SN Computer Science (2026) 7:616
https://doi.org/10.1007/s42979-026-05188-x

ORIGINAL RESEARCH

Dilated Separable Residual Network (DSRNet): Lightweight model for Personality Recognition using LLM Augmented Text Data

Devraj Patel[1] · **Sunita V. Dhavale**[1] · **Bhushan B. Mhetre**[2]

Received: 12 May 2025 / Accepted: 8 June 2026

Abstract

Personality significantly influences our attitudes/reactions/choices towards various situations and social interactions, affecting task suitability and team compatibility. Personality trait identification and analysis are vital in personal development, job selection, and performance enhancement. The Myers-Briggs Type Indicator (MBTI) is a widely used framework in social computing and human–computer interaction for understanding individual personality traits. Traditionally, MBTI assessments relied on psychologists, thus introducing subjectivity and bias. After post-pandemic, with increased online interactions using social media platforms, identifying traits without direct intervention through natural interactions is becoming inevitable before formal testing. Recently, deep learning algorithms have shown promise in developing such automated techniques for Personality Recognition; however, they face challenges with existing imbalanced data. We propose an efficient oversampling strategy using the transformer model GPT-2 and Synthetic Minority Oversampling Technique (SMOTE) to address imbalanced data problems with existing MBTI datasets. We introduce a novel lightweight CNN architecture called Dilated Separable Residual Network (DSRNet) that employs depth-wise separable convolutions to minimize computational cost while achieving an average accuracy of 90.05%, with a macro-averaged F1-score of 0.9003 across the four MBTI dimensions, demonstrating balanced performance on imbalanced personality classes.

Keywords Imbalanced data handling · MBTI · Personality recognition · Separable convolution network · Word embedding

Introduction

Personality recognition plays a vital role in understanding human behaviour, enabling better interpersonal interactions, adaptive systems, and decision-making processes. Although assessing personality can be challenging for individuals without psychological training, its impact is far-reaching—affecting job satisfaction, stress responses, preferences, aspirations, and more. In digital contexts, such as social media or online learning environments, identifying personality traits can enhance personalization and user engagement. For instance, tailoring educational content to suit a learner's personality can lead to more effective and enjoyable learning experiences.

While machine learning has been widely adopted for tasks such as sentiment analysis using text data, as presented in [1, 2], research on personality recognition is comparatively limited. A few studies, such as [3], have proposed ensemble approaches; however, progress has been hindered by the scarcity of balanced, high-quality datasets. Accurate

Sunita V. Dhavale and Bhushan B. Mhetre have contributed equally to this work.

✉ Devraj Patel
devraj.patel23@gmail.com

Sunita V. Dhavale
sunitadhavale@diat.ac.in

Bhushan B. Mhetre
bhushanbmhetre@gmail.com

[1] Department of Computer Science and Engineering, Defence Institute of Advanced Technology (DU), Girinagar, Pune, MH 411025, India

[2] Department of Psychiatry, Smt. Kashibai Navale Medical College and General Hospital, Narhe, Pune, MH 411041, India

Published online: 29 June 2026

MBTI prediction is beneficial not only for applications such as education [4, 5], recruitment [6–9], and adaptive human–computer interaction [10, 11] but also for advancing general strategies to handle class imbalance in text classification tasks. To address this gap, we propose a novel approach for recognising MBTI personality traits using general conversational text from blogs and social media platforms. Unlike prior models based on LSTMs or Transformers, our architecture is optimised for low-resource, real-time inference, a domain largely overlooked in the existing literature. Specifically, the primary research objective of this work is to investigate whether an efficiency-aware convolutional architecture can achieve competitive performance in personality recognition while significantly reducing computational complexity, inference latency, and memory requirements compared to Transformer-based models.

To the best of our knowledge, this is the first work that combines GPT-2–based text augmentation with SMOTE oversampling for MBTI classification. Unlike existing approaches that rely solely on feature-space oversampling or generative text augmentation, this work proposes a hybrid augmentation pipeline that operates at both the textual and embedding levels, explicitly targeting the severe class imbalance in MBTI datasets. However, the proposed approach is not intended to replace state-of-the-art Transformer models in high-resource settings; instead, it is designed for deployment scenarios where computational resources, memory, or inference time are constrained, such as edge devices and real-time systems.

This work focuses on automating MBTI personality prediction while overcoming challenges such as imbalanced data and suboptimal model performance. Key contributions of our research include:

(a) Leveraging Generative Pre-trained Transformer (GPT) models to generate synthetic samples, effectively addressing the class imbalance in the MBTI dataset.
(b) Designing a lightweight architecture (DSRNet) combining separable convolutions and residual connections, explicitly optimised for low-latency and low-memory personality recognition from text in resource-constrained environments.
(c) Performing systematic hyperparameter optimisation, including dilation rates, learning rates, and activation functions, to enhance model performance.
(d) Incorporating Global Average Pooling (GAP) layers to reduce overfitting while maintaining model efficiency.
(e) Developing a robust preprocessing pipeline to clean and refine input data, thereby improving accuracy.

By explicitly prioritising computational efficiency, this work positions itself as a deployment-oriented study rather than a pure accuracy-driven benchmark, complementing existing Transformer-based personality recognition approaches. The subsequent sections outline our work on personality recognition through deep learning methods. Section II provides an overview of related research, while Section III details our methodology and the experimental setup. Section IV presents the results and analyses of our proposed techniques. Finally, Section V summarises our findings, discusses potential applications, and offers suggestions for future research.

Related Work

The task of personality recognition has gained significant attention, driven by its potential applications across computational social science and human-computer interaction. Early studies by Vinciarelli et al. [12] laid the groundwork by exploring the utility of social media data for automatic personality recognition, highlighting the potential of textual and behavioural cues in personality assessment. Subsequent work by Xue et al. [13] advanced this understanding by introducing label distribution learning, which allowed for finer granularity in personality recognition tasks.

Over time, researchers have recognized the challenges posed by context variability and data imbalance in personality computing. Subramanian et al. [14] emphasized the importance of contextual factors and variability in personality expression, advocating for multifaceted computational approaches. Addressing data imbalance, Wang et al. [15] proposed a SMOTETomek-based resampling method, demonstrating its effectiveness on the MyPersonality dataset [16], while Aung et al. [17] reviewed literature emphasizing the richness of social media data for personality analysis.

Recent works have increasingly focused on leveraging deep learning models and contextual embeddings for personality prediction. For instance, Pavan et al. [18] developed a linguo-stylistic personality assessment system that utilized contextual language embeddings to infer latent personality traits from social network activity, achieving an average accuracy of 81.87%, with the highest accuracy for the Intuition-Sensing (N-S) dimension. However, their approach faced challenges due to data imbalance, which impacted performance consistency across all personality dimensions. Similarly, Zhang et al. [19] introduced PersEmoN, a deep Siamese-like network designed for the joint analysis of apparent personality and emotion, demonstrating the intertwined nature of these attributes. Despite its promising results, PersEmoN struggled with data imbalance and variations in dataset sources, which limited its generalizability and consistency across diverse datasets.

Sakdipat Ontoum et al. [20] explored the prediction of MBTI personality types using text from social network

profiles, achieving 83.55% accuracy with a bidirectional long-short term memory (BiLSTM) model. More recently, the PersoNet framework, introduced by Sandra et al. [21], has demonstrated significant advancements in personality classification through its BiLSTM architecture. PersoNet employs a mathematically optimized structure, incorporating sequence padding and dense-layer optimization to effectively process textual data. However, despite its effectiveness, the framework is computationally intensive, requiring substantial resources for both training and deployment, which may limit its accessibility for practical, large-scale applications.

For multimodal recognition, Liao et al. [22] proposed an open-source benchmark integrating audio-visual data, addressing both apparent and self-reported personality traits. In textual personality detection, Kwon et al. [23] employed a curriculum learning strategy with class-based TF-IDF features, significantly enhancing detection accuracy.

Efforts to tackle data imbalance and improve prediction reliability have also seen progress. Ryan et al. [24] applied Word2Vec embeddings and SMOTE to mitigate class imbalance in MBTI personality prediction, achieving an F1 score of 0.8337. Pradnyana et al. [25] combined Bi-LSTM with GloVe embeddings and QER, achieving 85.96% accuracy. Hao Lin et al. [26] introduced DLPPersonal Detection, utilizing data augmentation and pre-trained features, achieving 78.75% accuracy with a single-layer Bi-LSTM.

Beyond text, novel modalities for personality recognition have emerged. Ghods et al. [27] explored handwriting as a medium for personality inference using fuzzy inference systems, thereby expanding the domain of personality assessments. Despite these advancements, challenges remain, including developing lightweight, robust models suitable for resource-constrained environments, such as smartphones. This emphasises the need for models that balance complexity and efficiency while addressing data imbalance in personality recognition tasks. While separable/dilated CNNs, such as MobileNet [28] or Xception [29], have been explored for efficiency in vision tasks, our contribution lies in adapting dilated and separable convolutions with residual blocks specifically for text-based MBTI recognition.

Methodology and Experimental Setup

Overview of the proposed methodology

The high-level overview of the proposed methodology is presented in Fig. 1. The methods for MBTI personality prediction using the proposed Dilated Separable Residual Network (DSRNet) involve a systematic process divided into three main stages. In the Dataset Preparation Stage, we analyse the structure and content of the raw MBTI dataset to understand the data. Next, we clean the data to remove noise and irrelevant elements, ensuring the Dataset's quality. To address the issue of class imbalance in our data, we use methods to generate additional synthetic data, helping balance the classes. Next, we split the data into two parts: one for training and the other for testing.

In the Preprocessing Stage, the textual data undergoes tokenization and padding, converting it into numerical sequences of uniform length. The label encoding transforms the personality type labels into a machine-readable format. We then feed the prepared data into the Model Training and Evaluation Stage for proposed DSRNet model. We evaluate the model's performance using various metrics and conduct an ablation study to analyze the significance of different components within the model. This comprehensive methodology ensures an efficient and accurate prediction pipeline for MBTI personality traits.

The personality recognition task involves assigning one category from four binary dimensions—Extraversion/Introversion (E/I), Sensing/Intuition (N/S), Thinking/Feeling (F/T), and Judging/Perceiving (J/P)—to each online post x_i in a given MBTI dataset. We frame this task as a multi-label binary classification problem, dividing each MBTI trait into these four dimensions and creating separate binary labels for each.

