



Personality Assessment Based on Natural Stream of Thoughts Empowered with Machine Learning

Mohammed Salahat¹, Liaqat Ali¹, Taher M. Ghazal^{2,3,*} and Haitham M. Alzoubi⁴

¹College of Engineering and Technology, University of Science and Technology of Fujairah, Fujairah, UAE

²School of Information Technology, Skyline University College, University City Sharjah, 1797, UAE

³Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), 43600, Malaysia

⁴School of Business, Skyline University College, University City Sharjah, 1797, Sharjah, UAE

*Corresponding Author: Taher M. Ghazal. Email: ghazal1000@gmail.com

Received: 14 September 2022; Accepted: 03 March 2023; Published: 09 June 2023

Abstract: Knowing each other is obligatory in a multi-agent collaborative environment. Collaborators may develop the desired know-how of each other in various aspects such as habits, job roles, status, and behaviors. Among different distinguishing characteristics related to a person, personality traits are an effective predictive tool for an individual's behavioral pattern. It has been observed that when people are asked to share their details through questionnaires, they intentionally or unintentionally become biased. They knowingly or unknowingly provide enough information in much-unbiased comportment in open writing about themselves. Such writings can effectively assess an individual's personality traits that may yield enormous possibilities for applications such as forensic departments, job interviews, mental health diagnoses, etc. Stream of consciousness, collected by James Pennbaker and Laura King, is one such way of writing, referring to a narrative technique where the emotions and thoughts of the writer are presented in a way that brings the reader to the fluid through the mental states of the narrator. Moreover, computationally, various attempts have been made in an individual's personality traits assessment through deep learning algorithms; however, the effectiveness and reliability of results vary with varying word embedding techniques. This article proposes an empirical approach to assessing personality by applying convolutional networks to text documents. Bidirectional Encoder Representations from Transformers (BERT) word embedding technique is used for word vector generation to enhance the contextual meanings.

Keywords: Personality traits; convolutional neural network; deep learning; word embedding



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

An individual's personality combines his thought patterns, behaviors, and emotions. One's life standard, health, choices, social impacts, and several other factors are immensely influenced by personality type. Assessing one's personality factors is found to have various practical applications in many fields, such as through sentiment analysis in service and product recommendation systems [1,2], and in sentiment lexicon for word polarity disambiguation [3] since different people can be represented by different polarities of the same concept, reducing the circle of suspects in forensic is also facilitated when a prior judgment of personality is performed. In human resource management, personality traits assessment helps select suitable candidates. An individual's personality can be assessed in various ways; Gordon Allport claims that individuals' personalities can be differentiated and understood by differentiating their personality traits. These personality traits provide the basic dimensions upon which individuals differ [4]. Traits psychology, on the other hand, depicts a limited number of these dimensions, including Extraversion, Conscientiousness, or Agreeableness. Every individual is a blend of the varying extent of these dimensions or traits.

The Big Five Model, also known as the Five-Factor Model of personality, is the most widely accepted in human personality psychology. Primarily, it was developed by various independent research groups. Conversely, it was advanced by Ernest Types and Raymond Christal [5]; Digman performed additional progressions [6], and Lewis Goldberg later finalized it [7]. Previous research has been conducted on the automation of personality detection from plain text. James Pennebaker and Laura King [8] conducted a study where they collected stream-of-consciousness essays from volunteers, and then asked the authors to define their own Big Five personality traits. The essays were analyzed using Linguistic Inquiry, and Word Count (LIWC) features to determine the correlation between the essay and personality. François Mairesse and colleagues further expanded on this work by using additional features, such as imageability, to enhance the performance of the model. author performed a comprehensive study on the essay dataset and the My Personality Facebook status dataset. They used different combinations of feature sets to outperform Mairesse's results, which they referred to as the Mairesse baseline.

Fei Liu and colleagues more recently developed a language-independent and compositional model for personality trait recognition, specifically for short tweets. In a different vein, researchers have also successfully applied deep convolutional networks for tasks related to sentiment analysis. The use of LIWC features has been a common approach in previous studies for personality detection from plain text and has proven to be effective in many cases. However, researchers have continually sought to improve the performance of models for this task, whether using additional features, or through the application of new methods such as deep convolutional networks. It is worth noting that the essay dataset used in Pennebaker and King's study has been a commonly used dataset for research in this field. The continued refinement of methods and models for personality detection from plain text, as well as the use of new data sources such as social media, has resulted in increased accuracy and effectiveness of these models over time. Typically, a formal description of personality can be provided in terms of Big Five personality traits, [9] with varying extent of the values for the traits such as:

Openness (OPN) gives one's creative and inquiring character against dogmatism and cautiousness. Conscientiousness (CON) measures one's organizational ability and efficiency versus messiness and carelessness. Extroversion (EXT) Measures the extent to which an individual is talkative and outgoing. It also measures the energy against reserve and solitary. Agreeableness (AGR) relates to an individual's trustworthiness, generosity, specific behavior, and modesty. These characteristics are measured against unreliability, intricacy, meagerness, and boastfulness. Neuroticism (NEU) measures sensitivity and

nervousness against a person's confidence. Individuals' behaviors and characteristics are categorized under these groups; thus, every person has a distinct personality due to varying levels of these characteristics. The measurement of these characteristics gives a blueprint of an individual's personality. There are several ways to measure these characteristics; texts and writings have been found to interpret various aspects and levels of personality characteristics to greater extents.

Textual data has increasingly been disseminated daily; therefore, its usability and serviceability are adopted in various aspects. Natural Language Processing (NLP) is a technique that studies the classification of textual data such as website news, biographies, comments, summaries, microblogs, e-mails, essay databases, and different forms of electronic data. These textual means help classify and analyze the properties of sources they belong to. Hence the text documents can be utilized to perform classification and summarization tasks on the user-generated texts. Short-length texts, user comments, and question sires partially present one's characteristics. It is observed that open questions and detailed bibliography give more accuracy when depicting oneself and provide richer and more valuable information. The usability of open answers and essay documents is more practical to the ones who stay online either more frequently or least frequently [10].

This research article presents a way to extract the big five personality traits through stream-of-consciousness essays through a Convolutional Neural Network (CNN). Stream-of-consciousness is a style or technique of essay writing that goes to apprehend the natural flow of a person's extended thinking process. It often incorporates his sensory impressions, partial ideas, unusual syntax, and rough grammar. The stream-of-consciousness essays are used to train five different networks, one for each personality trait consisting of the same architecture. Each convolution neural network can predict the level to which an individual possesses a particular personality trait.

