICIEV)*, Jun. 2018, pp. 262–267.
- [42] Y. Mehta, N. Majumder, A. Gelbukh, and E. Cambria, "Recent trends in deep learning based personality detection," *Artif. Intell. Rev.*, vol. 53, no. 4, pp. 2313–2339, Apr. 2020, doi: [10.1007/s10462-019-09770-z](https://doi.org/10.1007/s10462-019-09770-z).
- [43] L. Akhtyamova, M. Alexandrov, and J. Cardiff, "Adverse drug extraction in Twitter data using convolutional neural network," in *Proc. 28th Int. Workshop Database Expert Syst. Appl. (DEXA)*, Aug. 2017, pp. 88–92.
- [44] L. C. Chen, C. M. Lee, and M. Y. Chen, "Exploration of social media for sentiment analysis using deep learning," *Soft Comput.*, vol. 24, no. 11, pp. 8187–8197, 2019.
- [45] C. Olah. (2015). *Understanding LSTM Networks*. [Online]. Available: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

- [46] F. An and Z. Liu, "Facial expression recognition algorithm based on parameter adaptive initialization of CNN and LSTM," *Vis. Comput.*, vol. 36, no. 3, pp. 483–498, 2019.
- [47] M. Gjurković, M. Karan, I. Vukojević, M. Boánjak, and J. Snajder, "PANDORA talks: Personality and demographics on reddit," in *Proc. 9th Int. Workshop Natural Lang. Process. Social Media*, 2021, pp. 138–152, doi: 10.18653/v1/2021.socialnlp-1.12.
- [48] P. Novikov, L. Mararitsa, and V. Nozdrachev, "Inferred traditional personality assessment: Are we predicting the same thing?" 2021, *arXiv:2103.09632*.
- [49] H. An and N. Moon, "Design of recommendation system for tourist spot using sentiment analysis based on CNN-LSTM," *J. Ambient Intell. Hum. Comput.*, vol. 5, pp. 1–11, Oct. 2019.
- [50] A. Sengupta and A. Ghosh, "Mining social network data for predictive personality modelling by employing machine learning techniques," in *Computational Advancement in Communication Circuits and Systems*. Singapore: Springer, 2020, pp. 113–127.
- [51] E. Rudkowsky, M. Haselmayer, M. Wastian, M. Jenny, Š. Emrich, and M. Sedlmair, "More than bags of words: Sentiment analysis with word embeddings," *Commun. Methods Measures*, vol. 12, nos. 2–3, pp. 140–157, Apr. 2018.
- [52] X. Wang, W. Jiang, and Z. Luo, "Combination of convolutional and recurrent neural network for sentiment analysis of short texts," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers*, 2018, pp. 2428–2437.
- [53] A. Ullah, M. Z. Asghar, A. Habib, S. Aleem, F. M. Kundi, and A. M. Khattak, "Optimizing the efficiency of machine learning techniques," in *Communications in Computer and Information Science*. Singapore: Springer, 2020, pp. 553–567.
- [54] Y. Zhang, H. Yuan, J. Wang, and X. Zhang, "YNU-HPCC at EmoInt-2017: Using a CNN-LSTM model for sentiment intensity prediction," in *Proc. 8th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Analysis*, 2017, pp. 200–204.
- [55] X. Ding, T. Liu, J. Duan, and J.-Y. Nie, "Mining user consumption intention from social media using domain adaptive convolutional neural network," in *Proc. AAAI*, vol. 15, 2015, pp. 2389–2395.
- [56] Q. H. Vo, H. T. Nguyen, B. Le, and M. L. Nguyen, "Multi-channel LSTM-CNN model for Vietnamese sentiment analysis," in *Proc. 9th Int. Conf. Knowl. Syst. Eng.*, Oct. 2017, pp. 24–29.
- [57] R. Bhatia. (2017). *Understanding the Difference Between Deep Learning & Machine Learning*. [Online]. Available: <https://analyticsindiamag.com/understanding-difference-deep-learning-machine-learning/>
- [58] J. Brownlee. (2019). *What Word Embeddings Text*. [Online]. Available: <https://machinelearningmastery.com/what-are-word-embeddings/>
- [59] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM neural network for text classification," 2015, *arXiv:1511.08630*.
- [60] M. Rhanoui, M. Mikram, S. Yousfai, and S. Barzali, "A CNN-BiLSTM model for document-level sentiment analysis," *Mach. Learn. Knowl. Extraction*, vol. 1, no. 3, pp. 832–847, 2019.
- [61] M. Nguyen. (2018). *Illustrated Guide to Recurrent Neural Networks*. [Online]. Available: <https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9>
- [62] A. Iqbal and F. Siddiqui, "Predicting personality using deep learning techniques," in *Proc. Int. Conf. Intell. Data Commun. Technol. Internet Things*. Cham, Switzerland: Springer, Aug. 2018, pp. 168–179.
- [63] R. K. Hernandez and L. Scott, "Predicting Myers-Briggs type indicator with text," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017, pp. 1–9.
- [64] N. Chen and P. Wang, "Advanced combined LSTM-CNN model for Twitter sentiment analysis," in *Proc. 5th IEEE Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Nov. 2018, pp. 684–687.
- [65] A. S. Khan, H. Ahmad, M. Zubair, F. Khan, A. Arif, and H. Ali, "Personality classification from online text using machine learning approach," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 3, pp. 1–5, 2020, doi: 10.14569/IJACSA.2020.0110358.