The model is independently trained for each dimension, generating prediction functions $f_{E/I}(X)$, $f_{N/S}(X)$, $f_{F/T}(X)$, and $f_{J/P}(X)$. For a post x_i, the final MBTI prediction is obtained by concatenating these predictions into a multi-label vector as,

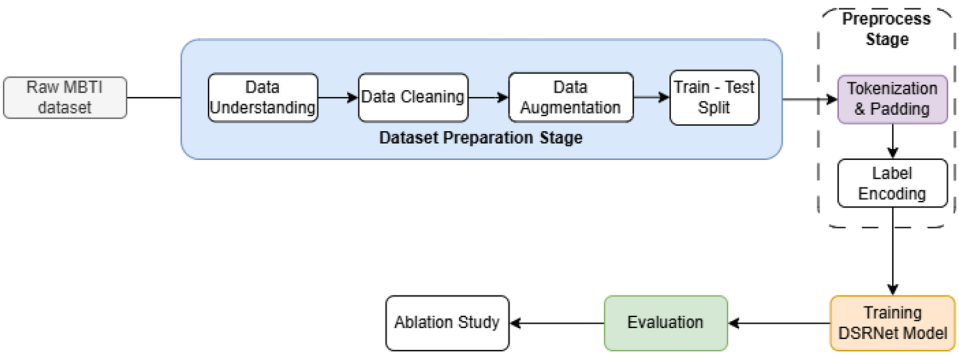

Fig. 1 Overview of the proposed method

$$Y_{\text{MBTI}} = \left[f_{E/I}(x_i), f_{N/S}(x_i), f_{F/T}(x_i), f_{J/P}(x_i)\right]. \quad (1)$$

For example, if $Y_{MBTI} = [0, 1, 1, 1]$, the predicted personality type is INFJ (Introversion, Intuition, Feeling, Judging), indicating insightfulness, compassion, creativity, and goal orientation. This automatic personality prediction can aid in understanding an individual's social and emotional tendencies, particularly enhancing team-building and cohesion in critical domains such as the military.

Dataset

This study utilises the *Kaggle MBTI Personality Dataset* [30], which comprises 8,675 user records collected from the *PersonalityCafe* online forum. Each record contains two primary fields: *Type*, representing one of the 16 Myers–Briggs Type Indicator (MBTI) personality labels (e.g., INTJ, ENFP, ISTP), and *Post*, comprising approximately 40–50 recent social media posts authored by the corresponding user.

The MBTI labels are self-reported by users based on standardised MBTI questionnaires administered on the platform prior to forum participation. While such self-reported labels may introduce noise and subjectivity, this dataset is widely adopted in computational personality recognition research and serves as a commonly used benchmark in social computing studies. The textual content reflects naturally occurring language use in online discussions, making it suitable for evaluating text-based personality classification models.

To mitigate label leakage and prevent trivial prediction cues, all explicit self-disclosures of personality types (e.g., "I am an INTJ", "as an ENFP") were systematically removed from the posts during preprocessing. This cleaning step ensures that the model learns linguistic and stylistic patterns rather than directly exploiting explicit mentions of personality labels, thereby providing a more realistic and challenging evaluation setting.

Data Understanding and Cleaning

Following an in-depth investigation, we discovered that the Dataset exhibits a sparse distribution of personality traits, as illustrated in Fig. 2. Specifically, the Dataset contains an overrepresentation of data for traits such as INFP, INFJ, INTP, and INTJ. In contrast, there is a significantly lower representation of traits such as ESTP, ESFP, ESTJ, and ESFJ. This imbalance poses a challenge for training robust, unbiased deep learning models.

Further exploration has shown that specific personality traits, such as "E-Extroversion" and "S-Sensing," are predominantly imbalanced in the Dataset, as shown in Fig. 3.

Additionally, several features and patterns in the text might create noise and degrade the accuracy of future studies. The Dataset includes several problems that were obtained from an internet forum. These problems involve variances in contractions, inconsistent letter case, noise elements (user mentions, hashtags, and URLs), and the potential for adding unrelated information. Additional Dataset complexity included unlemmatized words, mentions of explicit personality types, and specific text patterns. By carefully addressing dataset problems, the datasets were made suitable for personality prediction. Steps undertaken to prepare the textual data within a specified column in the Dataset are as enumerated below: -

(a) To ensure uniformity, the first step is to standardise the text case. Lowercasing is used to make sure that related words are consistent and recognised even when their representations in capital and lowercase differ.
(b) A specific function is then used to expand contractions inside the Dataset. When text is entered, this function expands contractions to their complete form. For instance, "don't" becomes "do not," while "can't" becomes "cannot."

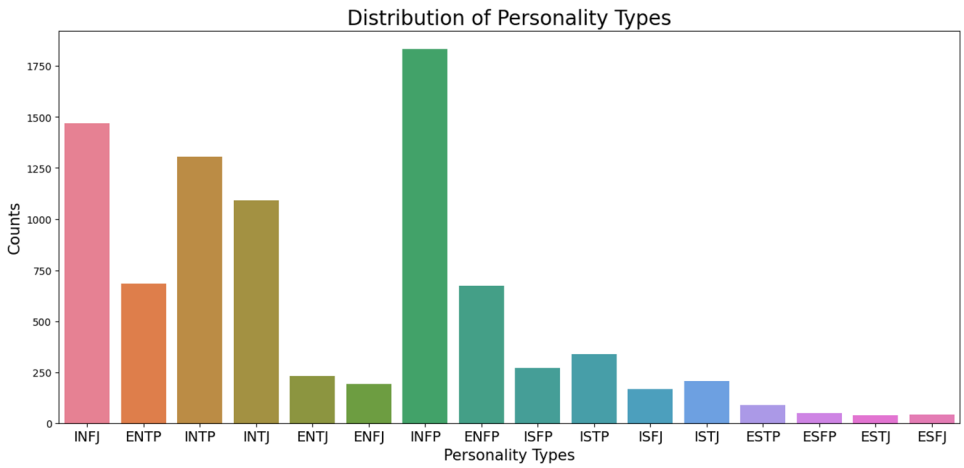

Fig. 2 Personality Distribution on MBTI Dataset

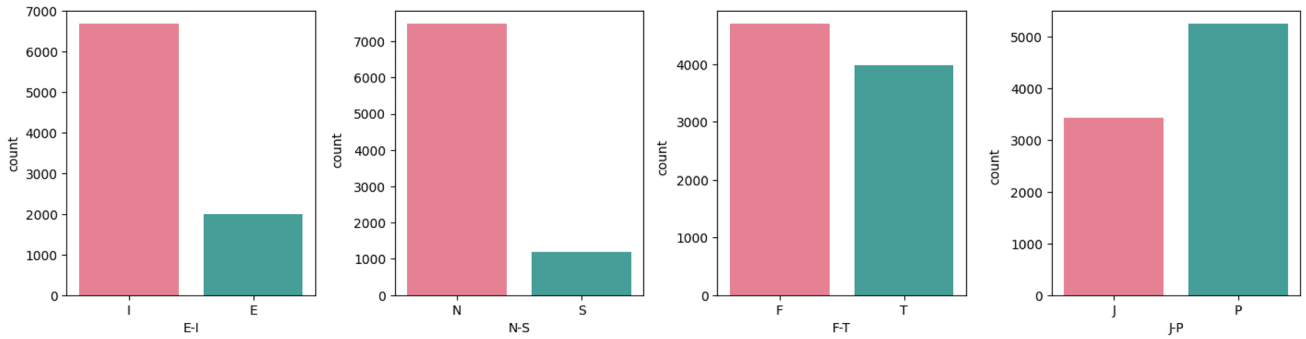

Fig. 3 Data Imbalance

(c) To exclude pointless hyperlinks, mentions (like a dataset and URLs from the text, a regular expression is employed.
(d) Understanding the role of language cues in expressing personality traits, the Dataset is filtered so that only alphabetic characters remain. Multiple spaces and non-alphabetic characters are substituted to maintain the primary language's fundamental terms.
(e) Certain personality type words, such as ENFJ and INTP in both cases (upper and lower), were eliminated from the posts to prevent direct assistance in efficient training.
(f) Lastly, the lemmatisation technique reduces terms to their base form. Lemmatisation creates synonym sets, thereby improving semantic consistency.

We excluded stop words that contribute little to contextual understanding to enhance the Dataset, such as particular characters, punctuation, and numerical data. Single, short-character words like I, me, and you are retained for their relevance to personality features.

To address the class imbalance in the Kaggle MBTI personality dataset, we adopt a hybrid data augmentation strategy that combines transformer-based text generation with feature-space oversampling. Specifically, a pre-trained transformer model, GPT-2 [31], is employed to generate synthetic raw textual posts for underrepresented MBTI classes. These generated samples are designed to preserve the linguistic and stylistic characteristics of minority personality types, thereby increasing textual diversity and improving class representation.

The augmentation process operates across two distinct representational spaces. In the first stage, GPT-2 generates additional raw text samples, which are subsequently passed through the same preprocessing, tokenisation, and embedding pipeline as the original dataset to ensure consistency. In the second stage, the Synthetic Minority Oversampling Technique (SMOTE) [15] is applied exclusively to the resulting embedding vectors, rather than to raw text, by interpolating between minority-class feature representations in the embedding space.

This hybrid approach leverages GPT-2 to enhance linguistic variability while utilising SMOTE to achieve balanced feature-space distributions, thereby avoiding the simple duplication of samples that can lead to overfitting. Although this multi-stage augmentation introduces additional computational cost, it is performed entirely during offline training and therefore does not impact inference-time efficiency of the proposed model.

Data Augmentation using Transformers

We utilised the transformer architecture, Generative Pre-trained Transformer 2 (GPT-2), to generate an additional Dataset from the original Dataset. GPT-2 is a language model trained on 1.5 billion parameters that uses the transformer architecture to generate text. The transformer architecture uses self-attention mechanisms to assess the relative importance of words in a sentence. GPT-2 is constructed using transformer decoder blocks that generate one token at a time and can handle a sequence of up to 1,024 tokens. Each token passes through all the decoder blocks along its path, resulting in a vector that can then be evaluated against the model's vocabulary.

Although more recent large language models, such as LLaMA [32] and Mistral [33], exist, GPT-2 was selected due to its low computational overhead, open availability, and suitability for controlled text generation in resource-constrained environments. The focus of this work is not to benchmark generative models, but to evaluate the impact of synthetic data augmentation on downstream personality classification performance.

