Information Processing and Management 60 (2023) 103217

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

A novel personality detection method based on high-dimensional psycholinguistic features and improved distributed Gray Wolf Optimizer for feature selection

Hao Lin, Chundong Wang *, Qingbo Hao

School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300384, China
Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin University of Technology, Tianjin 300384, China
Engineering Research Center of Learning-Based Intelligent System, Ministry of Education, Tianjin 300384, China

ARTICLE INFO

Dataset link: https://github.com/ml-papers-coders/Keras-BigFive-personality-traits/, https://www.kaggle.com/datasets/datanaek/mbit-type, https://github.com/minekirito/DLP-Personality-Prediction/tree/main/data/essays

Keywords:
Personality detection
Feature selection
Symmetric uncertainty
Grey Wolf Optimizer
Spark

ABSTRACT

Existing personality detection methods based on user-generated text have two major limitations. First, they rely too much on pre-trained language models to ignore the sentiment information in psycholinguistic features. Secondly, they have no consensus on the psycholinguistic feature selection, resulting in the insufficient analysis of sentiment information. To tackle these issues, we propose a novel personality detection method based on high-dimensional psycholinguistic features and improved distributed Gray Wolf Optimizer (GWO) for feature selection (IDGWOFS). Specifically, we introduced the Gaussian Chaos Map-based initialization and neighbor search strategy into the original GWO to improve the performance of feature selection. To eliminate the bias generated when using mutual information to select features, we adopt symmetric uncertainty (SU) instead of mutual information as the evaluation for correlation and redundancy to construct the fitness function, which can balance the correlation between features–labels and the redundancy between features–features. Finally, we improve the common Spark-based parallelization design of GWO by parallelizing only the fitness computation steps to improve the efficiency of IDGWOFS. The experiments indicate that our proposed method obtains average accuracy improvements of 3.81% and 2.19%, and average F1 improvements of 5.17% and 5.8% on Essays and Kaggle MBTI dataset, respectively. Furthermore, IDGWOFS has good convergence and scalability.

1. Introduction

Personality is a stable psychological construction that has been associated with thoughts, emotions, and behaviors of people. Any research field in information science and computer science that involves the understanding, prediction, and synthesis of human behavior, such as human–computer interaction (Shumanov & Johnson, 2021), recommender system (Aguiar, Fechine, & Costa, 2020), rumor spreading analysis (Acharya, Aryan, Saha, & Ghosh, 2022), mental illness diagnosis (Majaluoma, Seppala, Kautiainen, & Korhonen, 2020), cyber security management (Moustafa, Bello, & Maurushat, 2021) may benefit from personality detection. Personality detection is a burgeoning field at the intersection of psychology, information science and computer science (Phan & Rauthmann, 2021). Traditional manual measurement approaches of personality, such as Self-report Inventory, are widely used by psychology scholars, but gradually abandoned by computer science scholars due to their low efficiency and ecological validity. The

* Correspondence to: School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300384, China.
E-mail address: michael3769@tjut.edu.cn (C. Wang).

https://doi.org/10.1016/j.ipm.2022.103217
Received 17 September 2022; Received in revised form 9 November 2022; Accepted 28 November 2022
Available online 16 December 2022
0306-4573/

rejections spawned machine learning-based methods of automatic personality detection, which dominate today. In particular, the textual data-based detection method is the cornerstone of other sophisticated detection methods and the most practical because the textual data related to the subjects are easily accessible compared to audio (Principi, Palmero, Junior, & Escalera, 2021), images (Jeremy, Christian, Kamal, Suhartono, & Suryaningrum, 2021), and electroencephalogram (Li et al., 2020).

But till now, there is very little research work on the feature selection of textual data-based detection methods. For one thing, early methods relied on certain psycholinguistic features such as LIWC (Linguistic Inquiry and Word Count), MRC (MRC Psycholinguistic database), SenticNet emotional dictionary, etc. The values of psycholinguistic features are estimated on the basis of morphology and emotion of tokens in the analyzed texts (Devyatkin et al., 2017). Some examples of psycholinguistic features are: verb–adjective ratio, number of the pronouns in the first person singular form, emotional tendency (positive/negative) of words, etc. However, existing methods adopt few psycholinguistic features, resulting in the insufficient analysis of sentiment information, and there is no consensus on selection for psycholinguistic features. Psycholinguistic features are high-dimensional, that is, the number of features (variables observed) is close to or larger than the number of observations (or data points). If we use all the features for a typical text classification task, we may get poor results because some redundant features are not helpful for classification and some features may mislead the classifiers. In general, fewer features will get the high efficiency but low accuracy, while more features will only improve the accuracy to some extent (Wang, Yao, & Liu, 2019). Therefore, feature selection must be applied to eliminate noisy, less informative, and redundant features, to reduce the feature space to a manageable level, thus improving efficiency and accuracy of the classifiers used. For another thing, the existing textual data-based personality detection methods rely too much on pre-trained language models of transfer learning, such as BERT (Bidirectional Encoder Representations from Transformer), XLNet, and RoBERT (A Robustly Optimized BERT) to omit key psycholinguistic features. Individual differences in linguistic utilization have been considered as reflections of psychological phenomena since the early times of Freud (Mehta et al., 2020). The choice of words is driven not only by meaning, but also by psychological phenomena such as emotions, relational attitudes, power status, and personality traits (Tausczik & Pennebaker, 2010). So, these psycholinguistic features are also significant for personality detection and higher model interpretability.

The remainder is organized as follows: The research objectives are presented in Section 2. Section 3 introduces some previous related works and preliminaries about personality prediction. Section 4 demonstrates the personality detection method proposed by us on three levels. Section 5 demonstrates the key algorithm in our proposed method — IDGWOFS. Section 6 conducts analysis of experiments and results in detail. Section 7 discusses the implications and potential practical applications of our work. Finally, the paper ends with a discussion of limitations and future works.

2. Research objectives and contributions

In this work, we aim to improve the performance of machine learning-based personality detection methods, focusing on how to use traditional psycholinguistic features to improve the performance. Therefore, we propose a novel method for automatic personality detection, which consists of three main steps: (1) preprocessing step; (2) feature extraction step; (3) classification detection step. Based on this method, we focus on the following research questions (RQ):

- **RQ1** Is fusion of pre-trained language features with high-dimensional psycholinguistic features effective for personality detection?
- **RQ2** How to select high-dimensional psycholinguistic features for better detection performance?
- **RQ3** How to improve the efficiency of high-dimensional psycholinguistic feature selection?

Noteworthy, in the feature extraction step, we concatenate pre-trained language features with multiple psycholinguistic features and prove that the fusion of the two is effective by ablation experiments and comparison with the existing methods. Heuristic algorithms led by GWO with mutual information as the evaluation for the correlation between features are often used for feature selection. To improve the performance and efficiency of feature selection, we propose a novel method called Improved Distributed Grey Wolf Optimizer for Feature Selection (IDGWOFS). Specifically, the initial solution generation of the original GWO is improved based on the Gaussian Chaotic Map. A neighbor search strategy is introduced to enhance the global and local search capabilities of GWO. More importantly, to eliminate the bias generated when using mutual information to select features, we adopt symmetric uncertainty (SU) instead of mutual information as the evaluation to construct a novel fitness function that can balance the correlation between features–labels and the redundancy between features–features. Finally, although the heuristic algorithm has higher selection efficiency than other algorithms, we still hope to further reduce the selection time. So, the parallel design of the proposed feature selection method is carried out by using Spark.

The contributions of our paper are given as follows:

- This paper intends to introduce a novel personality detection method based on Bi-LSTMs with attention mechanism, multi-feature fusion, and a new distributed feature selection algorithm (IDGWOFS). To the best of our knowledge, this is the first time that such high-dimensional psycholinguistic features and feature selection method for them has been employed.
- To balance the correlation between features–labels and the redundancy between features–features, a new fitness function of IDGWOFS is proposed based on SU. Compared to the original GWO, IDGWOFS has improved initial solution generation and extra neighbor search strategy.
- To improve the efficiency of IDGWOFS, the common Spark-based parallelization design is improved and IDGWOFS is parallelized based on the new parallelization design. To the best of our knowledge, IDGWOFS is the first distributed SU-based feature selection method.

Table 1
The Big Five Model.

Personality traits	Feature
Openness	Imagination, adventure, curiosity
Extraversion	Passion, vitality, dominance
Neuroticism	Anxiety, anger, impulsiveness
Conscientiousness	Rationality, responsibility, self-discipline
Agreeableness	Trust, honesty, obedience

Table 2
The Myers–Briggs Type Indicator.

Dimension	Personality traits
Direction of attention	Extrovert or Introvert
Cognitive style	Intuition or Sensing
Judgment	Thinking or Feeling
Lifestyle	Judging or Perceiving

- The convergence of IDGWOFS is proved based on Markov chain.
- The effectiveness, scalability, and advancement of our proposed detection method are demonstrated on two public datasets including Essays and Kaggle MBTI.

3. Related works and preliminaries

3.1. Personality taxonomies

Throughout the lifespan of personality research, numerous taxonomies have been proposed to describe human personality traits. At present, the most representative and frequently used taxonomy is the Big Five Model (Big 5), which is shown in Table 1 (Stajner & Yenikent, 2020). Big 5 is constructed by the lexical method and describes the individual's personality from five personality traits: Neuroticism (NEU), Extraversion (EXT), Openness (OPN), Agreeableness (AGR), and Conscientiousness (CON).

Myers–Briggs Type Indicator (MBTI) attempts to assign personality traits into four categories: introversion or extraversion (E/I), sensing or intuition (S/N), thinking or feeling (T/F), judging or perceiving (J/P). The MBTI emphasizes naturally occurring differences and indicates people's differing psychological priority in perceiving world and making decisions. Table 2 presents a detailed explanation of MBTI. One letter from each personality trait is taken to generate a personality type, such as "INFP".

The detection of each personality trait can be regarded as a binary classification problem, which is the mainstream personality detection mode. In addition, other personality taxonomies such as Minnesota Multiple Personality Inventory (MMPI), sixteen personality factor questionnaire (16PF), and Eysenck Personality Questionnaire (EPQ) are widely used in psychology. However, due to the lack of relevant public datasets, there is no relevant personality detection research.

3.2. Personality detection

One of the early efforts in personality detection was proposed by Argamon, Koppel, and Pennebaker (2005). The words in corpora were grouped into four categories with psychological meaning: function, cohesion, assessment, and appraisal. The detection task was performed with a SVM, whose input was the frequencies of the words appearing in each category. Mairesse, Walker, Mehl, and Moore (2007) used the same corpora and SVM, but extra adopted LIWC and MRC psycholinguistic features to achieve an average accuracy of 57%. Poria, Gelbukh, Agarwal, Cambria, and Howard (2013) proposed a more sophisticated detection method whose inputs were LIWC, MRC, and SenticNet features. Moreover, these features were used to build a SMO classifier.

The success of BERT in NLP has led researchers to pay more attention to pre-trained language models in personality detection tasks since 2018. Mehta et al. (2020) reported their results on Essays and Kaggle MBTI dataset with two pre-trained language models including BERT-base and BERT-large. They believe that their model consisting of BERT and MLP (Multi-Layer Perception) dominated the detection of the Big 5 and MBTI personality traits and the features extracted by pre-trained language models consistently beat psycholinguistic features. Wang et al. (2021) proposed a novel classifier for personality detection from textual data with Capsule Networks and XLNet. Jiang, Zhang, and Choi (2020) presented a novel approach to automatic personality detection using RoBERT and attentive neural networks for the Big 5. Their model improves the SOTA results on the Essays dataset by 2.49%. Likewise, Vasquez and Ochoa-Luna (2021) proposed a personality detection approach with RoBERT for MBTI. Furthermore, Pabon and Arroyave (2022) adopted three pre-trained language models — Word2Vec, GloVe, and BERT to classify Big 5 personality traits. El-Demerdash, El-Khoribi, Ismail, and Abdou (2021) used three pre-trained models including Elmo, ULMFiT, and BERT to extract features and achieved SOTA results on the myPersonality dataset.