MUHAMMAD USAMA ASGHAR received the B.S. degree (Hons.) (four-years) in computer science from ICIT, GU, Dera Ismail Khan, Pakistan, where he is currently pursuing the M.S. degree in computer science (two-years) research. His research interests include text mining and social media analytics with emphasis on psychological aspects of online community. He is a freelance Software Developer in data sciences. His recent publications are listed as follows: 1) An efficient supervised machine learning technique for forecasting stock market trends, URL: https://link.springer.com/chapter/10.1007/978-3-030-75123-4_7; 2) Automatic detection of citrus fruit and leaves diseases using deep neural network mode, URL: <https://ieeexplore.ieee.org/abstract/document/9481921/authors#authors>; and 3) Fake review classification using supervised machine learning, URL: https://link.springer.com/chapter/10.1007/978-3-030-68799-1_19.



MUHAMMAD ZUBAIR ASGHAR is an HEC approved Supervisor recognized by Higher Education Commission (HEC), Pakistan. His Ph.D. research includes recent issues in opinion mining and sentiment analysis, computational linguistics, and natural language processing. His publications include more than 50 articles in journals of international repute (JCR and ISI indexed) and having more than 20 years of university teaching and laboratory experience in social computing, text mining, computational linguistics, opinion mining, and sentiment analysis. Furthermore, he is also acted as the Special Session Chair (Social Computing) of BESC 2018 International Conference (Taiwan) and a lead guest editor of special issues. He is currently acting as a Reviewer and Academic Editor of different top-tier journals, such as IEEE ACCESS and *PLOS ONE*.



AURANGZEB KHAN is currently working as an Associate Professor with the Department of Computer Science, University of Science & Technology Bannu. His research interests include big data, data science, machine learning, and web mining. His current projects include: 1) intelligent judicial decision support system based on case and reported judgments; 2) RIFT: Rule induction framework for twitter sentiment analysis; and 3) sentiment analysis-based intelligent system for e-governance.



AMIR H. MOSAVI has highly contributed to the research on climate change risk reduction by coining the term "predictive-decision model." His commercial data-driven computation platform provides the state-of-the-art consultations to policy makers, worldwide, for making informed decisions. Through the accurate anticipation of the consequences of the potential decisions a "predictive-decision model" suggests wiser decisions in an automated manner. Having the advanced tools for processing the potential decisions in the climate change realm is, in fact, vital as often the environmentally-friendly choices contradict with the economic and political interests of the nations. In this context, his idea is to support organizations and governments through introducing novel business models, which are sustainable and can maintain the profitability. His novel concept has been particularly found to be suitable to address the sustainability issues of the UNESCO's Biosphere reserve.



HUSSAIN AHMAD received the Ph.D. degree in computer science from ICIT, Gomal University, Dera Ismail Khan, Pakistan, under the supervision of M. Z. Asghar. His research interests include computational intelligence, social computing, and text mining.