This research proposed a novel document modeling approach using CNN features extractor. The model involves feeding sentences from essays into convolution filters to obtain sentence models in the form of n-gram feature vectors. The essay is then aggregated based on its sentence vectors and combined with Mairesse features [11] extracted from the texts. To improve the results, emotionally neutral sentences were discarded from the essays. The final classification was performed using a fully connected neural network consisting of one hidden layer. The results showed that this approach outperformed the current state of the art in terms of all five personality traits.

Deep learning is a subfield of machine learning, also known as hierarchical learning, deep machine learning, and deep structured learning. It involves the use of neural networks, where one set of neurons receives input signals and the other sends output signals. Deep learning models have been successfully used in various tasks such as computer vision, speech recognition, automatic handwriting generation, and natural language processing. [12].

The use of deep neural networks in social media scenarios is crucial due to their ability to automatically extract both local and global features and identify misinformation [13]. Deep learning models are particularly effective in detecting personality traits due to their capacity for learning [14].

The proposed model in this paper uses CNN for deep learning. The embedding layer transforms the input into word embedding representations, which are then fed into the CNN layer to create a feature vector. The output from the CNN layer is then passed to a dense layer with a sigmoid activation function, which labels the text according to personality traits. The first layer, the embedding layer, maps words or sentences in natural language to a vector format that can be processed by a machine.

This approach, called word embedding, was proposed by Hinton and aims to map words to a lower-dimensional space to overcome the issue of vector sparsity and better capture semantic links among words. The second step involves the classification layers, where the representation vectors from the first stage are input into the classification part. Convolutional filters in the CNN layer capture local dependencies, while the maximum pooling layer reduces the dimensionality and prevents overfitting. The final output labels the text as personality traits.

2 Literature Review

The use of Natural Language Processing (NLP) techniques and Machine Learning (ML) algorithms have become a popular method for analyzing and classifying text data. Text mining plays a crucial role in preprocessing, summarizing, classifying, and drawing inferences from text data [15]. The process of extracting electronic data from various sources, classifying them, and integrating them is crucial for the research area. In recent years, there has been a growing interest in personality analysis, as accurate personality analysis has the potential to benefit many areas. Experts in the field often use social networks and social media sites in various languages to perform personality analysis. The results of personality analysis have applications in fields such as recommendation systems, emotion analysis, and crime fighting. Furthermore, human resource departments are now using social media accounts to make hiring decisions, without the need for in-person interviews.

Therefore, the use of NLP and ML algorithms in the area of personality analysis is crucial for understanding and improving the human experience. [16].

Personality analysis is a growing field that has various applications. Social media sites and mobile networks are used by experts to analyze personalities in different languages. The results of these analyses can be used in recommendation systems, emotion analysis, crime fighting, and even hiring decisions. E-commerce companies use personality analysis to provide customers with more attractive products, and health authorities use it to study the relationship between personality traits and specific illnesses. One study even investigated the relationship between personality traits and software quality and found that personality analysis can be used to predict software developers' performance. Another study proposed a heterogeneous information ensemble framework to predict user personality by integrating different types of information such as emoticons, avatars, and self-language usage. [17,18].

The users, and their answers were analyzed using the ontology-based approach. The results showed that the proposed method is effective in detecting personality traits and can be used in various fields such as human-computer interaction and mental health diagnosis [19].

These studies show the importance and potential of using NLP and ML algorithms for personality analysis. The ability to extract personality traits from texts can provide valuable insights for various fields and have practical applications. However, it is crucial to note that these results should be used with caution and should not be used as the sole basis for making decisions about individuals. Further research is needed to improve the accuracy of these methods and to address ethical and privacy concerns [20,21].

In conclusion, there has been a growing interest in using NLP and machine learning techniques to analyze personality traits from text data. This research has applications in various areas such as recommendation systems, emotion analysis, health, software development, and legal and public sectors. Researchers have proposed various models, including the CNN deep learning model and the Glove Model, to predict the Big-Five personality traits. These models have shown promising results, with the proposed model achieving an average correlation of 0.33 over the Big-Five traits, which is

better than the baseline method. The findings of these studies are based on English Twitter data, and the research may be extended to other languages [22,23].

In conclusion, the use of NLP and machine learning algorithms for personality analysis has been extensively studied in recent years. The results of these studies indicate that it is possible to predict an individual's personality traits based on the text they produce. Deep learning models such as CNN and LSTM have shown promising results in the classification of personality traits. The research also shows that by integrating heterogeneous information such as emoticons, avatars, and self-language usage, the accuracy of the personality predictions can be further improved. Additionally, the use of word embeddings such as GloVe and LIWC has been shown to be effective in predicting personality traits from user tweets. The results of these studies have a wide range of applications, from e-commerce to human resources, and even in areas such as crime fighting and health. However, further research is required to improve the accuracy of the predictions and extend the findings to other languages. [24,25,13].

3 Methodology

It has been seen that the text can reflect various aspects of the speaker or the author. The dissertation utilizes this module to extract the 5 personality factors from the essays on the stream of consciousness [25].

Stream-of-consciousness essays give the written equivalent of one's thought process as a narrative device that may be provided as an interior monologue or a connection to an individual's actions. Usually, stream-of-consciousness writing can be termed a special interior monologue characterized by associated lead and lack of some or all punctuations and thought processes.

Personality traits are reflected by the selection of the word's humans do. In this research work, personality traits have been detected through the sentences used by a human being. The method utilizes Stream of consciousness essays fed to train one Convolutional Neural Network (CNN) for each trait. Each personality trait is learned by one convolutional network; all convolutional neural networks have similar architecture. Each neural network is a binary classifier that protects one personality trait from being positive or negative. Since personality traits are classified and scribed concerning the big five personality factors, the current personality detection module depicts personality traits in a probability distribution over the output layer of SoftMax, representing the extent to which a particular trait exists in humans.

Feature extraction, this dissertation was performed by implementing a CNN as a document modeling technique [23]. Convolution filters were fed with sentences from the essays, and sentence models were obtained as feature vectors of size n-gram. Every essay gets a representation through its aggregated sentences. Mairesse features are then used to concatenate these sentences [11], which were extracted at preprocessing stage directly from the text; this method improves the model's performance. Emotionally neutral input sentences were discarded from input essays since they contribute to no diversion or polarity to personality trait assessment. The final classification is performed by feeding these document vectors to the neural network that is fully connected and has one hidden layer. The SoftMax layer SoftMax layer obtains the output, and the probability distribution is received for each personality trait.