Text generation with the GPT-2 Transformer model, as shown in Fig. 4, involves encoding the input text, processing it with a context window using self-attention, and generating contextualised text via sampling or beam search. The context window size is set to 150 to capture token

Fig. 4 Transformer architecture

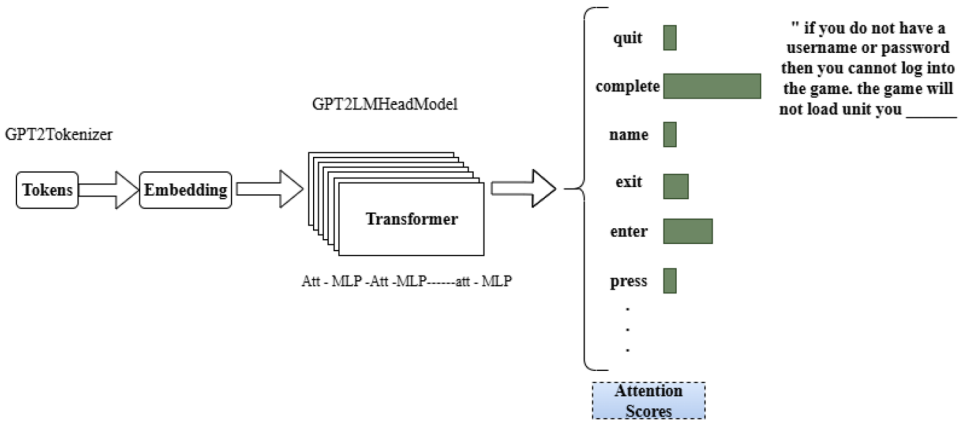

Table 1 Parameters used for text generation

Parameter	Value
Input text	Minority class posts
Context window size	150
Sampling/Beam Size	5
No repeat n-gram size	2
Temperature	0.7
Top_k	50
Top_p	0.95

relationships. The attention score a_{ij} for token i attending to token j is calculated as: -

$$a_{ij} = \text{softmax}\left(\frac{Q_i K_j^T}{\sqrt{d_k}}\right) \tag{2}$$

where Q_i is the query for token i, K_j is the key for token j, and d_k is the dimensionality of the key vectors [31].

Randomness in generation was controlled using the temperature parameter. A moderate value of 0.7 was chosen to balance diversity and coherence, ensuring contextually relevant yet varied synthetic posts. To further refine generation, we combined top-k and top-p (nucleus) sampling: top-k (k=50) restricts selection to the 50 most likely tokens, while top-p (p=0.95) adaptively includes tokens until their cumulative probability reaches 95%. This combination prevents incoherent words while preserving fluency.

Additional parameters were tuned for quality: a sampling/beam size of 5 to balance diversity and efficiency, and a no-repeat n-gram size of 2 to avoid repetitive phrasing. Together, these settings ensured that the augmented MBTI dataset was both diverse and grammatically consistent, while remaining computationally efficient.

Parameters for Text Generation

The parameter values used for text generation are summarised in Table 1.

Table 2 Sample text generated using GPT with precision, recall, and F1-score calculated by BERT-score

Original Text:
I think I used to be the same as you when I first started college (no matter how difficult the concept, I'd rather look for help online or just try to figure it out on my own). As I went through...

GPT Generated Text 1:
I got to the point where I didn't know what I wanted to do with my life. I started to feel like I was in the wrong place at the right time, and I had to find a way to get out of it. It took me a long time to realize that I wasn't alone in my desire to become a better person, but I knew that there was a lot of other people out there who could help me get there. So I decided to take a step back and look at what
Bert-Score: P: 0.8423, R: 0.8371, F1: 0.8397

GPT Generated Text 2:
I started to realize that I didn't have the time or resources to do it all myself. I had to learn how to make it work for myself and my family. It took me a while, but I finally came to the realization that it was worth it. And I'm glad I did, because it's been a great learning experience for me, and I can't wait to share it with you all! I hope you enjoy it as much as I do!
Bert-Score: P: 0.8471, R: 0.8442, F1: 0.8457

GPT Generated Text 3:
I started to feel like I was in the wrong place at the right time. I felt like my life was being taken over by people who wanted to make me feel bad about myself, and I didn't know what to do about it. It took me a long time to realize that I wasn't the only one who was feeling this way, but I knew that it was happening to me. So I decided to try something new and change the way I looked at myself and how I viewed myself.
Bert-Score: P: 0.8458, R: 0.8400, F1: 0.8429

We ran three iterations using the exact original text to demonstrate the variability in text generated by the Transformer model. The sample text generated for the 'ISFP' labelled post in all three iterations is shown in Table 2.

BERT-Score Evaluation

BERT (Bidirectional Encoder Representations from Transformers)-score [34], developed by Google, evaluates the text quality by comparing contextual embeddings of the reference and generated texts. Unlike traditional n-gram metrics, BERT-score captures detailed contextual

relationships of the Dataset, a nuanced quality assessment. The `score` function from the `bert_score` package calculates precision, recall, and F1-score. Mathematically,

$$\text{BERTScore}(P,R,F1) = \begin{cases} P = \frac{1}{|C|}\sum_{x\in C}\max_{y\in R}\text{Sim}(x,y) \\ R = \frac{1}{|R|}\sum_{y\in R}\max_{x\in C}\text{Sim}(x,y) \\ F1 = \frac{2\cdot P\cdot R}{P+R} \end{cases} \quad (3)$$

where:

C : Set of embeddings for the candidate sentence.
R : Set of embeddings for the reference sentence.
$\text{Sim}(x,y)$: Cosine similarity between embeddings x and y.

For the GPT-2-generated Dataset of 4917 posts, the average F1-score of 0.9286 demonstrates strong alignment between the original and generated Datasets indicating its effectiveness for data augmentation in this domain.

To further ensure the reliability of GPT-2-generated samples, we explicitly enforce label consistency by generating synthetic data only from minority-class instances. Additionally, a multi-stage filtering pipeline is employed to maintain semantic quality. This includes (i) repetition control using a no-repeat n-gram constraint, (ii) semantic validation using BERTScore, where samples with an F1-score below 0.80 are discarded, and (iii) manual inspection of a subset of generated samples (5%) to verify contextual coherence and linguistic plausibility. These steps ensure that the augmented data maintains both semantic fidelity and label correctness while minimising noise introduced during generation.

While transformer-based augmentation improves class balance, it may introduce artefacts or minor deviations from real-world language distributions. To mitigate this, controlled sampling strategies and semantic filtering were applied. Nevertheless, the potential impact of synthetic data on generalisation is acknowledged as a limitation.

Data Splitting

The Dataset is split into training and testing subsets to ensure robust model evaluation and prevent data leakage. The training set fits the model, while the testing set evaluates its generalisation ability on unseen data. Specifically, an 80–20 split is used, with 80% of the data allocated to training and the remaining 20% to testing. A random seed is set to maintain split reproducibility, ensuring consistent results across multiple runs. The stratified splitting ensures that the distribution of target labels is preserved across both subsets, particularly important for imbalanced datasets. This splitting methodology provides a reliable framework for assessing model performance and mitigating overfitting.

To strictly prevent data leakage, the dataset is split into training and test sets before any augmentation. All GPT-2-based synthetic data generation and SMOTE oversampling are applied exclusively to the training set, while the test set remains completely untouched and is used only for final evaluation. This ensures a fair and unbiased assessment of model performance and prevents inadvertent information leakage from test samples into the training data.

Preprocessing Stage

The textual data undergoes preprocessing to convert into a machine-readable format suitable for deep learning models. The process involves tokenisation, padding, and label encoding.

Tokenisation

Tokenisation is employed to transform the text into sequences of integers, where each integer represents a unique word. A vocabulary is constructed using the training data, and only the most frequent words (determined by a predefined threshold) are retained to reduce dimensionality. This ensures computational efficiency by filtering out infrequent words that may introduce noise. Mathematically, for an input text $X = \{w_1, w_2, ..., w_n\}$, the tokenization process can be represented as:

$$\text{Tokenizer}(X) = [t_1, t_2, ..., t_n] \quad (4)$$

where t_i is the integer index corresponding to the word w_i in the vocabulary. The vocabulary is constructed as follows:

$$\text{Vocabulary} = \{w_i \mid w_i \in X,\ f(w_i) \geq \text{threshold}\} \quad (5)$$

Padding

To standardise input lengths, sequences are padded to align with the size of the most extended sequence in the training set. This method ensures uniformity across samples, which is crucial for models requiring fixed-length inputs. Given a tokenized sequence $T_X = [t_1, t_2, ..., t_n]$, the padding process can be mathematically represented as:

$$\text{Pad}(T_X, L) = \begin{cases} [t_1, t_2, ..., t_n, 0, 0, ..., 0] & \text{if } n < L \\ [t_1, t_2, ..., t_L] & \text{if } n = L \end{cases} \quad (6)$$

where L is the maximum sequence length (determined dynamically based on the training data), padding adds zeros to sequences shorter than L to match the maximum length.

Label Encoding

The personality type labels are encoded into numeric values using label encoding. Each unique label, representing the MBTI personality type, is mapped to a distinct integer. Given a set of labels $y = \{y_1, y_2, ..., y_m\}$, the label encoding function can be represented as:

$$\text{LabelEncoder}(y) = [l_1, l_2, ..., l_m] \tag{7}$$

where l_i is the integer corresponding to the label y_i. This encoding enables the multi-class classification task by converting categorical labels into a numeric format suitable for the model.

This preprocessing method ensures efficiency by focusing on the most frequent words in the dataset. Determining the sequence length dynamically minimizes computational overhead while preserving relevant contextual relationships. The padding ensures compatibility with neural network architectures, and the encoded labels allow for effective learning in supervised settings. This preprocessing pipeline standardizes the dataset, providing robust and scalable model training.

Model Architecture

The proposed architecture, DSRNet, integrates dilated and separable convolutional blocks specifically adapted for one-dimensional text data. As shown in Fig. 5, the model combines dilated convolution, depth-wise separable convolution, and batch normalization to achieve high performance with reduced memory and computational cost.

From a linguistic perspective, personality traits are expressed through both local syntactic cues and long-range discourse patterns. Dilated convolutions enable the model to capture such long-distance dependencies without the sequential limitations of RNNs, while separable convolutions reduce redundancy in token-level processing, providing a lightweight alternative to dense self-attention mechanisms.

The model utilises static word embeddings trained on the training corpus. While contextual embeddings such as BERT can improve representational richness, they introduce significant computational overhead. The use of static embeddings aligns with the lightweight design goals of DSRNet.