In recent years, few research have used the combination of psycholinguistic features and pre-trained linguistic features for personality detection. Yuan, Wu, Li, and Wang (2018) combined the LIWC features and the features extracted by Word2Vec to build a detection model. Pavan and Gavrilova (2022) concatenated Term Frequency–Inverse Document Frequency (TF–IDF), features extracted by GloVe and statistical features of social applications, and used SVM and RF as classifiers to identify MBTI personality traits. Kazameini, Fatehi, Mehta, Eetemadi, and Cambria (2020) concatenated features extracted by BERT with the Mairesse features, which are made up of LIWC, MRC, prosodic and utterance-type features. They fed these features to multiple SVMs to detect personality traits in parallel like a bagging classifier. Similarly, Ren, Shen, Diao, and Xu (2021) leveraged BERT and SenticNet 5 features to detect personality from textual data.

The above works only use pre-trained language models or few psycholinguistic features additionally and do not pay attention to the psycholinguistic feature selection, resulting in the insufficient analysis of sentiment information. Individual differences in linguistic utilization have been considered as reflections of psychological. The psycholinguistic features have the same significance for personality detection as pre-trained language models and more interpretability. Apart from Mairesse features and SenticNet 5 features, other psycholinguistic features such as NRC Emotion Lexicon features, NRC Valence, Arousal, and Dominance (VAD) Lexicon features, Hourglass of Emotions features, and text readability features have been proved to relate to personality traits by correlation analysis or factor analysis. These features should be given more attention for personality detection.

3.3. Feature selection methods

Feature selection methods are usually classified into three categories: filter, wrapper, and embedded (Kumar & Sonajharia, 2014; Pintas, Fernandes, & Garcia, 2021). Filter methods are executed as a previous step and are independent of the learning activity. Wrapper methods encapsulate the classifier and utilize the performance of the classifier to evaluate the relevance of features and search for the best feature subset. Embedded methods include the feature selection methods as part of the training process. Since the filter and embedded methods involve the training process, applying them to deep learning-based classifiers will seriously reduce the training efficiency.

The filter methods, which are the focus of this paper, are adopted by most applications that require feature filtering, due to their simplicity and efficiency. Multiple literatures have reported that SU is one of the best evaluation metrics in the filter methods (Dai, Chen, Liu, & Hu, 2020; Song, Kang, Sun, & He, 2018; Yang & Li, 2021). For example, Wang et al. (2019) ranked features by the SU and then selected features with the genetic algorithm. Further, there is still a lack of an algorithm that can efficiently and accurately solve the Np-hard problem in discrete space, which is how to search the optimal feature subset according to the SU between features–labels and features–features.

3.4. Gray wolf optimizer

GWO is a nature-simulated metaheuristic algorithm, which was proposed based on an internal leadership hierarchy and group behavior of the grey wolves (Nadimi-Shahraki, Taghian, & Mirjalili, 2020). The internal hierarchy divides all grey wolves into α wolf (optimal solution), β wolf (second best solution), δ wolf (third best solution) and ω wolves. The search solution process of GWO is guided by α, β, and δ wolves in each iteration. The objective function is optimized by simulating the surrounding, hunting, and attacking behavior of wolves.

Surrounding the prey by the wolves can be modeled as

$$X(t+1) = X_p(t) - A \times D, \tag{1}$$

$$D = \mid C \times X_p(t) - X(t) \mid . \tag{2}$$

Where X indicates the position vector of a wolf, X_P indicates the position vector of the surrounded prey, t indicates the current iteration. C and A indicate the coefficient vectors which can be calculated by

$$A = 2\lambda \times r_1 - \lambda, \tag{3}$$

$$C = 2r_2. \tag{4}$$

Where r_1, r_2 are random number in $[0, 1]$. λ is called the distance control parameter and linearly decreases from 2 to 0 during the iterations.

$$\lambda = 2 - 2t/MaxIter. \tag{5}$$

Hunting the prey can be modeled as

$$X_1(t) = X_\alpha(t) - A_{t1} \times D_\alpha(t), \tag{6}$$

$$X_2(t) = X_\beta(t) - A_{t2} \times D_\beta(t), \tag{7}$$

$$X_3(t) = X_\delta(t) - A_{t3} \times D_\delta(t). \tag{8}$$

Determined by the leadership hierarchy, $X_\alpha(t), X_\beta(t), X_\delta(t)$ have better knowledge about the prey. A_{t1}, A_{t2}, A_{t3} are calculated by Eq. (3).

$$D_\alpha(t) = \mid C_1 \times X_\alpha(t) - X(t) \mid, \tag{9}$$

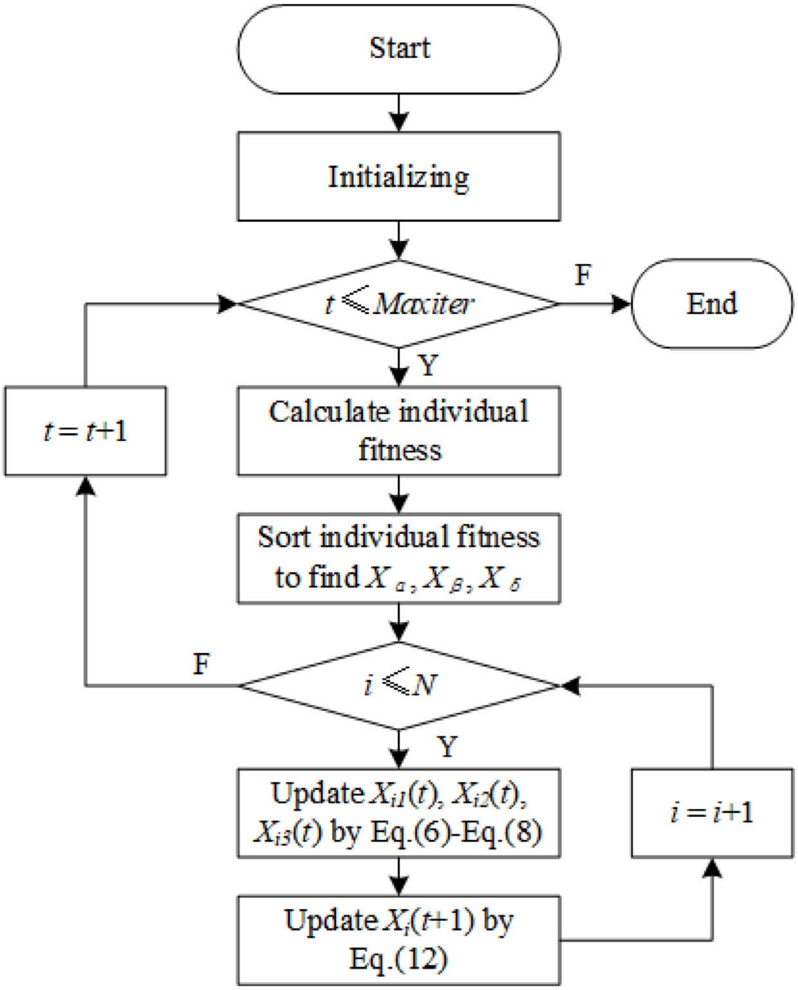


Fig. 1. Process of GWO.

$$D_{\beta}(t) = | C_2 \times X_{\beta}(t) - X(t) |, \tag{10}$$

$$D_{\delta}(t) = | C_3 \times X_{\delta}(t) - X(t) | . \tag{11}$$

Where C_1, C_2, and C_3 can be calculated by Eq. (4). Each ω wolf moves around three leading wolves. This behavior can be modeled as

$$X(t+1) = (X_1(t) + X_2(t) + X_3(t))/3. \tag{12}$$

In fact, the initial wolfs need to be randomly generated in the solution space and the location of the grey wolf is calculated by the fitness function. In conclusion, GWO has several advantages such as it is simple, easy to use, having fewer hyperparameters, and having an excellent switching mechanism between exploration and exploitation processes. The applications of GWO belong to the domains of global optimization, power engineering, bioinformatics, environmental applications, machine learning, networking and image processing, etc (Faris, Aljarah, Al-Betar, & Mirjalili, 2017). The flowchart of the original GWO is shown in Fig. 1.

3.5. Spark

In the current era of Big Data, the Spark framework has been widely used for large-scale machine learning training (Lou et al., 2021; Niu, Zheng, Fournier-Viger, & Wang, 2021). Spark is a MapReduce-like parallel computing engine designed for big data processing. Moreover, Spark is based on memory computing, which is more suitable for iterative optimization algorithms than the outdated MapReduce.

The key of Spark is a unique data format called Resilient Distributed Datasets (RDD), which can be processed in parallel on multiple nodes. RDD supports two types of operations: Transformation and Action. Transformation is to map an RDD into a new RDD, and Action will calculate with the RDD to return a result. Of course, these operations are performed in parallel.

4. Proposed methodology

As shown in Fig. 2, the proposed overall scheme is divided into a preprocessing step; a feature extraction step, and a classification detection step.

4.1. Preprocessing

Personality detection can be regarded as a combination of multiple binary classifications. Therefore, it is necessary to encode the personality traits into 0,1. In addition, data augmentation can help train more accurate and robust models, particularly when using smaller datasets. So, we augmented the training dataset with Easy Data Augmentation (EDA) (Wei & Zou, 2019), which includes synonym replacement, random insertion, random replacement, and random delete.

In order to extract psycholinguistic features accurately, we perform Text Surface Transformation (TST) on each sample before using the above operations. TST can expand the contractions, such as from "gimme" to "give me", to accurately count the word frequency.[1] Finally, we binary-coded the personality traits into 0,1 because personality detection can be regarded as multiple binary-classification problems.

4.2. Feature extraction

Benefiting from previous research, apart from the common Mairesse and SenticNet 5 features, we additionally adopt four psycholinguistic features: NRC Emotion Lexicon (Mohammad & Turney, 2013), NRC VAD Lexicon (Mohammad, 2018), Affectivespace (Chaturvedi, Satapathy, Cavallari, & Cambria, 2019), and Readability. The introduction of these psychological characteristics is as follows:

- The Mairesse has a set of psycholinguistic features consisting of LIWC, MRC, prosodic and utterance-type features. We abandoned prosodic features and finally adopted a total of 79 features. These are the most widely used features in traditional machine learning-based personality trait mining.
- The NRC Emotion Lexicon[2] has a lexicon of over 14,000 English words which are annotated with values of emotions such as anger, anticipation, disgust, etc. The final value of this sub-feature is the means of all values of emotionally charged words present in the text data.
- The NRC VAD Lexicon[3] has a lexicon of over 20,000 English words which are annotated with their valence, arousal, and dominance scores. As above, the VAD Lexicon value is the means of all constituent words in the text data.
- The Affectivespace[4] is a vector space of affective common sense available for English and has 100,000 concepts.
- The Readability[5] has a number of calculated readability measures which are based on simple surface characteristics of the text data. These measures are basically linear regressions based on the number of words, syllables, and sentences.
- The SenticNet 5[6] is a tool used for extracting common sense knowledge along with associated sentiment polarity and affective labels from the text data, including pleasantness value, attention value, sensitivity value, aptitude value, and polarity value (Cambria, Poria, Hazarika, & Kwok, 2018).

There may be redundancies between multiple psycholinguistic traits. It will affect detection model performance and training efficiency. Therefore, we filter the above psycholinguistic features by proposed IDGWOFS, which will be introduced separately in the next section.