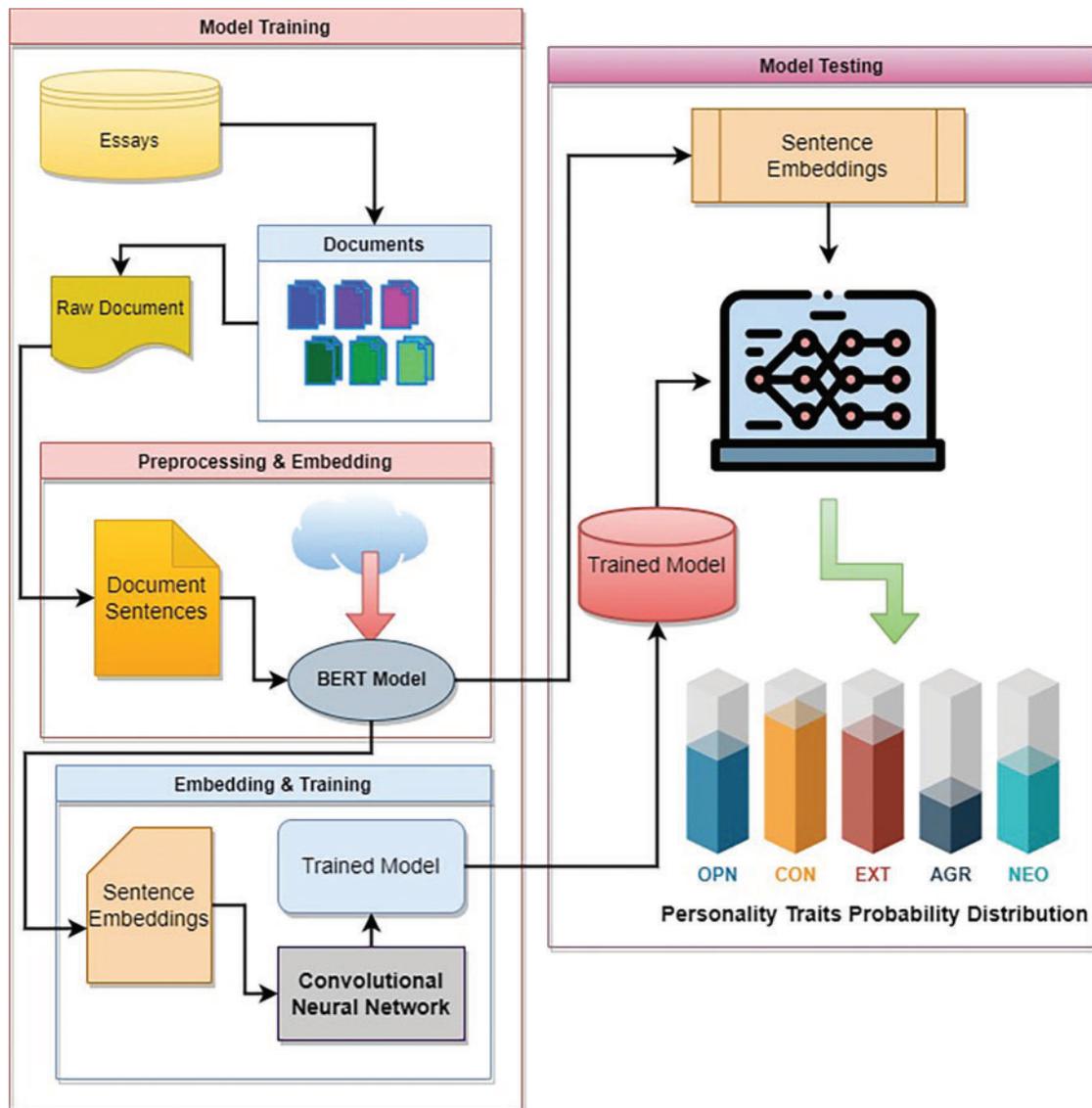


Figure 1: Proposed model of a personality prediction

The steps included in the proposed method included data processing, filtering, feature extraction, and trait classification Fig. 1. Two kinds of features are used: document-level stylistic features, fixed in the number of semantic features. These features are combined in two variable-length representations for the text input. Convolutional neural networks are then fed by these variables and representations and are processed hierarchically through the combination of words into n-gram, n-gram into sentences, and finally, the sentences into documents. The obtained documents are combined with stylistic features at the document level to obtain document representation for final classification. The complete methodology adopted for personality trait detection is divided into the following steps:

Preprocessing

At the preprocessing stage, the sentences are splitted; data is cleaned and unified; converting uppercase letters to lowercase is performed and represented as vectors of real numbers.

Feature extraction at the document level

Document-level features are extracted Mairesse baseline features, which include global features, the total number of words and the average sentence length.

Filtering

In essays, a few sentences may appear that may not possess clues for personality traits. In semantic feature extraction, such sentences may be ignored; they appear as noise and affect the performance of classifiers. Removing these sentences will also considerably reduce the overall input, thereby improving the classifiers' performance without negatively affecting the resultants. Therefore, such neutral sentences are removed during filtering.

Feature extraction at word level

Feature extraction at Word Level is performed by representing word embedding in continuous spectra space. The task is performed with BERT embedding, and for each document, a feature set of varying lengths is obtained. Then each document is represented with a different number of sentences, and every sentence is defined as fixed-length word feature vectors with a variable number of words.

Classification

Classification is performed using deep convolutional neural networks. The text is processed hierarchically in the initial layers of CNN. Each word gets a representation as a fixed-length vector in the input through Google's pre-trained BERT model, whereas sentences have a variable number of word vector representations. In an intermediate layer of CNN, this variable-length vector of the sentence is reduced to the vector with a fixed length. The sentence vector is an embedded sentence in continuous vector space. Documents get a presentation as a variable number of fixed-length sentence embedding. This variable length document is reduced to document vectors with fixed length at a deeper layer of CNN. After this fixed and document vector is obtained by concatenating feature vectors with fixed length with the features at the document level, these fixed length document vectors are utilized in the final classification. Aggregation of word vectors into sentence vectors is performed by convolution to generate n-gram features.