The DSRNet design adheres to the following key principles to balance efficiency and prediction accuracy:

1. **Dilated convolutions** expand the receptive field, allowing the model to capture salient, long-range textual features effectively [35, 36].
2. **Depth-wise separable convolutions** replace standard convolutions, significantly reducing computational complexity and the number of parameters [29].
3. **Global Average Pooling (GAP)** layers substitute fully connected layers before the softmax classifier, reducing overfitting and parameter count while preserving performance [28].
4. **Progressive learning rate decay** facilitates smooth convergence and mitigates training instability [37].

DSRNet addresses two critical limitations: (1) the inability of RNN-based models to capture sparse long-range dependencies efficiently, and (2) the lack of lightweight architectures suitable for deployment on edge devices. Unlike BiLSTMs and transformer-based models, DSRNet offers comparable accuracy with significantly lower computational overhead, making it ideal for real-time personality inference in resource-constrained environments.

While Transformer-based models such as BERT, RoBERTa, and DistilBERT represent the current state of the art for text classification, their self-attention mechanisms incur quadratic computational complexity with respect to sequence length, resulting in high memory usage and inference latency. This limits their practicality in edge computing and resource-constrained environments.

In contrast, the proposed Dilated Separable Residual Network (DSRNet) is explicitly designed for low-latency and low-memory inference. DSRNet employs 1D convolutions

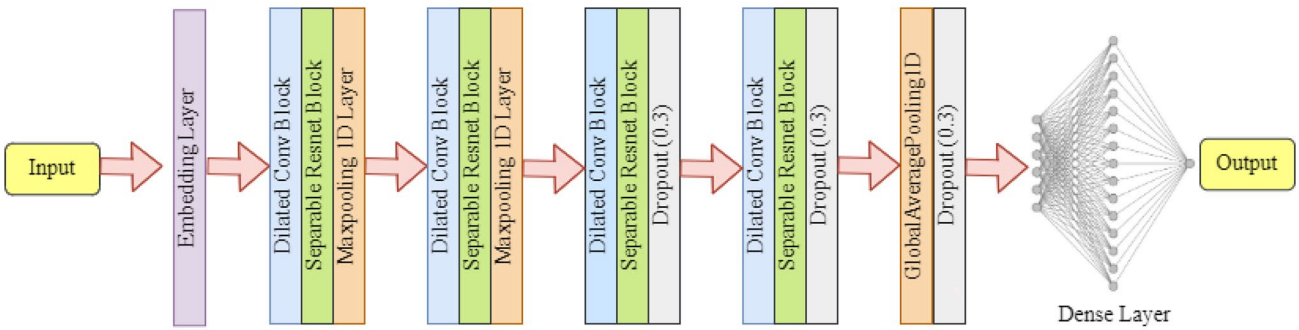

Fig. 5 DSRNet Model Architecture

with dilated receptive fields to effectively model long-range contextual dependencies while maintaining linear computational complexity. Furthermore, the use of depthwise separable convolutions and residual connections significantly reduces the number of trainable parameters without compromising representational capacity.

As a result, DSRNet is well-suited for deployment on mobile devices, embedded platforms, and real-time personality recognition systems, where computational efficiency is a primary constraint.

Dilated Convolution

In the context of text data, dilated convolutions offer a significant advantage by introducing a dilation rate to the convolution operation, which expands the receptive field without increasing the number of parameters [36, 38]. We use dilated convolutions when a larger field of view is required, but we need to minimise computational cost or memory consumption. A dilation rate of 5 on a 3x3 kernel increases the receptive field to cover the same area as a 5x5 kernel but with fewer parameters. Specifically, a dilated 3x3 kernel has only nine parameters, while a 5x5 kernel requires 25, making dilated convolutions an efficient option for expanding the receptive field with minimal computational overhead [36].

The dilated convolution operation can be mathematically represented as follows:

$$y = \text{LeakyReLU}\left(\text{BatchNorm}\left(\text{Conv1D}(x, f, k, d)\right)\right) \tag{8}$$

Where:

- x represents the input sequence, which is tokenized text data.
- f is the number of filters (output channels).
- k is the kernel size, e.g., 3 for a 3x3 kernel.
- d is the dilation rate, which controls the spacing between kernel elements (e.g., $d = 2$).
- $\text{Conv1D}(x, f, k, d)$ represents the 1D convolution operation with dilation, which can capture long-range dependencies by expanding the receptive field.
- BatchNorm applies batch normalization to stabilize the learning process.
- LeakyReLU applies the Leaky ReLU activation function to introduce non-linearity.

In text classification tasks, such as MBTI personality recognition, where the data consists of sequences (such as words or tokens in a post), dilated convolutions (as shown in Fig. 6a) allow the model to capture long-range dependencies without excessively increasing the number of filters or layers. This helps retain the ability to learn contextual relationships over larger text spans while controlling the computational complexity [39].

Separable Residual Network Block

Separable convolution, a popular compression technique in 3D vision tasks such as Xception [29] and MobileNet [40], effectively reduces memory consumption and computational cost. This approach involves two primary operations: depthwise and pointwise convolution (a 1×1 convolution). While depthwise convolutions are inherently suited to multi-channel image data [41, 42], their application to one-dimensional textual representations requires careful adaptation due to the structural differences between image tensors and text embeddings.

In the proposed architecture, textual inputs are represented as sequences of word embeddings with dimensionality $(L \times D)$, where L denotes the sequence length and D represents the embedding dimension. To adapt depth-wise separable convolutions to this 1D setting, convolutional filters are applied along the temporal (sequence) dimension, while embedding dimensions are processed independently. This formulation enables efficient extraction of local and contextual n-gram patterns without introducing excessive parameters.

Although depth-wise separable convolutions are predominantly used in vision tasks, recent studies have demonstrated their effectiveness in one-dimensional sequence modelling domains such as speech recognition and text classification. Within DSRNet, this adaptation substantially reduces the parameter count and computational overhead while preserving the model's discriminative capacity for learning linguistic features.

Specifically, for MBTI personality recognition from text, separable convolution is modified to suit a 1D CNN framework. As text embeddings effectively form a single-channel input, the depthwise convolution becomes computationally lightweight. Consequently, we employ optimised kernel sizes (e.g., $k \in 2, 3, 5$), following the principles outlined in the Optimised Text CNN Architecture [43]. Rather than deploying multiple filters per kernel size, a single filter is used for each, followed by a pointwise (1×1) convolution to increase representational depth and introduce non-linearity. This design strikes a balance between expressive power and efficiency, aligning with the lightweight objectives of the proposed DSRNet architecture.

The operation within the Separable Residual Network Block can be mathematically expressed as:

$$\text{DepthConv}(x) = x * f_d, \quad \text{PointConv}(x) = x * f_p \tag{9}$$

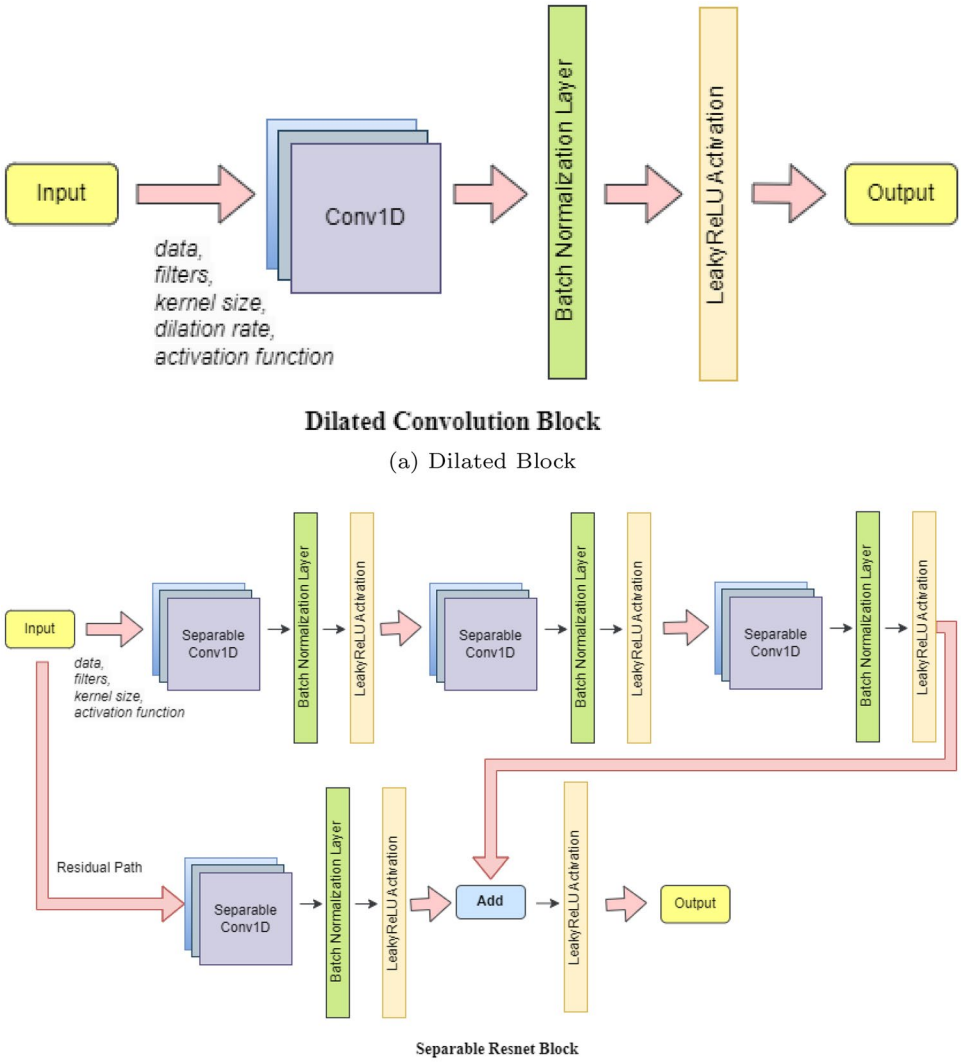

Fig. 6 Dilated and Separable Network Structure

(a) Dilated Block

(b) Separable ResNet Block

where:

- x represents the input feature map (text embedding),
- f_d is the depthwise convolution kernel (one kernel per input channel),
- f_p is the pointwise convolution kernel (a 1×1 kernel for channel projection),
- $*$ denotes the convolution operation.