The effectiveness of pre-trained language models for personality detection has been demonstrated in much previous literature. Benefiting from our previous research, we adopt ALBERT (A Lite BERT) to extract pre-trained language features of Big 5 and adopt BERT to extract pre-trained language features of MBTI. The experimental results are reported in Appendix A. BERT, a multi-layer bidirectional Transformer encoder with bidirectional self-attention, is pre-trained using large corpus of texts, including BooksCorpus and English Wikipedia. The bidirectional self-attention ensures that the output of BERT is obtained through weight distribution. ALBERT reduces the number of parameters in BERT through the factorization of the embedding parameters, without compromising performance (Lynnette & Carley, 2022). All psycholinguistic features will be normalized. Finally, the pre-trained language features concatenate with the filtered normalized psycholinguistic features by the early fusion method to achieve feature fusion. The advantage of early fusion methods is that the features are directly fused. The process is simple, the time-consuming is short, and it is more suitable for the efficient method proposed by us.

1 https://github.com/kootenpv/contractions
2 http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm
3 http://saifmohammad.com/WebPages/nrc-vad.html
4 http://sentic.net/downloads
5 pypi.org/project/readability
6 http://sentic.net/api

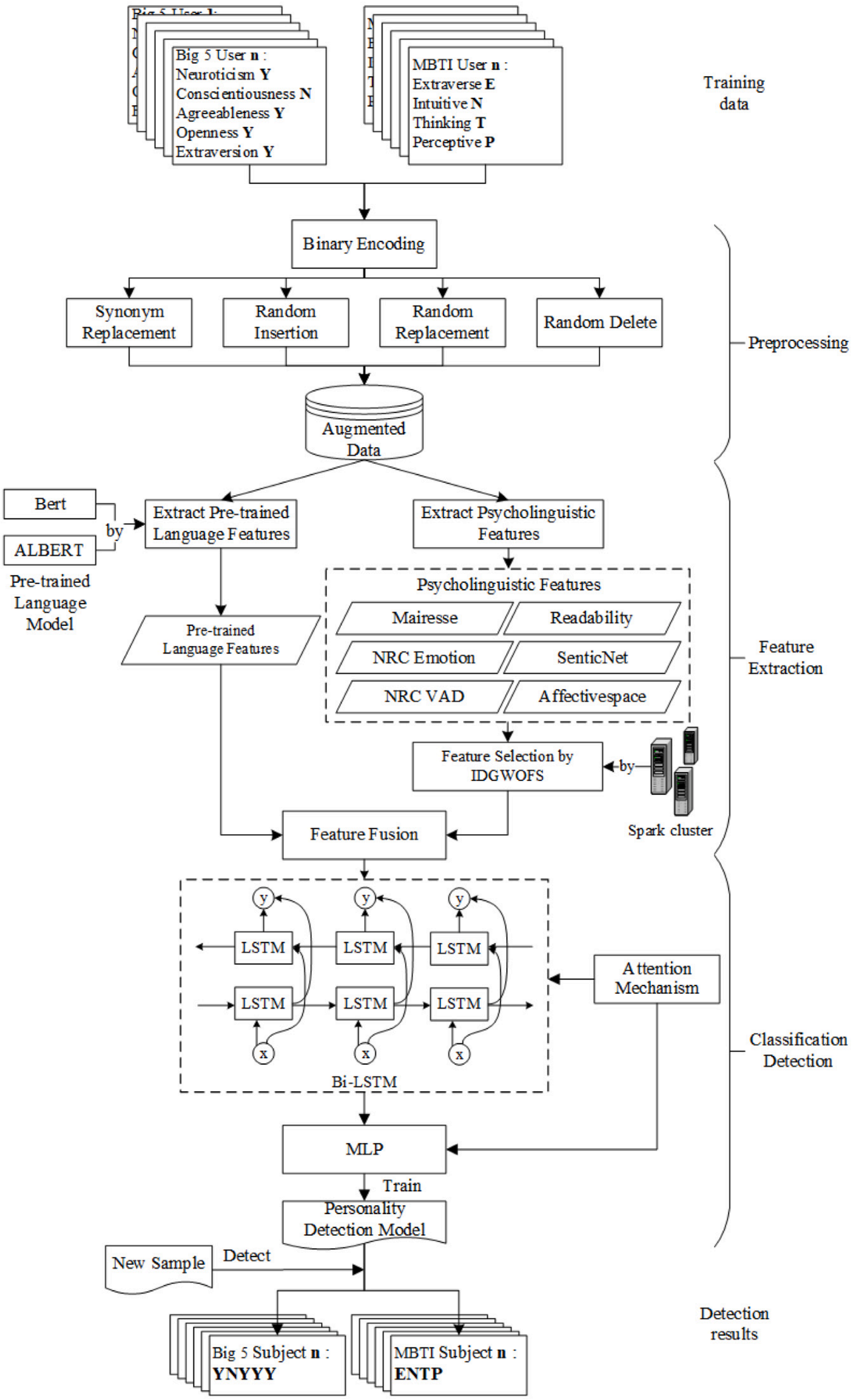


Fig. 2. Scheme of proposed personality automatic detection.

4.3. Classification detection

We build a two-layer Bi-LSTM with the many-to-one attention mechanism as a classifier. Bi-LSTM is improved from LSTM (Zheng & Chen, 2021), which uses three gates to achieve information storage, namely

$$f_t = sigmoid(W_f[h_{t-1}, x_t] + b_f), \tag{13}$$

$$i_t = sigmoid(W_i[h_{t-1}, x_t] + b_i), \tag{14}$$

$$o_t = sigmoid(W_o[h_{t-1}, x_t] + b_o). \tag{15}$$

Where W represents the weight of each gate, and b represents the offset. Generally, a Bi-LSTM contains a forward layer and a backward layer. $\overrightarrow{h_t}$ indicates the hidden output sequence of the forward layer and $\overleftarrow{h_t}$ indicates the hidden output of the backward layer. Then, the output of current LSTM cell C_t and $\overrightarrow{h_t}$ can be calculated as

$$C_t = f_t C_{t-1} + i_t \tanh(W_C[h_{t-1}, x_t] + b_C), \tag{16}$$

$$\overrightarrow{h_t} = o_t \times \tanh(C_{t-1}). \tag{17}$$

At each time t, x_t is the current input and the Bi-LSTM calculates the whole output h_t as

$$h_t = sigmoid(W_h[\overrightarrow{h_t}, \overleftarrow{h_t}] + b_h). \tag{18}$$

Additionally, we also experiment with SVM, Logistic Regression (LR), Random Forest (RF), MLP, Long short-term memory (LSTM), and multi-layer Bi-LSTM while fine-tuning, however, it results in no evident performance boost.

We adopt the attention to weighted fuse the output vectors to make the contribution distribution more reasonable. The attention function adopted by us is

$$Attention(h_t, \overline{h_s}) = \frac{exp(Dot(h_t, \overline{h_s}))}{\sum_{s'=1}^{s} exp(Dot(h_t, \overline{h_s}))}. \tag{19}$$

We adopt the Dot function as Score function. Additionally, we experiment with more complex attention mechanisms, such as Scaled Dot-product Attention, Bahdanau Attention, and Multi-Head Attention, yet they result in a performance drop.

The output results of the Bi-LSTM are input to a Dense layer with a Sigmoid activation function to normalize the personality trait results. If some result values are greater than or less than the probability threshold 0.5, they are considered to belong to a category.

5. Improved distributed grey wolf optimizer for feature selection

The difference between the proposed IDGWOFS and the original GWO lies in four parts: initial solution generation, fitness function, neighbor search strategy, and parallelization.

5.1. Initial solution generation

In the feature selection, the solution space is $X_i \in [0, 1]$. If $X_i \geq 0.5$, the feature f_i will be selected. If $X_i < 0.5$, the feature f_i will be not selected. Instead of random numbers, we use the Gaussian Chaotic Map to generate the initial solutions. It has the following two advantages:

- Gaussian Chaotic Map can generate more evenly distributed initial solutions and improve the diversity of the solutions (Ma & Sun, 2022).
- Compared with the commonly used Tent Chaotic Map (Ling et al., 2022), Gaussian Chaotic Map can generate more solutions for $X_i < 0.5$ and make the algorithm have faster convergence speed. That is, fewer features are selected in the initial solutions, resulting in faster computation of the fitness function.

The following equations define Gaussian Chaotic Map:

$$X_{i+1} = \begin{cases} 0, X_i = 0 \\ \frac{1}{X_i \cdot \mathrm{mod} 1} = \frac{1}{X_i} - \left[\frac{1}{X_i}\right], otherwise. \end{cases} \tag{20}$$

The comparison between the initial solution generated based on random number and Gaussian Chaotic Map is shown in Fig. 3.

5.2. Symmetric uncertainty-based fitness function

The correlation between feature–label is a publicly known measure on whether a feature is important. Many studies used mutual information to measure its importance, because it is a correlation measure based on entropy (Dai et al., 2020). Suppose the complete feature set is $F = \{f_1, f_2, f_3, \dots, f_n\}$, $H(F)$ is the information entropy of F. The higher $H(F)$ is, the more information the features carry. $H(F)$ can be expressed as

$$H(F) = -\sum_{f_i \in F} P(f_i) \log_2 P(f_i). \tag{21}$$

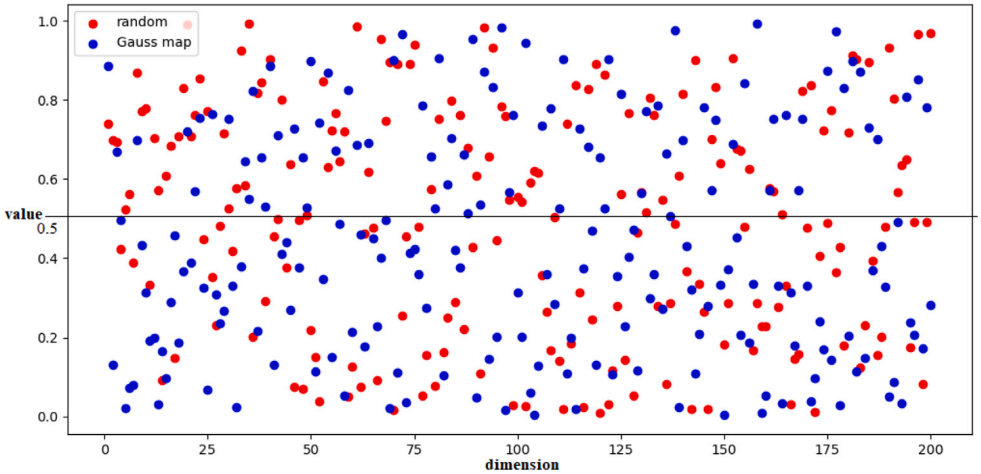


Fig. 3. The comparison of the distribution of the two initial solution generation methods. The proportion of red dots below 0.5 is about 52% and that of blue dots is about 55%. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The relative conditional entropy $H(F_i \mid F_j)$ represents the uncertainty of F_i when the F_j is known. $H(F_i \mid F_j)$ can be expressed as

$$H(F_i \mid F_j) = -\sum_{f_j \in F_j} P(f_i) \sum_{f_i \in F_i} P(f_i \mid f_j) \log_2 P(f_i \mid f_j). \tag{22}$$

Where $i, j \in \{0, 1, \ldots, n\}$. Moreover, the mutual information $I(F_i; F_j)$, calculated by using $H(F) - H(F_i \mid F_j)$, is often used for feature selection in machine learning. When the possible value of F_j is much larger than that of F_i, the conditional probability becomes smaller (Wang et al., 2019). Seemingly, the larger $I(F_i; F_j)$ indicates that the F_i and F_j are highly close. However, the correlation between the F_i and F_j may be low. So, we adopt the SU instead of the mutual information. The SU can be defined as

$$SU_{F_iF_j} = 2 \times \frac{H(F) - H(F_i \mid F_j)}{H(F_i) + H(F_j)}. \tag{23}$$

The SU normalizes the mutual information, which corrects the bias when selecting features using mutual information. When $SU_{F_iF_j}$ is 0, it means that F_i and F_j are totally independent. When $SU_{F_iF_j}$ is 1, it indicates that F_i can perfectly predict F_j.