3.1 Network Architecture

The classifier's network is depicted in [Fig. 2](#) and comprises seven layers; input layer for word vectorization, convolution for sentence vectorization, max pooling for sentence vectorization, 1-max pooling for document vectorization, concatenation for document vectorization, linear for classification using sigmoid activation, and two neurons SoftMax output to obtain probability distribution for the corresponding trait. These layers and work have been discussed in detail in the proceeding sections.

The method includes data input preprocessing, filtering, feature extraction, and classification. Two kinds of features have been used; document-level stylistic features that are fixed in number and semantic features for each word; these two features are then combined into a variable-length representation for input text is then fed by these variable-length representations; these sentences are processed by merging words into n-gram, n-gram into sentences and finally the sentences to complete documents.

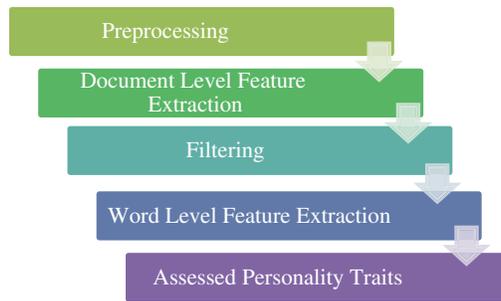


Figure 2: Steps performed during sentence preprocessing as 768×1 vectors

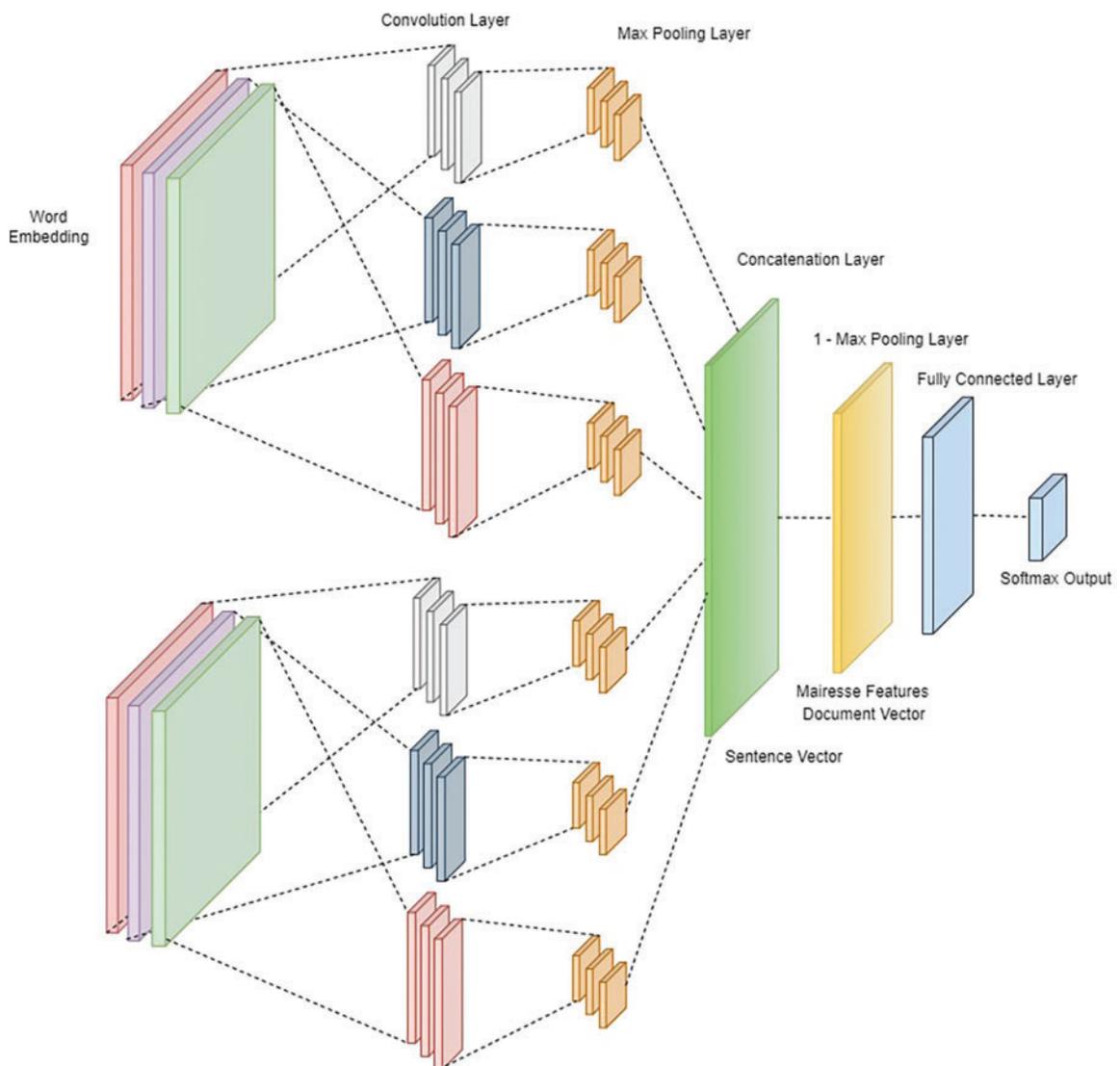


Figure 3: Network architecture of the proposed model

During feature extraction, the module will consider two types of features: stylistic features at the document level and pre-word semantics at the word level. It will then combine these features to form them into a variable-length representation of input text.

This variable-length feature vector will then be sent to a neural network to extract features hierarchically by combining words into n-gram strings, n-gram strings into sentences, and the sentences into a complete document. The values obtained will be combined with the stylistic features at the document level and construct document-level representation for the final probability distribution for a particular trait. The method, therefore, comprises the following steps:

3.2 Preprocessing

The documents are represented as vectors of real numbers before feeding them into the networks. Language representation may sometimes be used as context-free and other times as contextual language. The problem in context-free languages like word2vec and Glove is that the same word has the same representation when used in different contexts. When using a contextual language such as BERT the language is represented as a token based on sentences where they are used, thereby providing a better understanding of language. The current work uses a pre-trained model BERT for language modeling. This pre-trained model is fine tuned on a supervised dataset used for personality assessment through deep bidirectional layers of encoders. This step has been brought through the BERT preprocessor, whose inputs are raw sentences of the documents and output are encoded vectors with 768 dimensions.

3.3 Document-level Feature Extraction

At the document level features extraction, Mairesse baseline features [11], such as word count and average sentence length, are utilized.