The output of the Separable Convolution operation is given by:

$$\text{SeparableConv}(x) = \text{LeakyReLU}(\text{BatchNorm}(\text{PointConv}(\text{DepthConv}(x)))) \quad (10)$$

Residual connections are incorporated to align the input and output dimensions within the block. This can be expressed as:

$$\text{Output} = \text{LeakyReLU}(\text{BatchNorm}(\text{SeparableConv}(x) + \text{ResidualPath}(x))) \quad (11)$$

where:

- $\text{ResidualPath}(x)$ ensures dimensional alignment via a separable convolution layer.
- LeakyReLU introduces nonlinearity with $\alpha = 0.2$.
- BatchNorm stabilizes learning by normalizing feature distributions.

Implementing separable convolutions with an additional separable convolution layer increases feature extraction capabilities (see Fig. 6b). The 1×1 convolution contributes to deeper networks, empirically better than wider networks, while activation layers introduce nonlinearity. This ensures extracting meaningful features from text embeddings,

ultimately improving the model's ability to recognise MBTI personality traits effectively.

Batch Normalisation

Batch Normalisation (BN) is applied after each convolutional layer to address challenges like vanishing or exploding gradients and slow convergence during training. By normalising the activations, BN ensures that outputs remain within a reasonable range, stabilising the training process and accelerating convergence [44]. Specifically, in the context of `SeparableConv1D`, BN maintains efficient gradient flow, reduces internal covariate shift, and minimises dependency on weight initialisation [45]. When combined with the Leaky ReLU activation function, BN helps retain negative activations, improving model robustness and enhancing overall performance [46].

The mathematical formulation of Batch Normalisation can be expressed in Eq. 12:

$$\mathrm{BN}(x) = \gamma \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{12}$$

Where:

- x represents the input activation.
- μ and σ^2 are the mean and variance of the activations computed over the mini-batch.
- γ and β are learnable scaling and shifting parameters, respectively.
- ϵ is a small constant added for numerical stability.

This equation ensures that each input activation is normalised, scaled, and shifted to enable stable and efficient learning. The combination of `SeparableConv1D`, Batch Normalization, and Leaky ReLU activation forms a robust building block for deep sequence models.

Dilated Separable ResNet (DSRNet) Model

The proposed model integrates four blocks of dilated convolution and separable residual network blocks into a unified architecture optimized for binary classification tasks. The architecture also includes two layers of MaxPool1D and two layers of dropout (with a rate of 0.3), along with one layer each of global average pooling and a dense layer. This design effectively captures hierarchical features while mitigating overfitting and improving generalization for binary classification. The overall model function can be expressed as in Eq 13: -

$$y = \mathrm{Dense}\left(\mathrm{GAP}\left(\mathrm{Dropout}\left(\mathrm{SRNet}\left(\ldots \mathrm{Dilated}(x)\right)\right)\right)\right) \tag{13}$$

Input Layer

The input x is a sequence of tokenized text with shape $(\mathrm{max\_len},)$, where $\mathrm{max\_len}$ is the maximum sequence length.

Dilated Convolutional Blocks

The input is passed through a sequence of dilated convolution operations (f_{dilated}), defined as:

$$f_{\mathrm{dilated}}(x) = \mathrm{LeakyReLU}\left(\mathrm{BN}\left(\mathrm{Conv1D}(x; f, k, d)\right)\right) \tag{14}$$

where f is the number of filters, $k = 3$ is the kernel size, and $d = 5$ is the dilation rate. These blocks expand the receptive field efficiently, capturing long-range dependencies in text.

Separable Residual Blocks

The output from the dilated convolutional blocks is processed through separable residual blocks (f_{SepRes}), defined as:

$$f_{\mathrm{SepRes}}(x) = x + \mathrm{SeparableConv}(x) \tag{15}$$

where:

$$\mathrm{SeparableConv}(x) = \mathrm{PointConv}\left(\mathrm{DepthConv}(x)\right) \tag{16}$$

These blocks reduce computational overhead while maintaining robust feature extraction. Residual connections mitigate vanishing gradient issues, allowing the model to go deeper.

Dropout and MaxPooling Layers

Dropout with a rate of $p = 0.3$ is applied at intermediate layers to prevent overfitting:

$$x_{\mathrm{dropout}} = \mathrm{Dropout}(x; p) \tag{17}$$

MaxPooling layers are used to reduce spatial dimensions while retaining critical features.

Global Average Pooling (GAP) Layer

The output of the residual blocks is passed through a global average pooling layer (x_{gap}), which aggregates features across the sequence:

$$x_{\mathrm{gap}} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{18}$$

where n is the sequence length.

Dense Output Layer

A dense layer with sigmoid activation produces the final prediction:

$$y = \sigma(W \cdot x_{\text{gap}} + b) \tag{19}$$

where W and b are learnable parameters, and σ is the sigmoid function.

Optimiser and Loss Function

The Adam optimiser is employed with a learning rate of $\alpha = 0.0001$. It updates the parameters θ according to the following equation:

$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta) \tag{20}$$

Here, $\mathcal{L}$ represents the loss function. In this case, binary cross-entropy is utilized as the loss function, defined as follows: -

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)\right) \tag{21}$$

In this equation, y_i denotes the true labels, while $\hat{y}_i$ indicates the predicted labels.

Final Model Function

Let f_{dilated} represent the dilated convolution operation, f_{SepRes} represent the separable residual block, and f_{DSRNet} represent the overall model. The final output is expressed as:

$$f_{\text{DSRNet}} = \text{Dense}\left(\text{GAP}\left(\text{Dropout}\left(f_{\text{SepRes}}\left(f_{\text{dilated}}(x)\right)\right)\right)\right) \tag{22}$$

The DSRNet model efficiently captures long-range dependencies with its dilated convolution blocks while maintaining computational efficiency using separable residual blocks. Dropout and global average pooling ensure robust generalization and reduced overfitting, making it ideal for text-based binary classification tasks. This model was tested on the Kaggle MBTI dataset, demonstrating its effectiveness for imbalanced classification problems [30].

Model Training and Evaluation

The model is trained using a custom procedure designed to enhance generalisation and prevent overfitting. Given that there are four dimensions in the MBTI, and each dimension requires binary classification, four separate models are trained on the MBTI dataset, one for each dimension. This approach enables specialised learning and more accurate predictions for each MBTI personality dimension. The training process utilises the following key parameters: -

- **Epochs:** The model is trained for a maximum of 30 epochs, allowing sufficient training iterations for convergence.
- **Batch Size:** A batch size 16 is used, balancing computational efficiency and model performance.
- **Early Stopping:** To prevent overfitting, early stopping is applied, monitoring the validation loss (L_{val}) with a patience of 10 epochs. This ensures that training stops if no improvement is seen in the validation loss for ten consecutive epochs, thus restoring the best weights.
- **Learning Rate Scheduler:** The learning rate dynamically adjusts during training based on the number of epochs. As the number of epochs increases, the learning rate decreases, following a custom schedule to ensure stable convergence.

The following function gives the training procedure: -

$$\hat{y} = f_{\text{DSRNet}}(X_{\text{train}}, \theta_{\text{train}}, \text{epochs}, \text{batch size}) \tag{23}$$

Where:

- X_{train} represents the input training data.
- θ_{train} refers to the model parameters learned during training.
- epochs is the total number of training epochs.
- batch size is the number of samples processed per update step.

For evaluation, the model's performance is assessed using several metrics: -

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} \mathbf{1}(y_i = \hat{y}_i)}{N} \tag{24}$$

Where $\mathbb{1}(y_i = \hat{y}_i)$ is an indicator function that is 1 if the prediction $\hat{y}_i$ matches the true label y_i, and 0 otherwise.

Additional evaluation metrics include: -

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{25}$$

$$\text{Precision} = \frac{\sum_{i=1}^{N} \mathbf{1}(y_i = 1 \text{ and } \hat{y}_i = 1)}{\sum_{i=1}^{N} \mathbf{1}(\hat{y}_i = 1)} \tag{26}$$

$$\text{Recall} = \frac{\sum_{i=1}^{N} \mathbf{1}(y_i = 1 \text{ and } \hat{y}_i = 1)}{\sum_{i=1}^{N} \mathbf{1}(y_i = 1)} \tag{27}$$

$$\text{ROC AUC} = \text{Area Under ROC Curve} \tag{28}$$

The DSRNet model is evaluated by comparing predicted probabilities with actual labels, and the performance metrics listed above are used to assess its effectiveness in binary classification. It is worth noting that while confusion matrices are commonly used for binary or small-scale multi-class problems, they become less practical for MBTI classification, which has more categories, making the visual representation cumbersome. Instead, we rely on precision, recall, F1-score, and AUC-ROC, which together provide a clearer and more concise quantitative assessment of class-wise balance and overall performance.

Results and Discussions

Performance Metrics for MBTI Dimensions using augmented Dataset

The table 3 presents performance metrics for the four MBTI dimensions: Extraversion-Introversion (E-I), Intuition-Sensing (N-S), Feeling-Thinking (F-T), and Judging-Perceiving (J-P), measured by various classification metrics: Accuracy, F1-Score, Precision, Recall, and ROC-AUC.

To evaluate the robustness of the proposed model, experiments were repeated three times using different random seeds. The observed variation in accuracy remained low (approximately $\pm 0.6\%$), indicating stable and consistent performance across runs.

The model demonstrates exceptional performance on the N-S dimension, with high accuracy, precision, recall, and ROC_AUC. It also performs strongly on the E-I dimension, showing a good balance across all metrics and a high ROC_AUC. Although the F-T and J-P dimensions exhibit slightly lower performance, they still fall within substantial and acceptable ranges, particularly in precision and recall. Overall, the model is highly effective at classifying MBTI dimensions, with the best results for N-S and E-I (above 92%) and slightly lower but still strong performance for F-T and J-P (85%).

The relatively lower performance observed in the F-T and J-P dimensions can be attributed to subtle linguistic variations and overlapping semantic cues between classes. These traits often depend on contextual nuances, making them more difficult to distinguish. Misclassifications are frequently associated with neutral or ambiguous expressions that lack strong personality indicators.

To objectively substantiate the lightweight nature of the proposed DSRNet architecture, we report both the number of trainable parameters and the computational cost measured in floating-point operations (GFLOPs). The complete DSRNet model contains 652,513 trainable parameters, corresponding to a memory footprint of approximately 2.49 MB, which is significantly smaller than that of Transformer-based text classifiers, which typically require tens of millions of parameters. As shown in Table 3, the proposed model requires less than 0.085 GFLOPs per forward pass, confirming its low computational overhead. This efficiency enables real-time inference and supports deployment in low-resource, latency-sensitive environments, such as mobile devices and edge systems.