To obtain a feature subset with high correlation between features–labels and low redundancy between features–features, we propose the following fitness function for the feature selection of personality detection.

$$fitness = \frac{n\overline{SU_{f_ilabel}}}{\sqrt{n(n-1)\overline{SU_{f_if_j}}}}. \tag{24}$$

Eq. (24) balances the correlation between features–labels and the redundancy between features–features. The numerator of Eq. (24) measures the correlation, and the denominator measures the redundancy. Obviously, as the fitness increases, the corresponding feature subset becomes better.

To improve the computational efficiency of IDGWOFS, the SU between pairwise features is first calculated to construct a symmetric uncertainty matrix $Matrix_{SU}$. When calculating the fitness function of IDGWOFS, read the SU directly from the constructed $Matrix_{SU}$, which avoids double calculation. The pseudocode for generating $Matrix_{SU}$ is shown in Appendix B.

5.3. Neighbor search strategy

In the original GWO, α, β, and δ wolve leads all ω wolves toward the search space where it is promising that the optimal solution will be found. However, the behavior may reduce the population diversity in the later stage, resulting in only getting the locally optimal solutions. In the real world, in addition to group search, grey wolves have another behavior pattern called individual search. So, we increase population diversity by simulating the behavior of individuals obtaining hunting information from their neighbors.

In the neighbor search strategy, grey wolves will learn hunting experience from nearby grey wolves. The schematic of the group search strategy and the neighbor search strategy are shown in Figs. 4 and 5.

The position of the wolf i in the iteration t is represented as $X_i(t) = \{X_{i1}, X_{i2}, \ldots, X_{id}\}$. d indicates the dimension of the problem, that is, the number of variables in the problem. N indicates the individual number of the wolve group. Then the whole wolve group can be recorded as a matrix $Wolves$ with N rows and d columns.

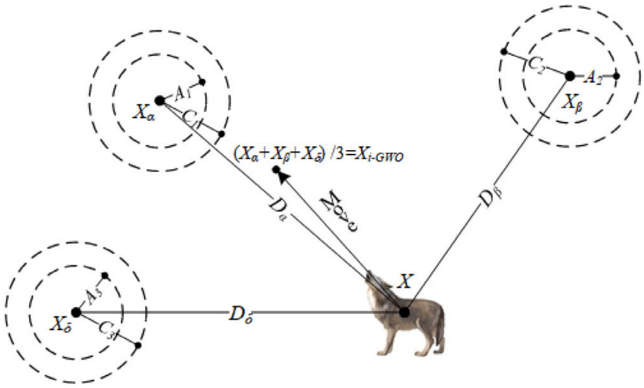


Fig. 4. Group search strategy. Where, D indicates the distance between two wolves. C and A indicate the coefficient vectors, same as Eqs. (3) and (4). X_{i-GWO} indicates the result obtained by group search strategy.

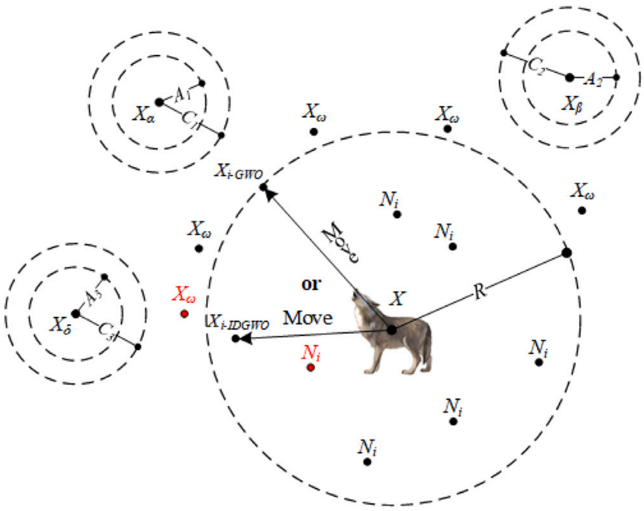


Fig. 5. Neighbor search strategy. Where, N_i indicates the nearby wolves of X. $X_{i-IDGWO}$ indicates the result obtained by neighbor search strategy.

The nearby wolves of $X_i(t)$ denoted by $N_i(t)$ can be constructed as

$$N_i(t) = \{X_j(t) \mid Euclidean(X_i(t), X_j(t)) \le R_i(t), X_j(t) \in Woloves, j \in [1, d]\}. \tag{25}$$

Where $Euclidean(X_i(t), X_j(t))$ is Euclidean distance between $X_i(t)$ and $X_j(t)$. $R_i(t)$ respects the hunting radius of wolf i. $R_i(t)$ can be defined as

$$R_i(t) = Euclidean(X_i(t), X_{i-GWO}(t+1)). \tag{26}$$

Where $X_{i-GWO}(t+1)$ is the result of the group search strategy according to Eq. (12). Then, the new position derived by the neighbor search strategy can be calculated as

$$X_{i-IDGWO}(t+1) = X_i(t) + r_3 \times (X_n(t) - X_r(t)), X_n(t) \in N_i(t), X_r(t) \in Woloves. \tag{27}$$

Where $X_n(t)$ is randomly sampled from $N_i(t)$ and $X_r(t)$ is randomly sampled from $Woloves$. r_3 is a random number in (0,1].

The results of neighbor search strategy should be compared with the results of the group search strategy, that is,

$$X_i(t+1) = \begin{cases} X_{i-GWO}, & fitness(X_{i-GWO}) < \\ & fitness(X_{i-IDGWO}) \\ X_{i-IDGWO}, & fitness(X_{i-GWO}) \ge \\ & fitness(X_{i-IDGWO}). \end{cases} \tag{28}$$

The neighbor search strategy is an additional strategy that is further optimized based on the results of the group search strategy. After performing the neighbor search strategy for all wolves, t is increased by one.

5.4. Parallelization

Although metaheuristic algorithms such as IDGWOFS can obtain a satisfactory feasible solution in a limited time, their efficiency can be greatly optimized. In order to reduce the computation time of IDGWOFS, we design a parallelized IDGWOFS based on Spark.

The existing parallel design of heuristics algorithms is to parallelize all key steps, including the update of solutions (e.g. mutation and crossover in GA, the update of particle velocity in PSO, the update of wolf position in GWO) and the computation of fitness function (e.g. MSE, MAPE, Eq. (25)) (Chen et al., 2019; Chen, Tu, & Xu, 2021; Tadist, Mrabti, Nikolov, Azeddine, & Said, 2021).

However, the design is not suitable for IDGWOFS. When the parallelized algorithm is started, the Spark cluster takes some time for basic operations such as starting jobs, dividing tasks, and allocating resources (Lin, Lin, Wan, Wang, & Gao, 2021). If the calculation in Executor of Spark cluster is less, the parallelization performance of the Spark cluster is poor.

Through previous experiments, we found that the computation time of the fitness function occupies almost 99% of the entire running time of IDGWOFS. Moreover, as N increases, the final solution of IDGWOFS becomes more accurate, and the above proportion becomes larger. Therefore, we design parallelization of IDGWOFS only against the fitness function. The pseudocode of parallelized IDGWOFS is shown in Algorithm 1.

Algorithm 1 Parallelized IDGWOFS

Input: $F = \{f_1, f_2, f_3, \dots, f_n\}$, N, $MaxIter$, Spark cluster parameters$conf$
Output: Optimal feature subsetF_{best}

1: Create SparkContext object sc using $conf$
2: Generate N initial solutions pop according to Eq. (20) and record them as pop.
3: Convert pop to RDD data format rdd_{pop} using *sc.paralleize*().
4: Compute the fitness of each solution in rdd_{pop} in parallel using *map*(*getFitness*()).
5: Sort the fitness of all solutions in pop and return them to fitness list $fitness$ using *collect*().
6: $tmp = 0$
7: **while** $tmp \geq MaxIter$ **do**
8:     Find α wolf, β wolf, and δ wolf according to $fitness$.
9:     Update pop according to Eqs. (6–12) and(25–28).
10:     Convert pop to RDD data format rdd_{pop} using *sc.paralleize*().
11:     Compute the fitness of each new solution in rdd_{pop} in parallel using *map*(*getFitness*()).
12:     Sort the fitness of all new solutions in pop and return them to fitness list $fitness$ using *collect*().
13:     Find the minimum one in $fitness$ and record it as $fitness_{best}$.
14:     tmp = tmp + 1
15: **end while**
16: Extract the best feature subset F_{best} from F according to $fitness_{best}$
17: **End algorithm**

In Algorithm 1, the solution set generated by IDGWOFS is converted into RDD through the parallelize(), and the purpose is to use the map() to evaluate the solution set in parallel, that is, to calculate the individual fitness in parallel. The collect function triggers the Transformation operations that have not been performed due to the lazy computing characteristics of spark. When the maximum number of iterations is met, the optimal subset of feature set found is returned. The map function on lines 4 and 11 is the core of the whole algorithm. The function can cause each element in a RDD to be processed by another specified function, resulting in a new RDD. The pseudocode of the getFitness() is shown in Appendix C.

In summary, the overall process of IDGWOFS is shown in Fig. 6. We provide a convergence analysis of IDGWOFS in Appendix D.

6. Experiments and results

6.1. Data

We adopt the publicly available Essays and Kaggle MBTI datasets in our experiments. Some of recent research still uses them (Ren et al., 2021). Essays, a scientific gold standard in psychology for personality detection, consists of 2468 student essays annotated with the Big 5 personality traits which were identified by a standardized Self-report Inventory. We augment the Essays dataset with 430 new samples, all of which belong to the minority class. It can alleviate the problem of sample imbalance in the Essays dataset and significantly improves the detection performance. Kaggle MBTI contains tweets posted by 8675 users and were labeled MBTI personality traits by a Self-report Inventory. The feature sets constructed from the above two datasets are shown in Table 3. Since data augmentation will affect the calculation of information entropy, we use the unaugmented dataset for feature selection experiments and the augmented dataset for model training.

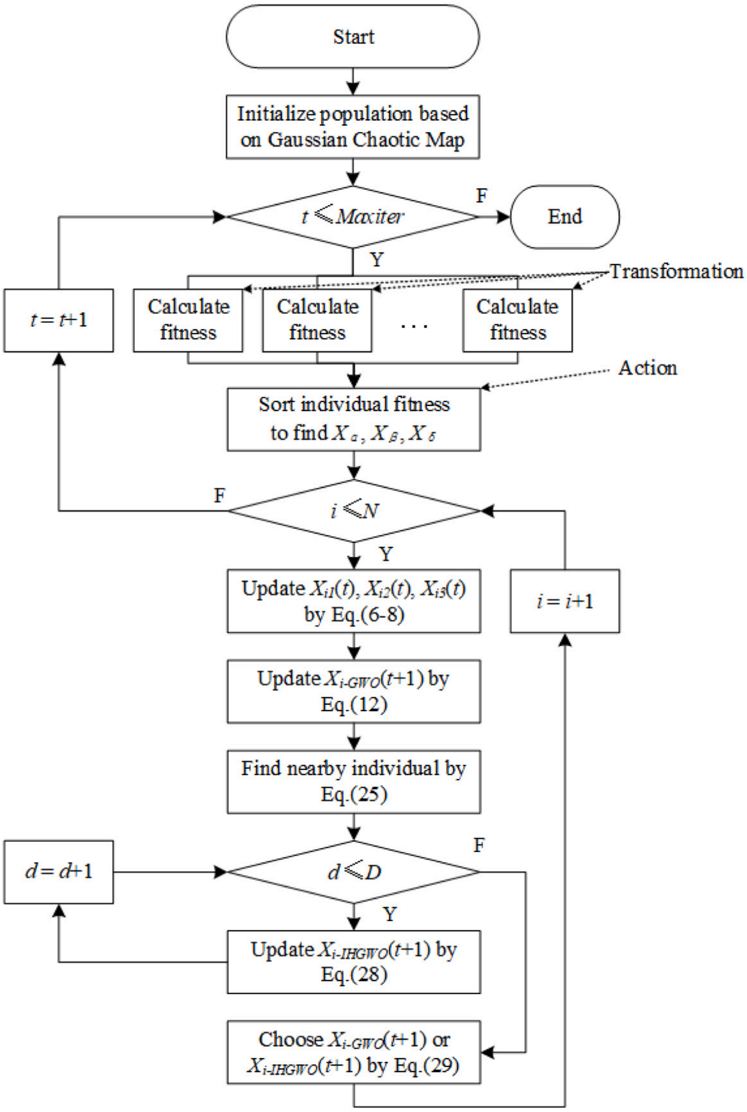


Fig. 6. Process of IDGWOFS.