3.4 Filtering

Certain sentences might not contain any clues of personality factors; therefore, in semantic feature extraction, these sentences need to be ignored. There are reasons for doing this: firstly, their present noise can downgrade the classifier performance, secondly, removal of such sentences will reduce the size of input documents and the training time and does not affect them negatively. Therefore, these sentences are removed before further steps.

3.5 Word-level Feature Extraction

In continuous vector space, word embedding through BERT uses individual representations for each word. The variable-length feature set for documents is obtained by doing this; these documents are represented as the variable number of sentences; the sentences are epitomized as the variable number of feature vectors with fixed-length words. The input with BERT for one, two, and three sentences with special beginning and end tokens are given as:

Each word in the input string is signified as a feature vector and is provided to the artificial neural network. A learning algorithm is expected to extract unigram, bigram, and trigram features from each sentence. The sentence vector will be a concatenation of feature vectors. These feature vectors are acquired from these algorithms after max pooling. François Mairesse has developed a feature set at the document level for personality detection. This feature set consists of 84 features. It consisted of Linguistic Inquiry and Word Count Features, utterance-type features, Medical Research Council features, and prosodic features. These 84 features are then made to concatenate with the document

vector. The final output is aggregated sigmoid activation and a softmax layer of size one to determine the probability of particular trait intensity.

3.6 Classification

CNN performs classification, whereas the early layers process the text hierarchically, and using BERT, the representation of each word is fixed. The feature vector sentences are represented as word vectors of variable size. A variable length factor is used to fix and factor each sentence at some layer of CNN, serving as a sentence embedding method in continuous input vector space. Documents are constructed with sentence embedding of fixed length, which are different in number for each document. The document with variable length is reduced to fixed-length document vectors at a deeper layer. The features at the document level are concatenated with the fixed-length feature vectors, thereby giving fixed-length document vectors. These fixed-length document vectors are used for final classification.

Word vectors are aggregated into sentence vectors using convolution to construct n-gram features. During aggregation of the sentence, vectors construct document vectors where convolution is not used.

Input

The data set consists of a set of documents where:

“d”: each document is denoted based on the sequence of sentences.

“ s_i ”: sequence of sentences a sequence of words.

“ w_j ”: each word as a real-valued vector known as word embedding whose length is fixed.

In the current implementation, BERT, pre-trained by Google, is used for embedding. Let x be the input to the convolution neural network,

$x = \mathbf{R}^{D \times S \times W \times D}$ Having four dimensions and

$D = \{d_1, d_2, d_3, \dots, d_i\}$ is the number of documents in the data set,

$S = \{s_1, s_2, s_3, \dots, s_i\}$ is the maximum number of sentences represented by all documents.

$W = \{w_1, w_2, w_3, \dots, w_j\}$ be the highest word count in a sentence in all the documents and

E represents the size of word embedding (768×1).

During implementation, shorter documents are padded with dummy sentences, and sentences with shorter lengths are padded with dummy words. Three convolutional filters extract each sentence’s unigram, bigram, and trigram features. The feature vectors obtained by the convolution filters are concatenated at the max pooling layer.

3.7 Word Vectors Aggregation to Sentence Vectors

3.7.1 Convolution Layer

Convolutional filters are applied to extract n-gram features, each with size $n \times E$ for each sentence $\mathbf{R}^{W \times E}$. 200 n-gram feature maps for unigram, bigram, and trigram are used. $F_n^{\text{conv}} \in \mathbf{R}^{200 \times n \times E}$ be the convolution filters applied on matrix s for each n . A bias $B_n^{\text{conv}} \in \mathbf{R}^{200}$ is added to the output of the filter to provide three feature maps $FM_n \in \mathbf{R}^{200 \times (W-n+1) \times 1}$, where $n = 1, 2, 3$. Rectified Linear Unit (ReLU) function is applied to the feature maps FM_n to introduce non-linearity.

3.7.2 Max Pooling

To each feature map FM_n , max pooling is applied to downsample the feature map to $DFM_n \in \mathbb{R}^{200 \times 1 \times 1}$, which is flattened to obtain a feature vector of size 200.

3.7.3 Convolution

The type of n-gram vectors is concatenated to obtain vector $s \in \mathbb{R}^{600}$ for sentence representation. In each sentence in the document, convolution, and max pooling are applied. Network parameters are shared between all sentences in the document. Since all sentences fairly dress dummy words to have a common size there is no need to pad documents with dummy sentences for the same.

Vector to Document Vector Aggregation

After individual sentence processing, the document vector becomes a concatenation of all its sentence vectors with variable sizes. Each sentence is a 600-dimensional vector. The sentence vector of maximum size is selected from the document vectors obtained for each feature. This resulted in a vector with 600 dimensions having all real values $d^{\text{network}} \in \mathbb{R}^{600}$ for the whole document.

Document-Level Feature addition to Document Vector

François Mairesse features at document level sets consisting of 84 features to detect personality factors, including linguistic inquiry word count features, Medical Research Council features, prosodic features, and utterance type features. Features included in the set are word count and the average number of words in a sentence, the total number of pronouns present, past, and future tense words, phonemes, syllables, interrogations, and declarations in the document. When these 84 features are concatenated with the document vector d^{network} we obtained a 684-dimensional vector of document

$$d = (d^{\text{network}}, d^{\text{Mairesse}}) \in \mathbb{R}^{684} \quad (1)$$

Classification

Final classification is performed through a two-layer perceptron that consists of a fully connected layer. This layer has a size of 200 and a SoftMax layer with a size of 2, showing the presence of the trait “yes” and the absence of the trait “no”.