Training and Validation Accuracy and ROC Analysis

The training and validation accuracy plots for the model, shown in Figs. 7, 9, 11, and 13, were obtained after training over 30 epochs with a custom learning rate schedule that decreases as training progresses. The corresponding Receiver Operating Characteristic (ROC) curves, displayed in Figs. 8, 10, 12, and 14, provide a detailed view of the classifier's ability to differentiate between classes at various threshold levels. These plots confirm that the model maintains consistent performance without overfitting.

Comparison with State-of-the-Art and Data Augmentation Baselines

Table 4 isolates the impact of different augmentation strategies. Using SMOTE alone yields moderate improvements (82.4%), while GPT-2 augmentation performs better (85.1%). However, the hybrid GPT-2 and SMOTE strategy significantly outperforms both, achieving 90.05% accuracy with balanced precision, recall, F1, and ROC-AUC, demonstrating that combining generative augmentation with synthetic oversampling provides complementary benefits

Table 3 Performance metrics for MBTI dimensions

Type	GFLOPs	Accuracy	F1-Score	Precision	Recall	ROC_AUC
E-I	0.0841	0.9246	0.9234	0.9420	0.9056	0.9705
N-S	0.0791	0.9778	0.9775	0.9868	0.9685	0.9968
F-T	0.0808	0.8529	0.8523	0.8561	0.8486	0.8663
J-P	0.0834	0.8468	0.8479	0.8462	0.8496	0.8792
Average	0.0818	0.9005	0.9003	0.9078	0.8931	0.9282

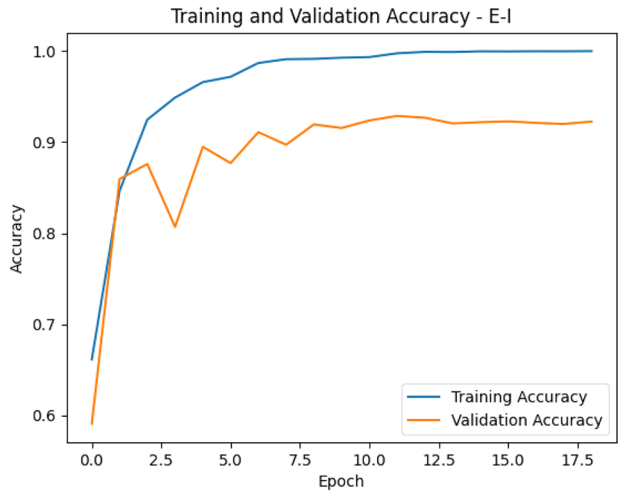

Fig. 7 Training Plot for MBTI Personality Dimension E-I

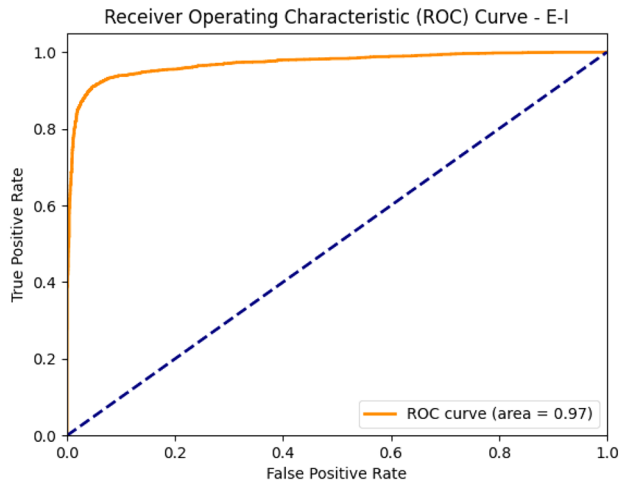

Fig. 8 ROC Curve for MBTI Personality Dimension E-I

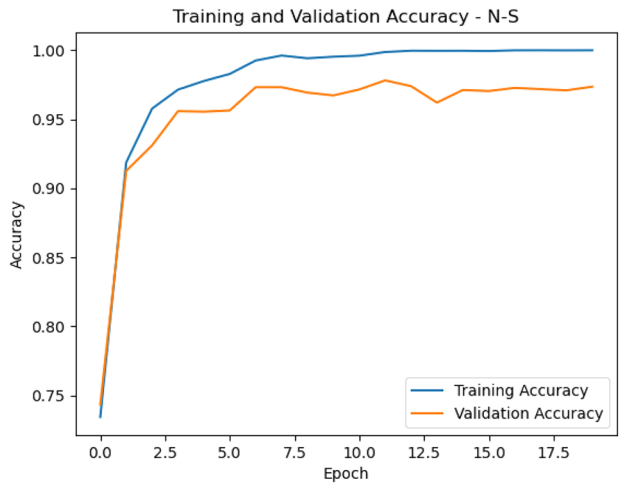

Fig. 9 Training Plot for MBTI Personality Dimension N-S

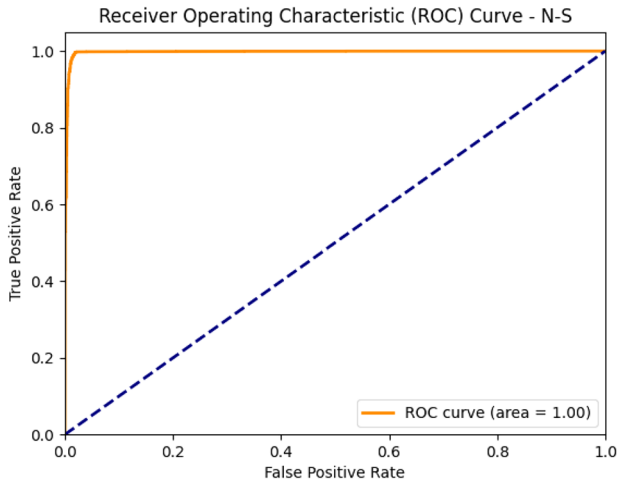

Fig. 10 ROC Curve for MBTI Personality Dimension N-S

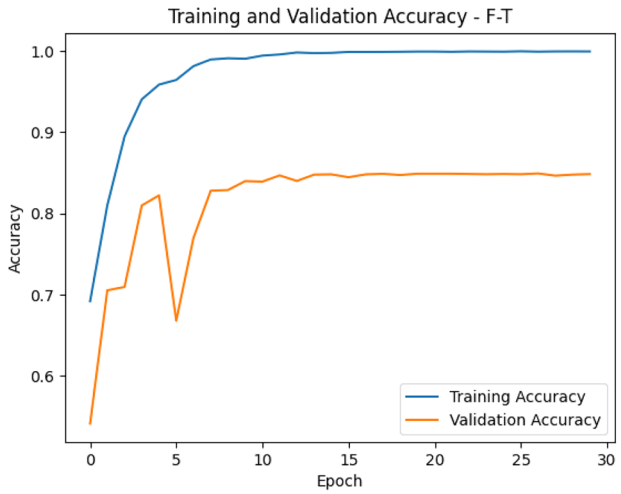

Fig. 11 Training Plot for MBTI Personality Dimension F-T

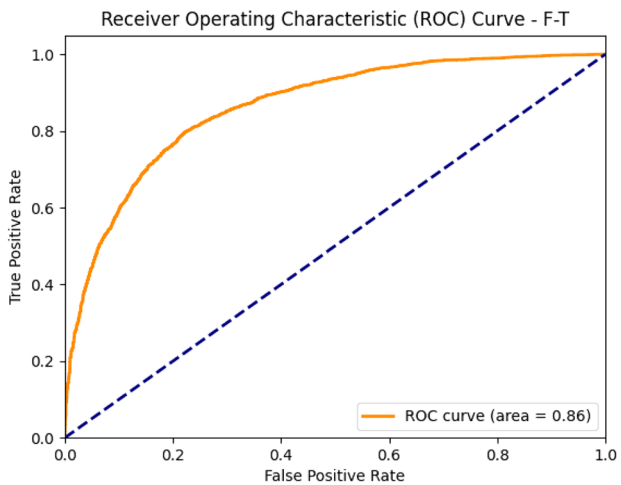

Fig. 12 ROC Curve for MBTI Personality Dimension F-T

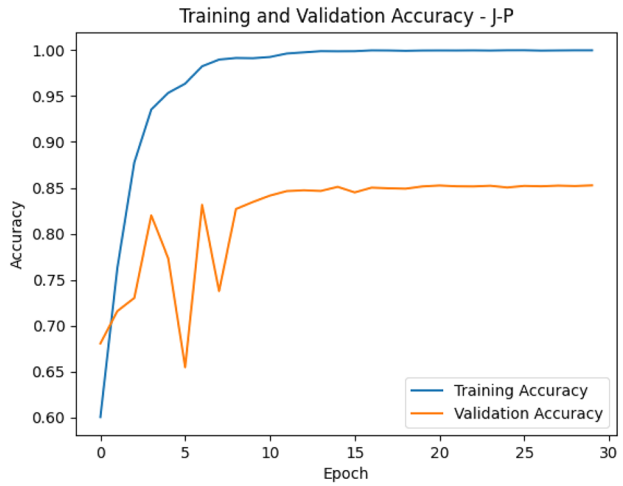

Fig. 13 Training Plot for MBTI Personality Dimension J-P

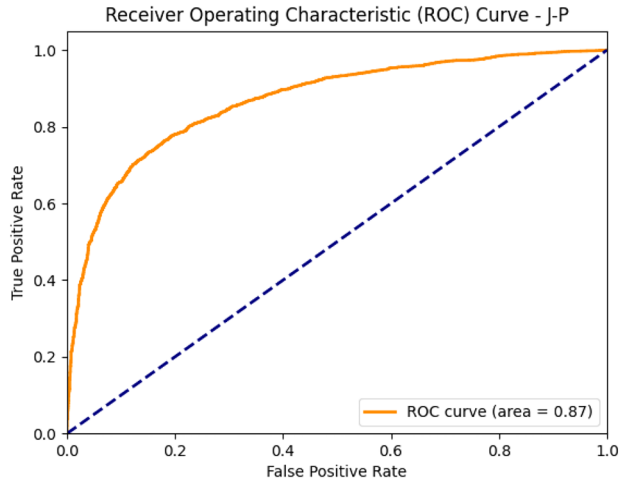

Fig. 14 ROC Curve for MBTI Personality Dimension J-P

for handling class imbalance. Although formal statistical hypothesis testing, such as a t-test, is not conducted in this study, the low variance observed across multiple runs (reported as mean ± standard deviation) indicates stable and consistent model performance.