Table 3
Experimental datasets.

Dataset	Sample	Feature	Dimension
Essays	2468 (2898)	Mairesse, NRC Emotion Lexicon, NRC VAD Lexicon, Affectivespace, Readability, SenticNet	228
kaggle MBTI	8675	Mairesse, NRC Emotion Lexicon, NRC VAD Lexicon, Readability	123

6.2. Environment

We conduct experiments using the high-performance computing cluster (HPCC) provided by Beijing ChinaHPC Technology Co., Ltd. (ChinaHPC). We build an eight-node spark cluster on the HPCC. The detailed parameters of Spark are as follows: spark.executor.pyspark.memory=2G, spark.executor.cores=1, num-executors=4, spark.driver.cores=1, spark.driver. memory=1G, spark.python.worker.memory=1G.

Table 4
Compare experiment of multiple heuristic algorithm for feature selection.

Algorithm	Essays				Kaggle MBTI			
	Best	AVG	SD	Dimension	Best	AVG	SD	Dimension
GA	0.4150	0.4174	0.0021	118	0.1978	0.2022	**0.0021**	60
GOA	0.4114	0.4164	0.0035	112	0.2081	0.2107	0.0022	63
SSA	0.3942	0.3997	0.0038	102	0.1838	0.1932	0.0067	57
MVO	0.4158	0.4166	**0.0014**	106	0.1822	0.1892	0.0049	56
GWO	0.3501	0.3604	0.0073	62	0.0821	0.1101	0.0272	22
IDGWOFS	**0.2355**	**0.2878**	0.0409	**57**	**0.0409**	**0.0821**	0.0265	**18**

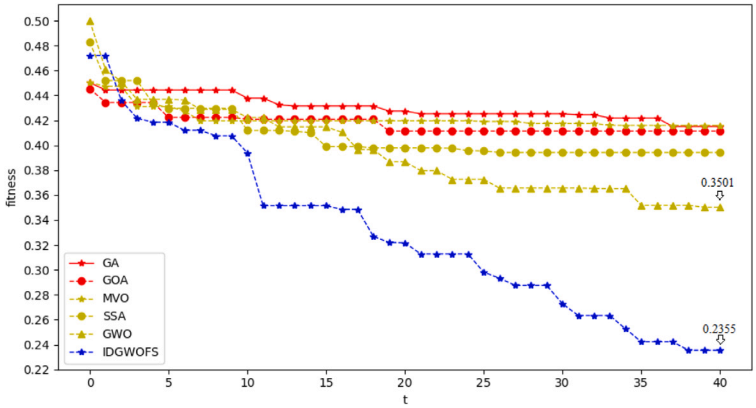


Fig. 7. Optimal fitness convergence curves of heuristics on Essays.

6.3. Feature selection experiments

Due to the No Free Lunch theorem of the optimization algorithm (Wolpert & Macready, 1997), we cannot directly choose feature Selection algorithms for personality detection. So, we compare IDGWOFS with other heuristic algorithms to prove the advantage of IDGWOFS in feature selection. The heuristic algorithms include: GA with the elite retention strategy, the original GWO, Grasshopper Optimization Algorithm (GOA) (Saremi, Mirjalili, & Lewis, 2017), Multi-Verse Optimizer (MVO) (Mirjalili, Mirjalili, & Hatamlou, 2016), and Salp Swarm Algorithm (SSA) (Mirjalili et al., 2017).

In order to balance the advantage of the neighbor search strategy of IDGWOFS, the population size of IDGWOFS is set to 5, and that of other heuristic algorithms is set to 10. MaxIter of all algorithms is set to 40. Each algorithm is run 10 times. Other hyperparameters include: 0.1 mutation probability of GA; 0.5 crossing probability of GA; 0.5 attraction strength parameter of GOA; 1.5 attraction scale parameter of GOA; decrease coefficient of GOA decreases linearly from 1 to 0.00004; decrease coefficient of SSA decreases linearly from 2 to 0; wormhole existence probability of MVO increases linearly from 0.2 to 1; shuttle distance ratio of MVO decreases from 0.6 to 0; λ of GWO and IDGWOFS decreases linearly from 2 to 0. The experimental results are shown in Table 4.

As shown in Table 4, the best fitness achieved by IDGWOFS beats the second best fitness achieved by 48.66% and the average fitness achieved by IDGWOFS beats the second best average fitness by 25.22% in the Essays dataset. In the Kaggle MBTI dataset, the best fitness achieved by IDGWOFS beats the second best fitness achieved by 100.7% and the average fitness achieved by IDGWOFS beats the second best average fitness by 34.1%.

In addition, the standard deviation achieved by IDGWOFS is slightly higher than other algorithms, but it is not enough to affect the convergence stability of IDGWOFS. The fitness curves of the optimal results of each algorithm are shown in Figs. 7 and 8.

As shown in Figs. 7 and 8, except for IDGWOFS, other algorithms fall into a long-term local optimum. In Fig. 7, these algorithms often fall into local optima at iteration 15–20. In Fig. 8, these algorithms often fall into local optima at iteration 15–25. Benefiting from the Gaussian Chaos Map and neighbor search strategy, IDGWOFS can usually break through the local optimum within 4 iterations. To sum up, IDGWOFS exhibits the best convergence accuracy, convergence speed, and convergence ability.

Figs. 9 and 10 show the cross-entropy matrix and SU matrix of the feature set of the Kaggle MBTI dataset filtered by IDGWOFS. The difference in cross-entropy between features–features in Fig. 9 is too large. Apparently, there is no such bigotry in Fig. 10. In addition, most SUs in Fig. 10 are less than 0.5, which is consistent with the idea of minimizing redundancy between features–features in the fitness function proposed by us. The SU matrix corresponding to the Essays dataset is similar to Fig. 10. But the order of the matrix is too large to be illustrated.

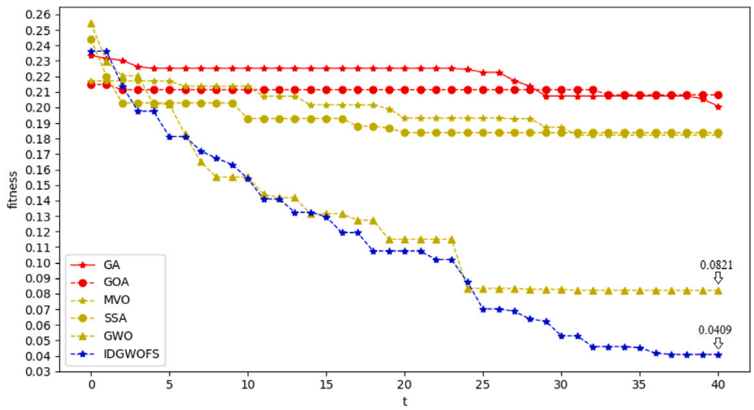


Fig. 8. Optimal fitness convergence curves of heuristics on Kaggle MBTI.

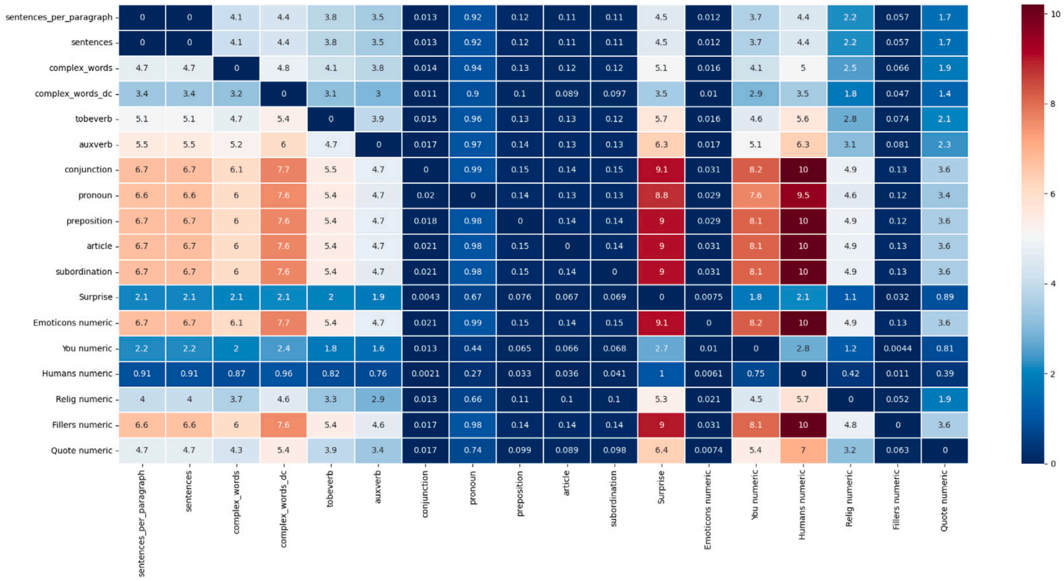


Fig. 9. Conditional entropy Matrix for selecting features using IDGWOFS on Kaggle MBTI. The redder the figure as a whole, the more correlated the selected features are, that is, the more redundant there is.

To verify the effectiveness of selected features, we input the best feature subset obtained by IDGWOFS and other algorithms into the same network to train detection models. Furthermore, we report aggregated 10 fold cross-validation performance of the outer re-sampling loop. The batch-size is 64 and the learning rate is 0.0003. For Essays, the epoch is 25. For Kaggle MBTI, the epoch is 50. The adopted optimizer is Adam with a binary cross entropy loss. The structure of the adopted network is shown in Fig. 11.

The effectiveness and advancement of SU-based feature selection have been proven in many literatures (Dai et al., 2020; Yang & Li, 2021). The experimental results are shown in Tables 5–8. Most models trained with filtered features outperformed the models trained with all features in accuracy and F1. It demonstrates that the effectiveness of SU as an evaluation for feature selection. Except for "NEU", "AGR", and "T/F" personality traits, IDGWOFS achieved the highest accuracy and F1 on both datasets. In addition, due to the lower dimension of the feature subsets obtained by IDGWOFS, the training and testing time of associated models is also less.