The document $d \in \mathbb{R}^{684}$ is multiplied by a matrix $W^{\text{fc}} \in \mathbb{R}^{684 \times 200}$ and a bias $B^{\text{fc}} \in \mathbb{R}^{200}$ is added to obtain the vector $d^{\text{fc}} \in \mathbb{R}^{200}$. Non-linearity with sigmoid activation is introduced to improve the results in Eq. (2),

$$d^{\text{fc}} \in \sigma(dW^{\text{fc}} + B^{\text{fc}}) \quad (2)$$

where

$$\sigma(x) = \frac{1}{(1 + \exp(-x))} \quad (3)$$

SoftMax Output

The probability distribution of document belonging to the extent of being fully present or fully absent, SoftMax function is used Eq. (3),

$$x_{\text{yes}}, x_{\text{no}} = d^{\text{fc}} W^{\text{sm}} + B^{\text{sm}} \quad (4)$$

where $W^{sm} \in \mathbb{R}^{200 \times 2}$ and the bias $B^{sm} \in \mathbb{R}^2$, the class probabilities are calculated as.

$$p(i|networkparameters) = \frac{e^{x_i}}{e^{x_{yes}} + e^{x_{no}}} \quad (5)$$

Suppose for an n-dimensional real vector space \mathbb{R}^d , $\delta: \mathbb{R}^d \Rightarrow \mathbb{R}^d$ (d represents the class probabilities and is 2 in this case) be a softmax function defined by

$$\delta(x_i) = \frac{e^{x_i}}{\sum_{j=1}^d e^{x_j}}; \text{ For } j = 2 \quad (5a)$$

where $x = x_1, x_2, \dots, x_d$ be a vector in \mathbb{R} . The class probability is calculated as [Eq. \(6\)](#)

$$p_i = \frac{e^{x_i}}{e^{x_{yes}} + e^{x_{no}}} \quad (6)$$

To normalize the SoftMax function, a factor k is introduced, and the normalized form of the SoftMax function becomes in [Eqs. \(7\) and \(8\)](#),

$$p_i = \frac{ke^{x_i}}{k(e^{x_{yes}} + e^{x_{no}})} \quad (7)$$

or

$$p_i = \frac{e^{x_i + \log(k)}}{k(e^{x_{yes} + \log(k)} + e^{x_{no} + \log(k)})} \quad (8)$$

We can choose an arbitrary value for $\log(k)$

$$\log(k) = -\max(x)$$

Derivative of the SoftMax function

In the SoftMax function is defined by $p_i = \frac{e^{x_i}}{e^{x_{yes}} + e^{x_{no}}}$, if the output of probability distribution is x_m then the same function can also be represented as [Eqs. \(9\) and \(10\)](#).

$$\frac{\partial p_i}{\partial x_m} = \frac{\partial}{\partial x_m} \left(\frac{e^{x_i}}{e^{x_{yes}} + e^{x_{no}}} \right) \quad (9)$$

$$\frac{\partial p_i}{\partial x_m} = \frac{(e^{x_{yes}} + e^{x_{no}}) \frac{\partial}{\partial x_m} (e^{x_i}) - e^{x_i} \frac{\partial}{\partial x_m} (e^{x_{yes}} + e^{x_{no}})}{(e^{x_{yes}} + e^{x_{no}})^2} \quad (10)$$

Using the quotient rule of derivative, the above equation can be written as,

When $i = m$ the probability change rate will be nonzero, whereas zero otherwise in [Eqs. \(11\)–\(17\)](#).

$$\frac{\partial p_i}{\partial x_m} = \frac{(e^{x_i} (e^{x_{yes}} + e^{x_{no}})) - e^{x_m} e^{x_i}}{(e^{x_{yes}} + e^{x_{no}})^2} \quad (11)$$

$$\frac{\partial p_i}{\partial x_m} = \frac{e^{x_i} ((e^{x_{yes}} + e^{x_{no}}) - e^{x_m})}{((e^{x_{yes}} + e^{x_{no}}))^2} \quad (12)$$

$$\frac{\partial p_i}{\partial x_m} = \frac{e^{x_i}}{(e^{x_{yes}} + e^{x_{no}})} \times \frac{(e^{x_{yes}} + e^{x_{no}}) - e^{x_m}}{(e^{x_{yes}} + e^{x_{no}})} = p_i - p_i p_m \quad (13)$$

$$\frac{\partial}{\partial x_m} \left(\frac{e^{x_i}}{(e^{x_{yes}} + e^{x_{no}})} \right) = p_i (1 - p_m) \quad (14)$$

For $i \neq m$, we have

$$\frac{\partial}{\partial x_m} \left(\frac{e^{x_i}}{(e^{x_{yes}} + e^{x_{no}})} \right) = \frac{((e^{x_{yes}} + e^{x_{no}})) \frac{\partial}{\partial x_m} (e^{x_i}) - e^{x_i} \frac{\partial}{\partial x_m} ((e^{x_{yes}} + e^{x_{no}}))}{((e^{x_{yes}} + e^{x_{no}}))^2} \quad (15)$$

$$\frac{\partial}{\partial x_m} \left(\frac{e^{x_i}}{(e^{x_{yes}} + e^{x_{no}})} \right) = \frac{0 - e^{x_i} e^{x_m}}{((e^{x_{yes}} + e^{x_{no}}))^2} = \frac{-e^{x_i} e^{x_m}}{((e^{x_{yes}} + e^{x_{no}}))^2} \quad (16)$$

$$\frac{\partial}{\partial x_m} \left(\frac{e^{x_i}}{(e^{x_{yes}} + e^{x_{no}})} \right) = \frac{-e^{x_m}}{(e^{x_{yes}} + e^{x_{no}})} \times \frac{e^{x_i}}{(e^{x_{yes}} + e^{x_{no}})} = -p_m p_i \quad (17)$$

Therefore, the derivative of the SoftMax function can be written as

$$\frac{\partial p_i}{\partial x_m} = \begin{cases} p_i (1 - p_m) & \text{if } i = m \\ -p_m p_i & \text{if } i \neq m \end{cases} \quad (18)$$

Eq. (5) can be represented in Kronecker delta notation as.

$$\delta_{im} = \begin{cases} 1 & \text{if } i = m \\ 0 & \text{if } i \neq m \end{cases} \quad (19)$$

Then,

$$\frac{\partial p_i}{\partial x_m} = p_i (\delta_{im} - p_m) \quad (20)$$

Derivative of cross-entropy with SoftMax in Eq. (21)

$$L = - \sum_k y_k \log(p_k) \quad (21)$$

Let L be the loss function defined in terms of cross-entropy.