We compared the results obtained using the proposed model architecture with those of state-of-the-art deep learning methods for recognising personality traits from text datasets, as shown in Table 5. Our GPT-SMOTE + DSRNet achieves the highest accuracy (90.05%), surpassing hybrid BiLSTM approaches and other augmentation-based baselines reported in prior work.

It is important to note that while Transformer-based architectures such as BERT [47] and RoBERTa [48] can achieve higher absolute accuracy in general NLP benchmarks, they incur substantially higher computational and memory costs, making them less suitable for resource-constrained environments. In contrast, this work focuses on a lightweight design, and comparisons are therefore made primarily with models under similar efficiency constraints. Additionally, the proposed model is evaluated on an augmented dataset, whereas some baseline methods are trained on non-augmented data, potentially introducing bias in direct comparisons. Accordingly, the results should be interpreted as indicative of relative performance trends rather than strict one-to-one benchmarking. Future work will aim to retrain baseline models under identical data conditions for more controlled evaluation.

For more information, see the project's GitHub repository: https://github.com/devrajpatel/DSRNets4MBTI.

Ablation Study

A thorough ablation study was conducted to assess the effects of key hyperparameters and architectural choices on the performance of our proposed model. Each parameter was systematically varied while keeping other settings constant, allowing us to understand its individual contribution

Table 4 Performance of different augmentation strategies on the MBTI dataset. Results are averaged over 5 runs

Method	Accuracy (%) ± Std	Precision	Recall	F1-score	ROC-AUC
SMOTE-only	82.4 ± 0.6	0.82	0.82	0.80	0.85
GPT-2-only	85.1 ± 0.5	0.85	0.85	0.84	0.87
GPT-2 + SMOTE (Proposed)	90.05 ± 0.7	0.90	0.90	0.89	0.92

Table 5 Comparison of models and datasets for MBTI classification

Authors	Models	Dataset	Avg accuracy
Wang et al. [15]	PSO-SMOTETomek + SVM	My Personality	88.15%
Pawan Kumar et al. [18]	LPTA (Linguostylistic + Ensemble)	Kaggle MBTI	81.87%
Sakdipat Ontoum et al. [20]	CRISP-DM + BiLSTM	Kaggle MBTI	83.55%
Ryan G et al. [24]	W2V + SMOTE with ML	Kaggle MBTI	83.37%
Aditra Pradnyuna et al. [25]	Hybrid BiLSTM with GloVe and QER	Kaggle MBTI	85.96%
Hao Lin et al. [26]	DLP-BiLSTM	Kaggle MBTI	78.52%
Proposed model (ours)	GPT-SMOTE + Dilated Separable ResNet	Augmented MBTI	90.05%

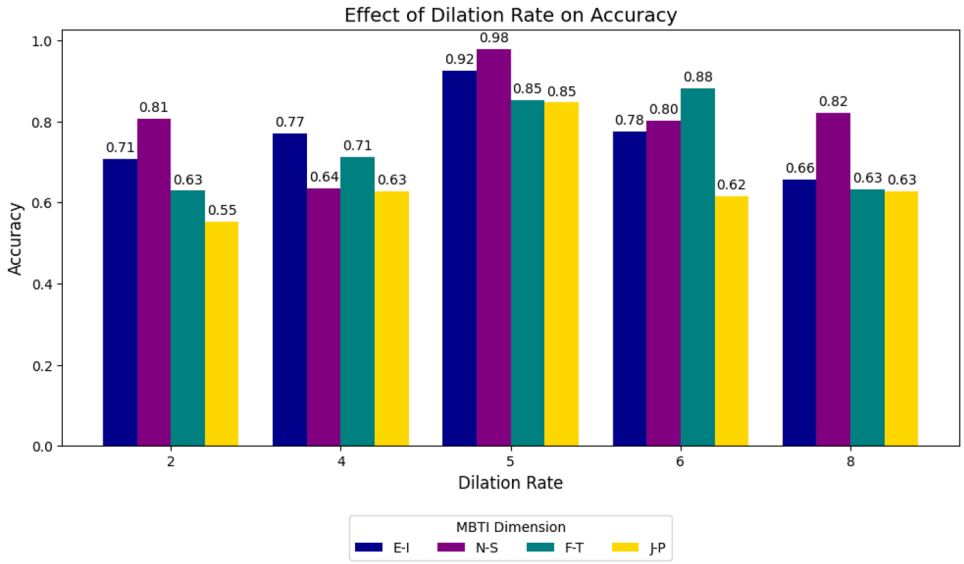

Fig. 15 Effects of dilation rate on accuracy

to overall performance. Below, we discuss how the dilation rate, learning rate, and activation functions affect the model's accuracy, precision, recall, and ROC AUC.

All experiments were conducted with fixed random seeds and consistent preprocessing pipelines to ensure reproducibility.

Effect of Dilation Rate

The dilation rate in the convolutional layers controls the spacing between kernel elements, effectively increasing the receptive field without increasing the kernel size. To analyse its impact, we experimented with dilation rates of {2, 4, 5, 6, 8}. The results showed that lower dilation rates (*e.g., 2 and 4*) captured local features well but were less effective at capturing global dependencies, leading to slightly lower performance metrics. On the other hand, high dilation rates (*e.g., 6 and 8*) introduced sparsity in the receptive field, causing a drop in performance due to the loss of fine-grained local patterns. The optimal balance was achieved at a dilation rate of 5, at which the model achieved the highest accuracy. The accuracy obtained with various dilation rates is presented in Fig. 15.

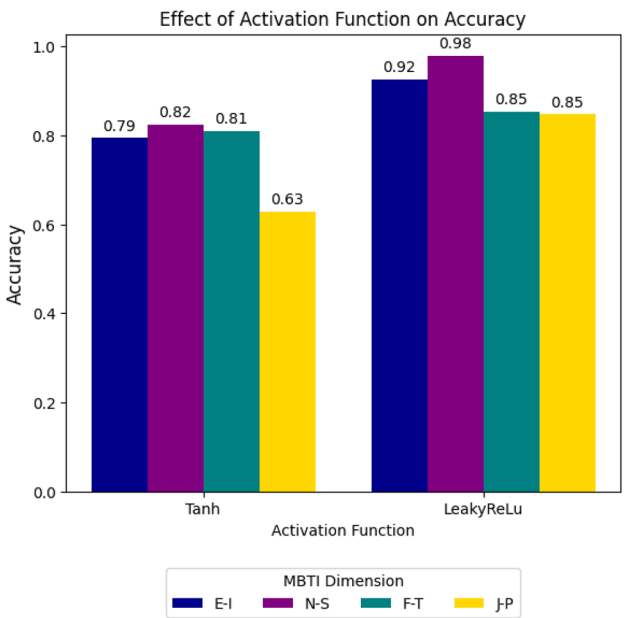

Fig. 16 Effects of activation function on accuracy

Effect of Learning Rate

Learning rate is a critical parameter that governs the speed and stability of model convergence. We conducted experiments with learning rates in the range {0.0001, 0.001, 0.002, 0.003, 0.1}. A meager learning rate (*e.g., 0.0001*) resulted in slow convergence, while a high learning rate (*e.g., 0.1*) led to unstable training dynamics and failure to converge. The model achieved optimal performance with the learning rate schedules, balancing convergence speed and stability.

Effect of Activation Function

Activation functions are essential for introducing nonlinearity into models, enabling them to learn complex patterns. In our analysis, we evaluated the performance of the model using four activation functions: ReLU, Leaky ReLU, Sigmoid, and Tanh. Both ReLU and its variant, Leaky ReLU, outperformed Sigmoid and Tanh. This can be attributed to their ability to mitigate the vanishing gradient problem and support faster computation. Interestingly, Leaky ReLU slightly outperformed ReLU by maintaining gradient flow for negative values, which contributed to enhanced ROC

AUC scores. The accuracy results for the Tanh and Leaky ReLU activation functions in the proposed model are shown in Fig. 16.

Conclusion

This study introduces the Dilated Separable Residual Network (DSRNet), a lightweight and robust model for MBTI personality recognition from textual data. By integrating GPT-2-based data augmentation and SMOTE oversampling, the proposed framework effectively mitigates class imbalance and improves generalization. Leveraging dilated and separable convolutions, DSRNet achieves over 90% accuracy across MBTI dimensions while maintaining low computational complexity. Experimental results highlight DSRNet's superior performance and scalability compared to existing methods, making it well-suited for real-world applications such as personalized learning, team dynamics optimization, and user engagement across digital platforms. This work lays the groundwork for the practical deployment of personality recognition systems in bandwidth-constrained and latency-sensitive environments, including mobile applications and chatbots.

Despite these strengths, the model was trained exclusively on the Kaggle MBTI dataset, which limits cultural and platform diversity. Moreover, GPT-2–based augmentation may introduce synthetic artefacts that affect real-world generalisation. Synthetic text generation introduces risks, including semantic drift and stylistic inconsistencies. To mitigate these effects, strict dataset partitioning was enforced, ensuring that GPT-2 was never exposed to evaluation data during training. Additionally, generated samples were validated using BERT-Score to maintain semantic coherence.

While GPT-2 was selected in this study due to its low computational overhead and controlled generation capabilities, the primary objective was to evaluate the effectiveness of synthetic data augmentation rather than to benchmark generative language models. The experimental results demonstrate that, even with a lightweight text generator, the proposed augmentation strategy significantly improves downstream MBTI classification performance when combined with the DSRNet architecture.

We acknowledge that the MBTI has faced criticism in the psychological literature for its limited test-retest reliability and its categorical approach to personality assessment. Therefore, this work does not assert psychological validity; instead, it uses the MBTI as a benchmark for a classification problem commonly adopted in computational personality recognition research.

Automated personality inference from text raises ethical concerns regarding privacy, profiling, and potential misuse. The proposed system is designed for decision-support applications, such as adaptive learning systems, rather than for deterministic personality labelling. Safeguards should be put in place to prevent discriminatory or manipulative uses of this technology.

Future work will explore lightweight LLMs (e.g., compact LLaMA or Mistral variants) for synthetic text generation to improve linguistic diversity and semantic coherence while maintaining efficiency. Systematic comparisons will assess trade-offs among model complexity, augmentation quality, and downstream performance in personality classification. The framework will be extended to continuous models, such as the Big Five (OCEAN), multilingual settings, and multimodal data (including audio and visual). Model interpretability will be enhanced using SHAP, LIME, and attention visualisations to identify key textual features for each MBTI dimension, thereby improving transparency, trust, and bias mitigation.