6.4. Scalability experiment of IDGWOFS

The scalability experiment is used to test whether the algorithm efficiency can be improved by adding additional nodes. We perform IDGWOFS 10 times each on a single node (stand-alone), 2 nodes, 4 nodes, and 8 nodes to record the time-consuming and

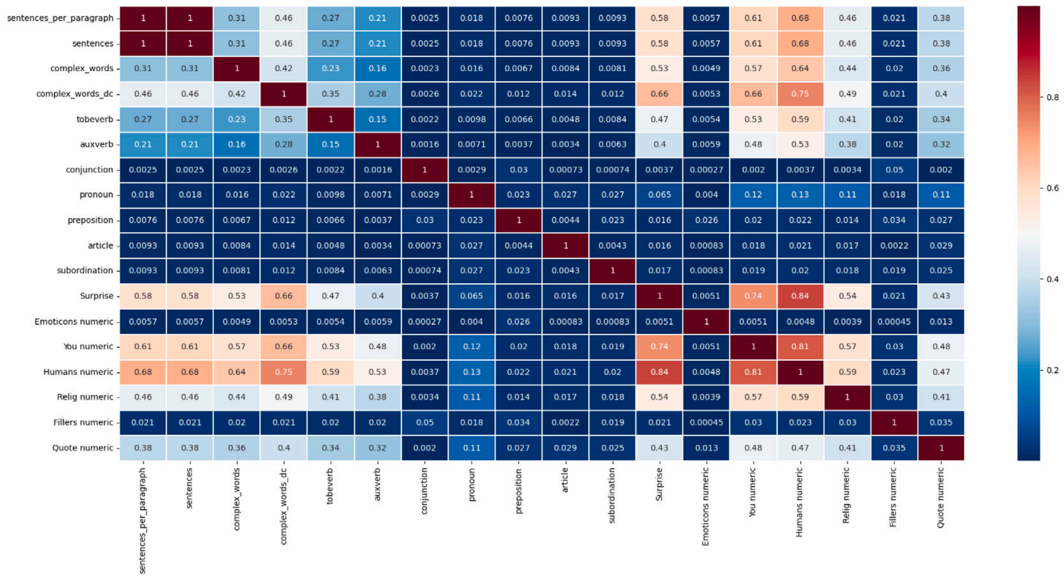


Fig. 10. SU Matrix for selecting features using IDGWOFS on Kaggle MBTI. Similar to Fig. 9, the redder the figure, the more redundant there is.

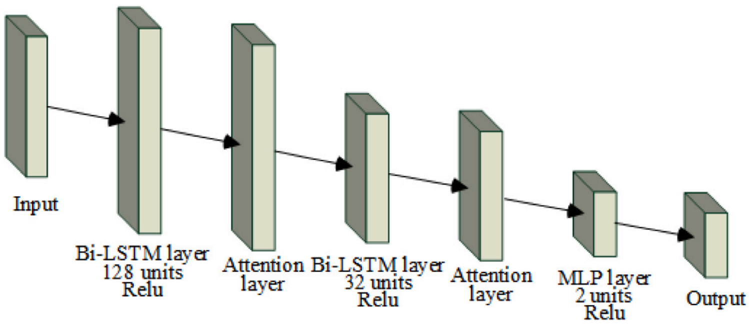


Fig. 11. Structure of classifier.

Table 5
Evaluation results of Essays' feature selection with accuracy.

Algorithm	Big 5				
	EXT	NEU	AGR	CON	OPN
Not select	69.16	70.43	63.68	71.996	72.55
GA	72.88	70.58	67.15	73.07	73.02
GOA	72.86	70.66	66.99	72.78	73.13
SSA	73.12	**72.56**	67.87	74.42	77.44
MVO	74.79	70.80	71.58	74.52	77.18
GWO	75.81	71.76	**74.46**	74.15	74.29
IDGWOFS	**75.97**	71.74	73.17	**76.77**	**77.74**

the speedup of IDGWOFS. The experimental dataset is Kaggel MBTI. Speedup is defined as follows:

$$SpeedUp = T_1/T_n. \quad (29)$$

Table 6
Evaluation results of Essays' feature selection with F1.

Algorithm	Big 5				
	EXT	NEU	AGR	CON	OPN
Not select	59.98	57.99	57.93	58.97	59.59
GA	63.71	63.86	60.73	66.198	62.81
GOA	64.26	64.44	61.297	65.79	63.32
SSA	67.02	65.94	62.497	67.38	67.62
MVO	67.64	66.55	63.11	67.99	68.22
GWO	67.80	66.87	66.72	67.91	67.81
IDGWOFS	**69.79**	**67.89**	**69.84**	**70.90**	**69.86**

Table 7
Evaluation results of Kaggle MBTI's feature selection with accuracy.

Algorithm	MBTI			
	I/E	N/S	T/F	P/J
Not select	77.66	86.37	72.49	62.78
GA	77.79	86.23	74.68	64.56
GOA	77.59	86.21	74.62	64.26
SSA	79.08	86.97	77.12	69.83
MVO	78.78	86.90	**77.27**	69.68
GWO	79.33	86.98	76.08	69.85
IDGWOFS	**79.51**	**87.09**	77.24	**71.87**

Table 8
Evaluation results of Kaggle MBTI's feature selection with F1.

Algorithm	MBTI			
	I/E	N/S	T/F	P/J
Not select	73.53	86.06	72.43	62.99
GA	72.28	84.29	71.31	62.49
GOA	72.24	84.25	71.27	62.52
SSA	72.85	84.63	71.63	64.51
MVO	73.55	85.36	72.88	66.697
GWO	77.31	86.74	75.22	66.56
IDGWOFS	**78.68**	**87.39**	**77.34**	**70.36**

Where, T_1 is the sequential calculation time. T_n is parallel calculation time based on n nodes. Ideally, the speedup is equal to the number of nodes.

The results of scalability experiment are shown in Fig. 12.

The size of N on the abscissa determines the calculation of IDGWOFS. As shown in Fig. 12(a), with the gradual increase of the calculation, the running time of IDGWOFS gradually increases. When the calculation is less, the running time difference of IDGWOFS on different numbers of nodes is small. The reason is that basic operations in the Spark cluster take lots of time.

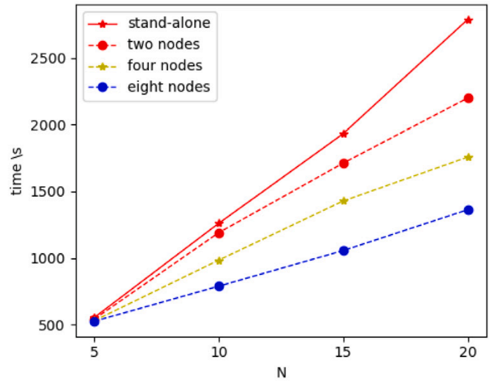
As shown in Fig. 12(b), when the calculation is less, the speedup of IDGWOFS is low, the acceleration of Spark cluster is not significant due to the basic operations of the Spark cluster. With the gradual increase of the calculation, the advantages of proposed parallel design become more and more obvious, and the speedup of IDGWOFS gradually tends to the ideal value. In summary, the experimental results verify that IDGWOFS has good parallelism and scalability.

6.5. Ablation experiment

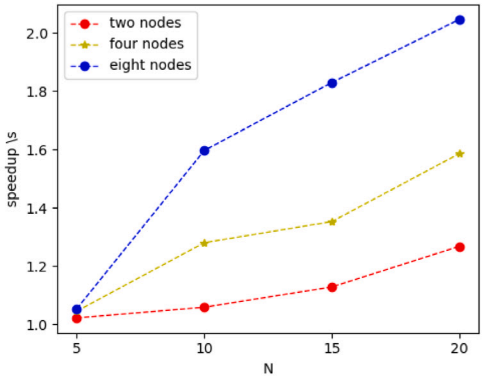
The purpose of the ablation experiment is to prove the necessity of the key parts of the proposed personality detection method. The key parts include:

- P: Pre-trained language model features
- L: Psycholinguistic features
- SU: Selecting psycholinguistic features using IDGWOFS

The classifier structure and parameters adopted in this experiment are the same as in Fig. 11. The results of the ablation experiments are shown in Tables 9–12.



(a) Time cost of IDGWOFS under different circumstances.



(b) Speedup of IDGWOFS under different circumstances. The ideal value of speedup is equal to the number of nodes.

Fig. 12. Results of scalability experiment.

Table 9
Ablation experiment on Essays with accuracy.

Method	Big 5				
	EXT	NEU	AGR	CON	OPN
L	69.16	70.43	63.68	71.996	72.55
L+SU	71.82	72.92	65.26	66.898	75.90
P+L	74.91	**73.66**	70.35	76.72	77.71
P+L+SU	**75.97**	71.74	**73.17**	**76.77**	**77.74**

Table 10
Ablation experiment on Essays with F1.

Method	Big 5				
	EXT	NEU	AGR	CON	OPN
L	59.98	57.99	57.93	58.97	59.59
L+SU	61.01	60.21	59.14	60.03	62.24
P+L	58.00	56.21	56.14	57.03	58.20
P+L+SU	**69.79**	**67.89**	**69.84**	**70.90**	**69.86**

Table 11
Ablation experiment on Kaggle MBTI with accuracy.

Method	MBTI			
	I/E	N/S	T/F	P/J
L	77.66	86.37	72.49	62.78
L+SU	77.52	86.40	73.11	62.45
P+L	78.89	86.74	**77.53**	69.37
P+L+SU	**79.51**	**87.09**	77.24	**71.87**

Table 12
Ablation experiment on Kaggle MBTI with F1.

Method	MBTI			
	I/E	N/S	T/F	P/J
L	73.53	86.06	72.43	62.99
L+SU	76.37	86.24	72.43	62.65
P+L	72.57	85.56	71.86	62.096
P+L+SU	**78.68**	**87.39**	**77.34**	**70.36**

Table 13
The comparison between existing research and our research on Big 5 with accuracy.

Method	Big 5				
	EXT	NEU	AGR	CON	OPN
BERT-MLP	60	60.5	58.8	59.2	64.6
RoBERT-MLP	60.62	61.07	59.72	58.55	65.86
Bagged-SVM	59.3	59.4	56.5	57.8	62.1
CNN-AdaBoost	61.85	62.08	59.92	64.93	60.56
BERT-fusion	61.15	62.2	60.8	59.52	65.6
BERT-SenticNet	71.8*	70.16*	72.68*	**77.13***	70.16*
Ours	**75.97**	**71.74**	**73.17**	76.77	**77.74**

* means current state-of-the-art result.

Table 14
The comparison between existing research and our research on Big 5 with F1.

Method	Big 5				
	EXT	NEU	AGR	CON	OPN
BERT-MLP	58	56.21	56.14	57.03	58.2
RoBERT-MLP	58.4	56.37	56.87	56.2	59
Bagged-SVM	58.3	57.8	56.2	57.4	59.9
CNN-AdaBoost	60.8	61.4	60.2	64.2	60.7
BERT-fusion	61.03	59.5	58.37	59.45	61.78
BERT-SenticNet	66.86*	63.48*	67.13*	70.02*	63.97*
Ours	**69.79**	**67.89**	**69.84**	**70.90**	**69.86**

The results show that, except for the "NEU" and "T/F" personality traits, the accuracy obtained by using "P+L+SU" is better than the accuracy obtained by using other key parts. For all personality traits, the F1 obtained by using "P+L+SU" is better than the F1 obtained by using other key parts. In summary, the key parts involved in the proposed method are all necessary.

6.6. Compare with the existing methods

In order to validate the performance of proposed method, we compare our proposed method with the current SOTA methods on the Essays and the Kaggle MBTI dataset. These methods include:

- **BERT-MLP** represents a MLP model using features extracted by BERT-base (for Big 5) and BERT-large (for MBTI) (Mehta et al., 2020).
- **RoBERT-MLP** represents a MLP model using features extracted by RoBERT (Jiang et al., 2020).
- **Bagged-SVM** represents a model using features extracted by BERT and Mairesse features to feed to Bagged-SVM (Kazameini et al., 2020).
- **CNN-AdaBoost** represents a model with features obtained from various filters of CNN are fed to an AdaBoost (Mohades, Sadr, & Tarkhan, 2022).
- **BERT-fusion** is a model using both data and classifier level fusion. The adopted features are extracted with three pre-trained language models including ELMo, ULMFiT, and BERT (El-Demerdash et al., 2021).
- **BERT-SenticNet** is a model using the pre-trained BERT model and a neural network. The adopted features are extracted with BERT and SenticNet 5 (Ren et al., 2021).
- **LSTM-RMSprop** represents a LSTM model using the RMSprop optimizer (Mawadatul & Hilman, 2021).
- **Transformer-MD** represents a Multi-Document Transformer model with a dimension attention mechanism to focus each personality dimension on the relevant post (Yang, Quan, Yang, & Yu, 2021).
- **TrigNet-GAT** represents a graph network that injects structural psycholinguistic knowledge in LIWC (Yang, Yang, Ouyang, & Quan, 2021).