Taking the derivative of Eq. (6) concerning O_k ; where O_k is output from the output (fully connected layer) in Eq. (22)–(29),

$$\frac{\partial L}{\partial O_k} = \frac{\partial}{\partial O_k} \left(- \sum_k y_k \log(p_k) \right) \quad (22)$$

$$\frac{\partial L}{\partial O_k} = - \sum_k y_k \frac{\partial}{\partial O_k} (\log(p_k)) \quad (23)$$

Applying chain rules.

$$\frac{\partial L}{\partial O_k} = - \sum_k y_k \frac{\partial (\log(p_k))}{\partial p_i} \times \frac{\partial p_i}{\partial O_k} \quad (24)$$

$$\frac{\partial L}{\partial O_k} = - \sum_k y_k \frac{1}{p_i} \times \frac{\partial p_i}{\partial O_k} \quad (25)$$

$$\frac{\partial L}{\partial O_k} = -y_k (1 - p_k) - \sum_{i \neq k} y_i \frac{1}{p_i} (-p_i p_k) \quad (26)$$

$$\frac{\partial L}{\partial O_k} = -y_k (1 - p_k) + \sum_{i \neq 1} y_i p_k \quad (27)$$

$$\frac{\partial L}{\partial O_k} = -y_k + y_k p_k + \sum_{i \neq k} y_i p_k \quad (28)$$

The following equation gives the rate of change of the current module's loss function for decreasing the loss at the SoftMax function.

$$\frac{\partial L}{\partial O_k} = p_k \left(y_k + \sum_{i \neq k} y_i \right) - y_k = p_k - y_k \quad (29)$$

4 Results and Discussions

A personality trait assessment module has assessed personality traits based on CNN. The model has been tested on a well-known dataset having typical usage in several cognitive-behavioral tests, whereas the implementation is mainly performed with python's TensorFlow library. The dataset of stream-of-consciousness essays designed and constructed by James Pennebaker and Laura King's 2,468 miscellaneous tagged with the author's personality traits; EXT, NEU, CON, and OPN, whereas the experimentation was carried out with 2,467 essays. During the experiment, 10-fold cross-validation is used to evaluate the trained network.

The pseudocode for the algorithm was implemented in the personality trait assessment module. The text was split into sentences with periods and question marks. The current work has implemented BERT for sentence embedding; therefore, the reduction of words to lowercase and the removal of characters other than ASCII letters is performed by BERT encoding. Several sentences were present without periods combined with other sentences to form longer sentences. Sentences longer than 150 words were split into sentences of 20 words each. This action resulted in the last sentence having a length shorter than others.

Mairesse features were used to extract 84 Mairesse features from every document. It is given that the relevant sentences have at least one emotionally charged word. After document-level feature extraction and before applying BERT the sentences with no emotionally charged words were discarded. For this, the NRC emotion lexicon was used to obtain emotionally charged words. 14,182 words were contained by lexicon and tagged with 10 attributes: negative, positive, Joy, fear, disgust, anger, anticipation, trust, and surprise. It was considered that a sentence would have emotional cues only when one of these words was present.

Pseudo code for the proposed model

```

Begin
//Scanning the Text
SET Text = Scan Essays
#Preprocessing
SET Sentences = Sentence_Splitter(essay)
# Providing sentences to BERT for encoding
SET SentenceVectors = BERT ()
# Split the sentences Dataset items into Test/Train
SET TestDataSet = 20%
TrainSet, TestSet = Split (SentenceVectors)
#Apply Classifiers
SET ClassifierModel = classifiers ()

```

(Continued)

Continued

```

SET ClassificationModel = Model: fit (TrainSet)
#Predictions
SET PredictionModel = Classification: predictions (TestSet)
#Accuracy
SET AccuracyModel = Accuracy (PredictionModel,TestSet)
#Performance
Measure (PrecisionModel, FMeasure)
SET Result = classificationReport (TestSet, Prediction-Model, PeronalityDist)
Return (Result)
End

```

BERT pre-trained model was used to convert each word into 768-dimensional vectors. In a word not found in BERT, 768 coordinates were assigned a value from a uniform distribution in $[-0.25, +0.25]$. Feature extraction from the documents process is performed through the scikit-learn library.

Table 1: Result comparison of proposed model

Document vector d	Embedding	Classifier	Convolution filter	EXT	NEU	AGR	CON	OPN
CNN + Mairesse [23]	Word2vec	sMLP/FC	1, 2, 3	54.61	57.81	55.84	57.30	62.13
		sMLP/MP	1, 2, 3	58.09	57.33	56.71	56.71	61.13
		MLP	1, 2, 3	55.54	58.42	55.40	56.30	62.68
		SVM	1, 2, 3	55.65	55.57	52.40	55.05	58.92
BB-SVM [26]	BERT	–	–	59.30	59.39	56.52	57.84	62.09
Proposed Model	BERT	sMLP/FC	1, 2, 3	60.75	63.5	55.3	68.8	68.5
		sMLP/MP	1, 2, 3	59.08	67.5	58.9	76.6	70.5
		MLP	1, 2, 3	54.96	58.7	54.6	71.6	75.3
		SVM	1, 2, 3	61.2	61.8	56.69	69.5	78.8

Finally, for classification, we used the neural network, a multilayer perceptron (MLP) with one hidden layer in CNN. The network with linear SVM using CNN [27–30] has been used as baseline results which have implemented a similar network scheme with word2vec for word embedding. Table 1 shows the proposed model’s accuracy comparison.

The model has been compared with [23] and [27] state-of-the-art results. The comparative analysis results have been provided in Table 1. It has been observed that the proposed model has improved performance against the previous attempts. The model has been trained on Google’s collab environment under python 3.8 and tensor flow version 2.2 and took around 2 h to complete its 30 epochs. Table 1 gives a comparison of the proposed model with the two models. The contextual model performs better in either way, and the platform selection has made the model outperform.

5 Conclusion

Behavior analysis and personality detection have become highly demanding and challenging in cognition-based modeling-based multi-agent environments. Personality detection from text is a growing trend since people socialize more in this context. Datasets based on closed questions and straightforward answers deviate more towards partiality and non-realistic expressions. This attempt has utilized stream-of-consciousness essays generated under controlled environments and limited time slots where writers were offered almost no time to deliberate. Meanwhile, word embedding has been performed in the contextual model BERT, which further raised the cognitive behavior of the model. The model is expected to incorporate into environments where human-level cognitive behavior is expected. In the future, the model will integrate emotions and habits as personality indicators along with the text.