Acknowledgements This research utilized the resources of the Paramshakti supercomputing facility at IIT Kharagpur, established under the National Supercomputing Mission (NSM) of the Government of India and supported by CDAC, Pune. The authors sincerely appreciate the computational resources and support provided, which significantly contributed to the successful completion of this study.

Funding This work was not supported by any external funding sources.

Data Availability The data can be obtained from the corresponding author upon reasonable request.

Materials Availability Not applicable.

Code Availability The code can be obtained from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no known conflicting financial interests or personal ties that might have influenced the work presented in this study.

Ethical Approval and Consent to Participate Not applicable.

Consent for Publication All authors have given their consent for the publication of this work.

References

1. Bruno A, Singh G. Personality traits prediction from text via machine learning. In: 2022 IEEE World Conference on Applied Intelligence and Computing (AIC), 2022:588–594. https://doi.org/10.1109/AIC55036.2022.9848937
2. Kamalesh MD, Bharathi B. Analysing the personality traits using machine learning techniques. In: 2022 International Conference on Innovative Computing, Intelligent Communication and Smart

Electrical Systems (ICSES), 2022:1–6. https://doi.org/10.1109/ICSES55317.2022.9914112
3. Sivakumar S, Priyanka S, Selvam P. Enhancing personality type prediction with ensemble models: A robust predictive approach. In: 2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), 2023:1–7. https://doi.org/10.1109/ICSES60034.2023.10465452
4. Beltrán VMB, Cabada RZ, Estrada MLB, López HMC, Escalante HJ. A multiplatform application for automatic recognition of personality traits for learning environments. In: 2022 International Conference on Advanced Learning Technologies (ICALT), 2022:49–50. https://doi.org/10.1109/ICALT55010.2022.00022.
5. Bousalem S, Benchikha F, Marir N. Personalized learning through mbti prediction: a deep learning approach integrated with learner profile ontology. IEEE Access. 2025;13:30861–73. https://doi.org/10.1109/ACCESS.2025.3542701.
6. Dimmita N, Shaik AA, Sree P, Chinthapalli P. Automatic personality recognition in interviews using cnn. In: 2023 4th IEEE Global Conference for Advancement in Technology (GCAT), 2023:1–7. https://doi.org/10.1109/GCAT59970.2023.10353423.
7. Kamble P, Kulkarni U. An innovative approach of personality recognition for e-recruitment. In: 2022 5th International Conference on Advances in Science and Technology (ICAST), 2022:324–328. https://doi.org/10.1109/ICAST55766.2022.10039605.
8. Neria Y. Neria, Yuval, pp. 1–3. Springer, Cham 2016. https://doi.org/10.1007/978-3-319-28099-8_2026-1.
9. Thapa L, Pandey A, Gupta D, Deep A, Garg R. A framework for personality prediction for e-recruitment using machine learning algorithms. In: 2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2024:1–5. https://doi.org/10.1109/Confluence60223.2024.10463354.
10. Ait Baha T, El Hajji M, Es-saady Y, Fadili H. The power of personalization: a systematic review of personality-adaptive chatbots. SN Computer Science 2023;4: https://doi.org/10.1007/s42979-023-02092-6.
11. Chen OT-C, Tsai C-H, Ha M-H. Automatic personality recognition via xlnet with refined highway and switching module for chatbot. In: 2024 IEEE International Symposium on Circuits and Systems (ISCAS), 2024:1–5. https://doi.org/10.1109/ISCAS58744.2024.10558116.
12. Vinciarelli A, Mohammadi G. More personality in personality computing. IEEE Trans Affect Comput. 2014;5(3):297–300. https://doi.org/10.1109/TAFFC.2014.2341252.
13. Xue D, Hong Z, Guo S, Gao L, Wu L, Zheng J, et al. Personality recognition on social media with label distribution learning. IEEE Access. 2017;5:13478–88. https://doi.org/10.1109/ACCESS.2017.2719018.
14. Subramanian R, Wache J, Abadi MK, Vieriu RL, Winkler S, Sebe N. Ascertain: Emotion and personality recognition using commercial sensors. IEEE Trans Affect Comput. 2018;9(2):147–60. https://doi.org/10.1109/TAFFC.2016.2625250.
15. Wang Z, Wu C, Zheng K, Niu X, Wang X. Smotetomek-based resampling for personality recognition. IEEE Access. 2019;7:129678–89. https://doi.org/10.1109/ACCESS.2019.2940061.
16. Kosinski M, Matz SC, Gosling SD, Popov V, Stillwell D. Facebook as a research tool for the social sciences: Opportunities, challenges, ethical considerations, and practical guidelines. Am Psychol. 2015;70(6):543–56. https://doi.org/10.1037/a0039210.
17. Aung ZMM, Myint PH. Personality prediction based on content of facebook users:a literature review. In: 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2019:34–38. https://doi.org/10.1109/SNPD.2019.8935692.
18. KN, Gavrilova PK, ML. Latent personality traits assessment from social network activity using contextual language embedding. IEEE Transactions on Computational Social Systems. 2022;9(2):638–49. https://doi.org/10.1109/TCSS.2021.3108810.
19. Zhang L, Peng S, Winkler S. Persemon: a deep network for joint analysis of apparent personality, emotion and their relationship. IEEE Trans Affect Comput. 2022;13(1):298–305. https://doi.org/10.1109/TAFFC.2019.2951656.
20. Ontoum S, Chan JH. Personality Type Based on Myers-Briggs Type Indicator with Text Posting Style by using Traditional and Deep Learning 2022. https://arxiv.org/abs/2201.08717.
21. Sandra L, Prabowo H, Gaol FL, Isa SM. Personet: a novel framework for personality classification-based apt customer service agent selection. IEEE Access. 2024;12:25200–14. https://doi.org/10.1109/ACCESS.2024.3364352.
22. Liao R, Song S, Gunes H. An open-source benchmark of deep learning models for audio-visual apparent and self-reported personality recognition. IEEE Trans Affect Comput. 2024;1–18. https://doi.org/10.1109/TAFFC.2024.3363710.
23. Kwon N, Yoo Y, Lee B. Novel curriculum learning strategy using class-based tf-idf for enhancing personality detection in text. IEEE Access. 2024;12:87873–82. https://doi.org/10.1109/ACCESS.2024.3417180.
24. Ryan G, Katarina P, Suhartono D. Mbti personality prediction using machine learning and smote for balancing data based on statement sentences. Information 2023;14(4): https://doi.org/10.3390/info14040217.
25. Pradnyana GA, Anggraeni W, Yuniarno EM, Purnomo MH. Enhancing mbti personality trait prediction from imbalanced social media data using hybrid query expansion ranking and glove- bilstm. In: 2023 IEEE International Conference on Fuzzy Systems (FUZZ), 2023:1–6. https://doi.org/10.1109/FUZZ52849.2023.10309718.
26. Lin H. Dlp-personality detection: a text-based personality detection framework with psycholinguistic features and pre-trained features. Multimedia Tools and Applications. 2023;83:1–20. https://doi.org/10.1007/s11042-023-17015-z.
27. Ghods V. Personality recognition based on handwriting types using fuzzy inference. IEEE Access. 2023;11:86456–69. https://doi.org/10.1109/ACCESS.2023.3303477.
28. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks 2019. https://arxiv.org/abs/1801.04381.
29. Chollet F. Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017:1251–8.
30. Kaggle: MBTI Personality Types Dataset. Accessed: 2024-11-16 (2017). https://www.kaggle.com/datasets/datasnaek/mbti-type
31. Radford A, Wu J, Amodei D, Clark J, Brundage M, Sutskever I. Language models are unsupervised multitask learners. OpenAI. 2019;
32. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux M-A, Lacroix T, Rozière B, Goyal N, Hambro E, Azhar F, Rodriguez A, Joulin A, Grave E, Lample G. LLaMA: open and Efficient Foundation Language Models 2023. https://arxiv.org/abs/2302.13971.
33. Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, Casas D, Bressand F, Lengyel G, Lample G, Saulnier L, Lavaud LR, Lachaux M-A, Stock P, Scao TL, Lavril T, Wang T, Lacroix T, Sayed WE. Mistral 7B 2023. https://arxiv.org/abs/2310.06825.
34. Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y. BERTScore: evaluating Text Generation with BERT 2020. https://arxiv.org/abs/1904.09675.
35. Chen L-C, Papandreou G, Schroff F, Adam H. Rethinking Atrous Convolution for Semantic Image Segmentation 2017. https://arxiv.org/abs/1706.05587.
36. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL. Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans

Pattern Anal Mach Intell. 2018;40(4):834–48. https://doi.org/10.1109/TPAMI.2017.2699184.
37. Loshchilov I, Hutter F. SGDR: Stochastic Gradient Descent with Warm Restarts 2017. https://arxiv.org/abs/1608.03983.
38. Yu F, Koltun V. Multi-Scale Context Aggregation by Dilated Convolutions 2016. https://arxiv.org/abs/1511.07122.
39. Kim Y. Convolutional Neural Networks for Sentence Classification 2014. https://arxiv.org/abs/1408.5882.
40. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. MobileNets: efficient Convolutional Neural Networks for Mobile Vision Applications 2017. https://arxiv.org/abs/1704.04861.
41. Waghumbare A, Singh U, Kasera S. Diat-dscnn-eca-net: separable convolutional neural network-based classification of galaxy morphology. Astrophys Space Sci. 2024;369: https://doi.org/10.1007/s10509-024-04302-w.
42 Koluguri NR, Li J, Lavrukhin V, Ginsburg B. SpeakerNet: 1D depth-wise separable convolutional network for text-independent speaker recognition and verification, 2020. https://arxiv.org/abs/2010.12653
43 Balderas L, Lastra M, Ben´ıtez JM. Optimizing convolutional neural network architectures. Mathematics. 2024:12(19), 3032. https://doi.org/10.3390/math12193032
44 Luo P, Wang X, Shao W, Peng Z. Towards Understanding Regularization in Batch Normalization 2019. https://arxiv.org/abs/1809.00846
45 Santurkar S, Tsipras D, Ilyas A, Madry A. How Does Batch Normalization Help Optimization? 2019. https://arxiv.org/abs/1805.11604
46 Mastromichalakis S. ALReLU: a different approach on Leaky ReLU activation function to improve Neural Networks Performance 2021. https://arxiv.org/abs/2012.07564
47 Devlin J, Chang M-W, Lee K, Toutanova K. Bert: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, 2019. https://doi.org/10.18653/v1/N19-1423
48 Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. Roberta: a robustly optimized bert pretraining approach. 2019. https://doi.org/10.48550/arXiv.1907.11692