We give preference to citing the experimental results in the above papers, and if not, we reproduce their model with the hyperparameters we optimized. The comparison results between existing research and our research are shown in Tables 13–16.

Our proposed personality detection method achieves the state-of-the-art accuracy on both datasets. In the Essays dataset, the accuracy achieved by ours beats the current state-of-the-art accuracy by 3.81% and the F1 achieved by ours beats the current state-of-the-art F1 by 5.17%. In the Kaggle MBTI dataset, the accuracy achieved by ours beats the current state-of-the-art accuracy by 2.19% and the F1 achieved by ours beats the current state-of-the-art F1 by 5.8%.

Table 15
The comparison between existing research and our research on Kaggle MBTI with accuracy.

Method	MBTI			
	I/E	N/S	T/F	P/J
LSTM-RMSprop	77.35	86.34	72.85	66.28
Bagged-SVM	79.0*	86.0	74.2	65.4
BERT-MLP	78.8	86.3	76.1	67.2
RoBERT-MLP	77.73	86.42	73.71	63.24
BERT-SenticNet	78.42	78.54	77.5	71.35*
Transformer-MD	76.69	86.45*	**78.21***	67.98
TrigNet-GAT	77.43	86.37	78.07	68.06
Ours	**79.51**	**87.09**	77.24	**71.87**

Table 16
The comparison between existing research and our research on Kaggle MBTI with F1.

Method	MBTI			
	I/E	N/S	T/F	P/J
LSTM-RMSprop	61.24	67.68	68.21	58.79
Bagged-SVM	56.67	52.85	75.42	65.94
BERT-MLP	68.05	79.35	66.1	55.73
RoBERT-MLP	58.33	53.88	69.36	60.88
BERT-SenticNet	68.92	80.58*	67.33	66.54
Transformer-MD	66.08	69.10	**79.19***	67.50
TrigNet-GAT	69.54*	67.17	79.06	67.69*
Ours	**78.68**	**87.39**	77.34	**70.36**

7. Discussion

Implications of results. The above result analysis show that our personality detection model works better than the state-of-the-art model. Importantly, it will lead personality detection research to avoid the misunderstanding of over-reliance on pre-trained language models and not needing feature selection.

Universality of approach. Although IDGWOFS has only been proved effective on the datasets related to personality detection, we believe that IDGWOFS can also achieve good results on datasets in other fields after parameter tuning.

Implications of work. The work also has significant implications for practice that involves the understanding, prediction, and synthesis of human behavior. Accurate personality detection can provide psychologists with an alternative to crowdsourcing to collect large amounts of research data. Accurate detection of the personality traits of social media users can support a variety of personalized downstream tasks, such as recommendation of information, information seeking, driving behavior analysis, corporate management, human–computer interaction, etc.

Moreover, personality detection is not the end of the personality calculation. The results of personality detection should be further analyzed to make the personality detection model really useful in research and life. For example: personality transfer caused by major events (Acharya et al., 2022), the relationship between personality traits and cyberspace security behavior (Shappie, Dawson, & Debb, 2020), and human-centered rumor research.

8. Conclusion

We propose a novel personality detection method, whose performance is verified by multiple experiments. Our results show that the method consistently beats the current state-of-the-art on the Essays and Kaggle MBTI dataset with a less complex classification network structure.

The limitations of our research and future works are as follows:

- "Multimodal Learning" (Huang et al., 2021) must be the future of personality detection. The multimodal training of personality detection models with multi-source heterogeneous data such as images, audio, video, social software and even EEG is our future work.
- In practice, for each personality trait, the features can be screened separately. Based on this, multiple detection models for personality traits can be trained. If we do this, the results achieved by our method will significantly outperform all baseline models.

• We will research the parallel design of feature selection under large feature dimensions based on mix multiple parallel mechanisms, and investigate whether late fusion and hybrid fusion methods can improve the performance of our detection model.

CRediT authorship contribution statement

Hao Lin: Conceptualization, Methodology, Software, Data curation, Visualization, Writing – original draft, Writing – review & editing. **Chundong Wang:** Investigation, Supervision, Writing – review. **Qingbo Hao:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The Essays and Kaggle MBTI datasets are downloaded from "https://github.com/ml-papers-coders/Keras-BigFive-personality-traits/" and "https://www.kaggle.com/datasets/datanaek/mbit-type". The augmented Essays dataset can be downloaded from "https://github.com/minekirito/DLP-Personality-Prediction/tree/main/data/essays".

Acknowledgments

This work was supported by National Natural Science Foundation of China-Joint Fund Project [U1536122], Key Special Project of "Technology Boosts Economy 2020" by Ministry of Science and Technology [SQ2020YFF0413781], and 2021 Tianjin Postgraduate Research and Innovation Project [2021YJSB252]. The authors thank Yuhan Yang and Chunyu Miao from Tianjin University of Technology for their revision on the manuscript.

Appendix A. Comparison of multiple pre-trained language models

In order to explore the effects of different pre-trained language models, we employ several performance comparison experiments by fixing model structures, hyper parameters, and psycholinguistic features. Each pre-trained language feature is concatenated with all psycholinguistic features in Fig. 2. The details of the pre-trained models are shown below.

Model	Layers	Hidden	Token length
BERT-base	12	768	512
BERT-large	24	1024	512
RoBERTa	12	768	512
Xlnet-base	12	768	512
Xlnet-large	24	1024	512
ALBERT-base	12	768	512
ALBERT-large	24	1024	512
BERTweet	12	768	128

The experiment results measured by accuracy are shown below. For Big 5 personality traits, we get the best results with ALBERT-base. For MBTI personality traits, the best pre-trained language model is BERT-large.

Model	Big 5						MBTI				
	Avg	*EXT*	*NEU*	*AGR*	*CON*	*OPN*	*Avg*	*I/E*	*N/S*	*T/F*	*P/J*
BERT-base	59.51	58.17	61.05	57.80	57.81	62.71	75.41	77.90	**86.444**	73.91	63.38
BERT-large	59.85	58.37	61.70	57.80	58.41	**62.95**	**78.08**	**78.70**	86.20	**78.14**	**69.27**
RoBERTa	57.98	55.45	60.56	56.22	55.86	61.81	74.47	77.43	86.22	71.95	62.27
Xlnet-base	56.75	55.61	58.89	56.63	54.16	58.45	74.75	77.52	86.29	72.32	62.85
Xlnet-large	56.39	54.64	57.48	56.42	54.60	58.81	74.30	77.49	86.26	71.73	61.72
ALBERT-base	**60.28**	**58.82**	**62.30**	**58.98**	**59.43**	61.86	75.47	77.97	86.441	73.88	63.60
ALBERT-large	59.32	58.37	60.15	57.60	58.74	61.74	74.90	77.71	86.37	73.00	62.52
BERTweet[a]	–	–	–	–	–	–	74.68	77.59	86.39	72.23	62.51

[a]The Essays dataset is not composed of tweets, so we only extract features of Kaggle MBTI with BERTweet.

Algorithm 2 generating $Matrix_{SU}$

Input: $F = \{f_1, f_2, f_3, \dots, f_n\}$
Output: $Matrix_{SU}$

for i **in** F **do**
    Calculate the entropy of i and store it in the entropy list $List_e$
end for
for i **in** F **do**
    for j **in** F **do**
        Calculate relative conditional entropy $H(F_i \mid F_j)$ and store $H(F_i \mid F_j)$ in the conditional entropy list $List_c$.
    end for
end for
tmp = 0
for i **in** $List_e$ **do**
    for j **in** $List_e$ **do**
        Calculate the symmetric uncertainty SU_{ij} according to i, j and $List_c[tmp]$. Store SU_{ij} in the symmetric uncertainty list $List_s$
        tmp = tmp+1
    end for
end for
reshape $List_s$ to $n * n$ symmetric uncertainty matrix $Matrix_{SU}$
End algorithm

Appendix B. Pseudocode of generating $Matrix_{SU}$

Appendix C. Pseudocode of getfitness()

Algorithm 3 *getFitness*()

Input: $F = \{f_1, f_2, f_3, \dots, f_n\}$, label set L, $Matrix_{SU}$, a set of solutions $solution_i$ in pop
Output: $fitness_i$ corresponding to $solution_i$

Decode the selected feature F_{select} according to $solution_i$
for i **in** F_{select} **do**
    Calculate symmetric uncertainty $SU_{f_i label}$ between i and L. Store it in the symmetric uncertainty list $List_{FL}$.
    Find the position index of i in F and store it in the index list $List_i$
end for
for i **in** $List_i$ **do**
    for j **in** $List_i$ **do**
        Store $Matrix_{SU}[i,j]$ in the symmetric uncertainty list $List_{FF}$.
    end for
end for
Calculate the average value of elements in $List_{FL}, List_{FF}$, and record as avg_{FL}, avg_{FF}.
Calculate $fitness_i$ corresponding to $solution_i$ according to avg_{FL}, avg_{FF} and Eq. (24).
End algorithm

Appendix D. Convergence analysis of IDGWOFS

Markov chain is used to analyze the convergence of IDGWOFS.

Definition 1. In IDGWOFS, the state of the grey wolfs is recorded as γ. The state space of the grey wolfs is recorded as $\Gamma = \{\gamma \mid \gamma \in Z\}$. Where Z is feasible solution space.

Definition 2. In IDGWOFS, the state of the grey wolf groups is recorded as $\phi = (\gamma_1, \gamma_2, \dots, \gamma_i), i = 1, 2, \dots, N$. The state space of the grey wolf groups is recorded as $\Phi = \{\phi = (\gamma_1, \gamma_2, \dots, \gamma_i) \mid \gamma_i \in \Gamma, i = 1, 2, \dots, N\}$.

Definition 3. In IDGWOFS, $\forall \gamma_i, \gamma_j \in \Gamma$, the state transition of the grey wolfs is recorded as $Trans_{\phi}(\gamma_i) = \gamma_j$.

Definition 4. In IDGWOFS, $\forall \phi_i, \phi_j \in \Phi$, the state transition of the grey wolf groups is recorded as $Trans_{\Phi}(\phi_i) = \phi_j$. Then, the transition probability of the grey wolf groups is recorded as

$$P(Trans_{\Phi}(\phi_i) = \phi_j) = \prod_{n=1}^{N} P(Trans_{\phi}(\gamma_{im}) = \gamma_{jm}). \tag{30}$$

Theorem 1. *In the original GWO, the state sequence of the grey wolf groups is a finite homogeneous Markov chain (Solis & Wets, 1981).*

Theorem 2. *According to the convergence criterion of the optimization algorithm (Solis & Wets, 1981), if the state sequence of the grey wolf groups is a finite homogeneous Markov chain, GWO is convergent (Zhang, Long, Wang, & Yang, 2020).*

Theorem 3. *In IDGWOFS, the state sequence of the grey wolf groups is a finite homogeneous Markov chain.*

Next, we prove Theorem 3.

Proof. In order to prove Theorem 3, the finiteness, Markov property, and homogeneity of the state sequence of the grey wolf groups in IDGWOFS must be proved in turn.

Finiteness The search space of solution of IDGWOFS is finite, so γ_i is finite, and Γ is finite. Φ consists of Γ_i, the number of Γ_i is finite positive integer, so Φ is finite, and then the state sequence of the grey wolf groups $\{\phi(t) \mid t > 0\}]$ has finiteness.