Funding Statement: Self-funded; no external funding is involved in this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Human Linguistics Technological Processing Conf.*, Minneapolis, Minnesota, pp. 4171–4186, 2019.
- [2] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intelligent System*, vol. 31, no. 2, pp. 102–107, 2016.
- [3] S. Poria, R. Bajpai and B. Schuller, "Sentic Net 4: A semantic resource for sentiment analysis based on conceptual primitives," in *Proc. of Coling, the 26th Int. Conf. on Computational Linguistics: Technical Papers*, Osaka, Japan, pp. 2666–2677, 2016.
- [4] S. Hussain, R. A. Naqvi, S. Abbas, M. A. Khan and T. Sohail, "Trait based trustworthiness assessment in human-agent collaboration using multi-layer fuzzy inference approach," *IEEE Access*, vol. 9, pp. 73561–73574, 2020.
- [5] L. Fung and R. B. Durand, "Personality traits," *Investor Behavior: The Psychology of Financial Planning and Investing*, vol. 4, no. 2, pp. 99–115, 2014.
- [6] E. C. Tupes and R. E. Christal, "Recurrent personality factors based on trait ratings," *Journal of Personality*, vol. 60, no. 2, pp. 225–251, 2014.
- [7] D. M. Buss, "Evolutionary personality psychology," *Annual Review of Psychology*, vol. 42, no. 1, pp. 459–491, 2021.
- [8] L. R. Goldberg, "The structure of phenotypic personality traits," *American Psychologist Journal*, vol. 48, no. 1, pp. 26–34, 2016.
- [9] J. W. Pennebaker and L. A. King, "Linguistic styles: Language use as an individual difference," *Journal of Personality Sociology Psychology*, vol. 77, no. 6, pp. 1296–1312, 2015.
- [10] N. S. Elmitwally, A. Kanwal, S. Abbas and M. A. Khan, "Personality detection using context-based emotions in cognitive agents," *Computers Materials & Continua*, vol. 70, no. 3, pp. 4947–4964, 2022.
- [11] V. Ong, A. D. Rahmanto and D. Suhartono, "Exploring personality prediction from text on social media: A literature review," *Internetworking Indonesia*, vol. 9, no. 1, pp. 65–70, 2017.
- [12] F. Mairesse, M. A. Walker, M. R. Mehl and R. K. Moore, "Using linguistic cues for the automatic recognition of personality in conversation and text," *Journal of Artificial Intelligence Research*, vol. 30, no. 2, pp. 457–500, 2017.

- [13] S. Ahmad, M. Z. Asghar, F. M. Alotaibi and I. Awan, "Detection and classification of social media-based extremist affiliations using sentiment analysis techniques," *Human Centric Computing and Information Sciences*, vol. 9, no. 1, pp. 1–23, 2019.
- [14] D. Xue, L. Wu, Z. Hong, S. Guo, L. Gao *et al.*, "Deep learning-based personality recognition from text posts of online social networks," *Applied Intelligence*, vol. 48, no. 11, pp. 4232–4246, 2015.
- [15] W. Yun, W. X. An, Z. Jindan and C. Yu, "Combining vector space features and convolution neural network for text sentiment analysis," *Advances in Intelligent Systems and Computing*, vol. 72, pp. 780–790, 2019.
- [16] A. Khan, B. Baharudin, L. H. Lee and K. Khan, "A review of machine learning algorithms for text-documents classification," *Journal of Advances in Information Technology*, vol. 1, no. 1, pp. 4–20, 2017.
- [17] A. H. Khan, R. Shahid and H. A. Younas, "Assessing and prediction of living standards in smart cities using machine learning," *International Journal of Computational and Innovative Sciences*, vol. 1, no. 3, pp. 1–16, 2022.
- [18] A. S. Barroso, J. S. M. D. Silva, T. D. Souza, S. D. A. Bryanne, M. S. Soares *et al.*, "Relationship between personality traits and software quality-big five model vs. object-oriented software metrics," in *Int. Conf. on Enterprise Information Systems*, Portugal, Avenida de S. Francisco Xavier, pp. 63–74, 2017.
- [19] H. Wei, F. Zhang, N. J. Yuan, C. Cao, H. Fu *et al.*, "Beyond the words: Predicting user personality from heterogeneous information," in *Proc. of the Tenth Acm Int. Conf. on Web Search and Data Mining*, New York United States, pp. 305–314, 2017.
- [20] L. Zhou, D. P. Twitchell, T. Qin, J. K. Burgoon and J. F. Nunamaker, "An exploratory study into deception detection in text-based computer-mediated communication," in *Hawaii Int. Conf. on System Sciences*, United State, University of Hawaii, Manoa, pp. 73–93, 2013.
- [21] B. Plank and D. Hovy, "Personality traits on Twitter or how to Get 1,500 personality tests in a week," in *Computational. Approaches to Subject Sentiment Social Media Analysis*, PA, USA, pp. 92–98, 2015.
- [22] D. Sewwandi, K. Perera, S. Sandaruwan, O. Lakchani, A. Nugaliyadde *et al.*, "Linguistic features-based personality recognition using social media data," in *Proc. 6th National Conf. Technology Management*, United State, Baltimore Convention Center, Maryland, pp. 63–68, 2017.
- [23] H. P. Hsieh, C. T. Li and X. Gao, "T-gram: A time-aware language model to predict human mobility," *Proc. of the Int. AAAI Conf. on Web and Social Media*, vol. 9, no. 1, pp. 614–617, 2017.
- [24] N. Majumder, S. Poria, A. Gelbukh and E. Cambria, "Deep learning-based document modeling for personality detection from text," *IEEE Intelligent System*, vol. 32, no. 2, pp. 74–79, 2017.
- [25] F. Liu, J. Perez and S. Nowson, "A language-independent and compositional model for personality trait recognition from short texts," in *15st Conf. European Chapter Associate Computational Linguist*, Valencia, Spain, vol. 2, pp. 754–764, 2016.
- [26] M. Asif, S. Abbas, M. A. Khan, A. Fatima and S. W. Lee, "Mapreduce based intelligent model for intrusion detection using machine learning technique," *Journal of King Saud University-Computer and Information Sciences*, vol. 5, no. 3, pp. 1–14, 2021.
- [27] D. B. Wallace and H. E. Gruber, "Creative people at work: Twelve cognitive case studies," *Choice Review*, vol. 27, no. 9, pp. 27–49, 2018.
- [28] T. Yilmaz, A. Ergil and B. İlgen, "Deep learning-based document modeling for personality detection from turkish texts," *Advanced Intelligent System Computational*, vol. 1069, no. 3, pp. 729–736, 2020.
- [29] M. A. Rahman, A. Faisal, T. Khanam, M. Amjad and M. S. Siddik, "Personality detection from text using convolutional neural network," in *1st Int. Conf. Advance Science and Technology*, Dhaka, Bangladesh, pp. 1–6, 2019.
- [30] S. Muneer and M. A. Rasool, "A systematic review: Explainable Artificial Intelligence based disease prediction," *International Journal of Advanced Sciences and Computing*, vol. 1, no. 1, pp. 1–6, 2022.