Markov property According to Definition 4, $\forall \phi(t-1), \phi(t) \in \Phi$, the transition probability $P(Trans_{\Phi}(\phi(t-1)) = \phi(t))$ of the state sequence of the grey wolfs $\{\phi(t) \mid t > 0\}]$ is determined by the transition probability $P(Trans_{\Phi}(\gamma(t-1)) = \gamma(t))$ of all grey wolfs in the grey wolf groups. According to Eqs. (1–12,25–28), $P(Trans_{\Phi}(\gamma(t-1)) = \gamma(t))$ is only related to $\Phi(\gamma(t-1))$, distance D and the parameters r_3 of neighbor search strategy. So, $P(Trans_{\Phi}(\phi(t-1)) = \phi(t))$ is only related to the state at $t-1$. According to the basic properties of Markov chain, $\{\phi(t) \mid t > 0\}]$ has Markov property.

Homogeneity If the one-time-step transition probability of the state sequence is independent of the starting time, the Markov chain is homogeneous. $P(Trans_{\Phi}(\gamma(t-1)) = \gamma(t))$ is only related to the state at $t-1$ and unrelated to $t-1$. So, $\{\phi(t) \mid t > 0\}]$ has homogeneity.

The state sequence of the gray groups in the IDGWOFS is a finite homogeneous Markov chain. i.e., our proposed improvements for GWO have not changed the Markov chain in the original GWO and its properties.

In summary, our proposed IDGWOFS is convergent.

References

Acharya, A., Aryan, A., Saha, S., & Ghosh, A. (2022). Impact of COVID-19 on the human personality: An analysis based on document modeling using machine learning tools. *The Computer Journal*, *2022*, bxab207.

Aguiar, J. J. B., Fechine, J. M., & Costa, E. B. (2020). Collaborative filtering strategy for product recommendation using personality characteristics of customers. In *Proceedings of the Brazilian symposium on multimedia and the web* (pp. 157–164).

Argamon, S., Koppel, D. S. M., & Pennebaker, J. (2005). Lexical predictors of personality type. In *Proceedings of the joint annual meeting of the interface and the classification society of North America* (pp. 1–16).

Cambria, E., Poria, S., Hazarika, D., & Kwok, K. (2018). Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 1795–1802).

Chaturvedi, I., Satapathy, R., Cavallari, S., & Cambria, E. (2019). Fuzzy commonsense reasoning for multimodal sentiment analysis. *Pattern Recognition Letters*, *125*, 264–270.

Chen, H., Han, L., Hu, Z., Hou, Q., Ye, Z., Zeng, J., et al. (2019). A feature selection method of parallel grey wolf optimization algorithm based on spark. In *10th IEEE international conference on intelligent data acquisition and advanced computing systems: technology and applications* (pp. 81–85).

Chen, H., Tu, S., & Xu, H. (2021). The application of improved grasshopper optimization algorithm to flight delay prediction-based on spark. In *Proceedings of the 15th international conference on complex, intelligent and software intensive systems* (pp. 80–89).

Dai, J. H., Chen, J. L., Liu, Y., & Hu, H. (2020). Novel multi-label feature selection via label symmetric uncertainty correlation learning and feature redundancy evaluation. *Knowledge-Based Systems*, *207*, Article 106342.

Devyatkin, D., Smirnov, I., Ananyeva, M., Kobozeva, M., Chepovskiy, A., & Solovyev, F. (2017). Exploring linguistic features for extremist texts detection (on the material of Russian-speaking illegal texts). In *2017 IEEE international conference on intelligence and security informatics*.

El-Demerdash, K., El-Khoribi, R. A., Ismail, S. M. A., & Abdou, S. (2021). Deep learning based fusion strategies for personality prediction. *Egyptian Informatics Journal*, *23*(1), 47–53.

Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. (2017). Grey wolf optimizer: A review of recent variants and applications. *Neural Computing and Applications*, *30*(2), 413–435.

Huang, Y., Du, C., Xue, Z., Chen, X. Y., Zhao, H., & Huang, L. B. (2021). What makes multi-modal learning better than single (provably). In *35th conference on neural information processing systems*.

Jeremy, N. H., Christian, G., Kamal, M. F., Suhartono, D., & Suryaningrum, K. M. (2021). Automatic personality prediction using deep learning based on social media profile picture and posts. In *4th international seminar on research of information technology and intelligent systems* (pp. 166–172).

Jiang, H., Zhang, X. Z., & Choi, D. J. (2020). Automatic text-based personality recognition on monologues and multiparty dialogues using attentive networks and contextual embeddings. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 13821–13822).

Kazameini, A., Fatehi, S., Mehta, Y., Eetemadi, S., & Cambria, B. (2020). Personality trait detection using bagged svm over bert word embedding ensembles. In *Proceedings of the The Fourth Widening Natural Language Processing Workshop*.

Kumar, V., & Sonajharia, M. (2014). Feature selection: A literature review. *The Smart Computing Review*, *4*(3), 211–229.

Li, W., Hu, X., Long, X., Tang, L., Chen, J., Wang, F., et al. (2020). EEG responses to emotional videos can quantitatively predict big-five personality traits. *Neurocomputing*, *415*, 368–381.
Lin, L. X., Lin, H., Wan, J., Wang, Y. S., & Gao, J. (2021). A novel method for driving path planning with spark. *Journal of Engineering*, *2021*, 1–10.
Ling, G., Wang, Z., Shi, Y., Wang, J., Lu, Y., & Li, L. (2022). Membrane fouling prediction based on tent-SSA-BP. *Membranes*, *12*(691).
Lou, P., Lu, G., Jiang, X., Xiao, Z., Hu, J. W., & Yan, J. W. (2021). Cyber intrusion detection through association rule mining on multi-source logs. *Applied Intelligence*, *51*, 4043–4057.
Lynnette, H. X., & Carley, K. M. (2022). Is my stance the same as your stance? A cross validation study of stance detection datasets. *Information Processing and Management*, *59*(6), Article 103070.
Ma, L. Y., & Sun, J. M. (2022). NOx emission optimization based on SDAE prediction model and improved SSA. *Proceedings of the CSEE*, *42*(14), 5194–5202.
Mairesse, F., Walker, M., Mehl, M., & Moore, R. (2007). Using psycholinguistic cues for the automatic recognition of personality in conversation and text. *Journal of Artificial Intelligence Research*, *30*(1), 457–500.
Majaluoma, S., Seppala, T., Kautiainen, H., & Korhonen, P. (2020). Type D personality and metabolic syndrome among finnish female municipal workers. *BMC Women's Health*, *20*(1), 202–209.
Mawadatul, M., & Hilman, F. P. (2021). Prediction of Myers–Briggs type indicator personality using long short-term memory. *Jurnal Elektronika dan Telekomunikasi*, *21*, 104–111.
Mehta, Y., Fatehi, S., Kazameini, A., Stachl, C., Cambria, E., & Eetemadi, S. (2020). Bottom-up and top-down: Predicting personality with psychopsycholinguistic and language model features. In *Proceedings of 2020 IEEE international conference on data mining* (pp. 1184–1189).
Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, *114*, 163–191.
Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications*, *27*, 495–513.
Mohades, D. F., Sadr, H., & Tarkhan, M. (2022). Contextualized multidimensional personality recognition using combination of deep neural network and ensemble learning. *Neural Processing Letters*.
Mohammad, S. (2018). Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of the 56th annual meeting of the association for computational linguistics* (pp. 174–184).
Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, *29*, 436–465.
Moustafa, A. A., Bello, A., & Maurushat, A. (2021). The role of user behaviour in improving cyber security management. *Frontiers in Psychology*, *12*, Article 561011.
Nadimi-Shahraki, M. H., Taghian, S., & Mirjalili, S. (2020). An improved grey wolf optimizer for solving engineering problems. *Expert Systems with Applications*, *166*, Article 113917.
Niu, X. Z., Zheng, Y. H., Fournier-Viger, P., & Wang, B. (2021). Parallel grid-based density peak clustering of big trajectory data. *Applied Intelligence*.
Pabon, F. O. L., & Arroyave, J. R. O. (2022). Automatic personality evaluation from transliterations of YouTube vlogs using classical and state of the art word embeddings. *Ingenieria e Investigacion*, *42*(2), Article e93803.
Pavan, K. K. N., & Gavrilova, M. L. (2022). Latent personality traits assessment from social network activity using contextual language embedding. *IEEE Transactions on Computational Social Systems*, *9*(2), 638–649.
Phan, L. V., & Rauthmann, J. F. (2021). Personality computing: New frontiers in personality assessment. *Social and Personality Psychology Compass*, *15*(7), Article e12624.
Pintas, J. T., Fernandes, L. A. F., & Garcia, A. C. B. (2021). Feature selection methods for text classification: A systematic literature review. *Artificial Intelligence Review*, *54*, 6149–6200.
Poria, S., Gelbukh, A., Agarwal, B., Cambria, E., & Howard, H. (2013). Common sense knowledge based personality recognition from text. In *Mexican international conference on artificial intelligence* (pp. 484–496).
Principi, R. D. P., Palmero, C., Junior, J. C. S. J., & Escalera, S. (2021). On the effect of observed subject biases in apparent personality analysis from audio-visual signals. *IEEE Transactions on Affective Computing*, *12*(3), 607–621.
Ren, Z., Shen, Q., Diao, X., & Xu, H. (2021). A sentiment-aware deep learning approach for personality detection from text. *Information Processing and Management*, *58*, Article 102532.
Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software*, *105*, 30–47.
Shappie, A. T., Dawson, C. A., & Debb, S. M. (2020). Personality as a predictor of cybersecurity behavior. *Psychology of Popular Media*, *9*(4), 475–480.
Shumanov, M., & Johnson, L. (2021). Making conversations with chatbots more personalized. *Computers in Human Behavior*, *117*, Article 106627.
Solis, F. J., & Wets, J. B. (1981). Minimization by random search techniques. *Mathematics of Operations Research*, *6*, 19–30.
Song, Z. C., Kang, J., Sun, G. L., & He, Y. J. (2018). The comparison of three measures in feature selection. *Journal of Harbin University of Science and Technology*, *23*(01), 111–116.
Stajner, S., & Yenikent, S. (2020). A survey of automatic personality detection from texts. In *Proceedings of the 28th international conference on computational linguistics* (pp. 6284–6295).
Tadist, K., Mrabti, F., Nikolov, N. S., Azeddine, Z., & Said, N. (2021). SDPSO: Spark distributed PSO-based approach for feature selection and cancer disease prognosis. *The Journal of Big Data*, *8*(19).
Tausczik, Y., & Pennebaker, J. (2010). The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, *29*(1), 24–54.
Vasquez, R. L., & Ochoa-Luna, J. (2021). Transformer-based approaches' for personality detection using the MBTI model. In *2021 XLVII Latin American computing conference* (pp. 1–7).
Wang, C., Yao, H., & Liu, Z. (2019). An efficient DDoS detection based on SU-genetic feature selection. *Cluster Computing*, *22*, 2505–2515.
Wang, Y., Zheng, J., Li, Q., Wang, C., Zhang, H., & Gong, J. (2021). Xlnet-caps: Personality classification from textual posts. *Electronics*, *10*(1360).
Wei, J. W., & Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 6381–6387).
Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*, 67–82.
Yang, D., & Li, W. (2021). A comprehensive investigation of the impact of feature selection techniques on crashing fault residence prediction models. *Information and Software Technology*, *139*, Article 106652.
Yang, F., Quan, X., Yang, Y., & Yu, J. X. (2021). Multi-document transformer for personality detection. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 35*, (16), (pp. 14221–14229).
Yang, T., Yang, F., Ouyang, H., & Quan, X. J. (2021). Psycholinguistic tripartite graph network for personality detection. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing* (pp. 4229–4239).
Yuan, C., Wu, J., Li, H., & Wang, L. (2018). Personality recognition based on user generated content. In *15th international conference on service systems and service management* (pp. 1–6).
Zhang, M. J., Long, D. Y., Wang, X., & Yang, J. (2020). Research on convergence of grey wolf optimization algorithm based on Markov chain. *Acta Electronica Sinica*, *48*, 9–18.
Zheng, X., & Chen, W. (2021). An attention-based Bi-LSTM method for visual object classification via EEG. *Biomedical Signal Processing and Control*, *63*, Article 102174.