

Received September 25, 2021, accepted October 7, 2021. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.3121137

A Method for MBTI Classification Based on Impact of Class Components

NINOSLAV CERKEZ¹, (Member, IEEE), **BORIS VRDOLJAK²**, (Member, IEEE),
AND SANDRO SKANSI³

¹College of Information Technologies (VSITE), 10000 Zagreb, Croatia

²University of Zagreb, Faculty of Electrical Engineering and Computing, 10000 Zagreb, Croatia

³University of Zagreb, Faculty of Croatian Studies, 10000 Zagreb, Croatia

Corresponding author: Ninoslav Cerkez (ninoslav.cerkez@vsite.hr)

This work was supported by the European Regional Development Fund through the Operational Programme Competitiveness and Cohesion 2014-2020, under the project System for real-time monitoring and control of distributed processes, anomaly detection, early warning and forensic transaction analysis - PCC (KK.01.2.1.02.0097).

ABSTRACT Predicting the personality type of text authors has a well-known usage in psychology with practical applications in business. From the data science perspective, we can look at this problem as a text classification task that can be tackled using natural language processing (NLP) and deep learning. This paper proposes a method and a novel loss function for multiclass classification using the Myers–Briggs Type Indicator (MBTI) approach for predicting the author’s personality type. Furthermore, this paper proposes an approach that improves the current results of the MBTI multiclass classification because it considers components of compound class labels as supportive elements for better classification according to MBTI. As such, it also provides a new perspective on this classification problem. The experimental results on long short-term memory (LSTM) and convolutional neural network (CNN) models outperform baseline models for multiclass classification, related research on multiclass classification, and most research with four binary approaches to MBTI classification. Moreover, other classification problems that target compound class labels and label parts with binary mutually exclusive values can benefit from this approach.

INDEX TERMS Binary classification, compound class labels, cross-entropy loss, custom loss function, deep learning, machine learning, MBTI, Myers-Briggs Type Indicator, multiclass classification, natural language processing, personality computing.

I. INTRODUCTION

The evaluation of personality type classification has an important practical role, especially in the business environment, when hiring new employees, managing careers, and giving promotions. Moreover, research [1] has shown that predicting personality type is useful in health care because it can help predict mental illnesses. However, standard approaches in psychology for personality type evaluation are slow and expensive because they include surveys and highly qualified professionals. On the other hand, from a data science perspective, predicting the personality type of a text author is an example of NLP classification problems. Therefore, including deep learning and NLP is a natural choice to improve this process [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Liu.

Even though there is no general definition of personality accepted by all personality theorists, there is a consensus that personality is a pattern of relatively permanent traits and unique characteristics that result in consistency and individuality in a person’s behavior [3]. Therefore, personality assessments require reliable and verified techniques. Standard techniques in psychology for personality assessment are self-assessment, projections, and sampling methods, to name a few of them. If we can verify consistency in measured values with acceptable variance, we qualify the technique as reliable. In addition, when there is a commitment that the technique measures targeted traits, the technique is verified. For this purpose, psychologists have developed techniques and tools for personality assessment that result in personality prediction. There are widely known reliable and verified instruments to predict personality type, and among them are the Big Five (OCEAN) [4], Enneagram [5], and DiSC

Assessment [6]. Most papers related to text author personality prediction studies consider the Big Five or Myers-Briggs Type Indicator (MBTI) personality models. The Big Five personality model defines personality through the following five dimensions: extroversion, agreeableness, conscientiousness, neuroticism, and openness [7]. However, in this study, we focus on the MBTI method. We only focus on the computational approach and do not go deeply into psychological studies to detect the personality of the text author.

The typical approach to solving the classification of text authors based on MBTI instrument includes binary classification, where each component of the MBTI type is treated as a binary classification problem. However, in this research, we propose a method that considers the impact of individual components in multiclass classification; for this purpose, we introduce a custom loss function. As such, the method enables better results in multiclass classification compared to the present research and provides a new perspective and directions to solve the multiclass classification problem. With this method, we solve the problem of multiclass MBTI classification in a new way. This approach is vital because it allows the use of multiclass classification with the impact of compound class labels.

Another motivation for this approach was to create a base for new experiments regarding the deeper meaning of MBTI types related to cognitive functions. In addition, we conducted research using long short-term memory (LSTM) and convolutional neural network (CNN) models to prove the idea and benchmark the efficiency of our method. The present research on multiclass classification reports relatively low results compared to the binary approach, and additional motivation was to improve these results.

We define the problem with the following questions: How to conduct MBTI multiclass classification while including all compound classes? How to overcome the overlap and unbalance problem between the compound classes? An input is a dataset in textual format with two columns: textual content of the author's post and MBTI type label for the author. The output of our model is a predicted MBTI label for a given text.

To solve this problem, the contributions of our paper are as follows: (1) a method for encoding and extracting the impact of the compound class, (2) a novel loss function for training, and (3) training, evaluation, and benchmark of LSTM and CNN models for MBTI personality prediction.

We organized the paper as follows: Section II gives an overview of MBTI as an approach for personality prediction; Section III presents the proposed method for encoding MBTI labels, approaching individual components' probability, including label components' probability in the custom loss function; Section IV presents related work on machine learning approaches to MBTI personality prediction; Section V presents the results of the proposed method and loss function and discusses the results; and finally, Section VI concludes the paper.

II. MBTI AND PERSONALITY PREDICTION

The first personality test was developed during World War I for the US military. Taibi Kahler, with NASA funding, developed one of the most frequently used personality models to this day. Modern approaches model personality by classifying it into a certain number of dimensions and developing an appropriate questionnaire as a measurement tool [8], [9].

Based on Jung's personality type theory, the MBTI is a questionnaire-based instrument for evaluating personality types [10]–[12]. Its purpose is to make a distinction between participants regarding the two categories in each of the four core dimensions. Isabel Briggs Myers and Katharine Cook Briggs originated the MBTI during the 1940s and first published it in 1962. Since this instrument has enormous popularity, almost two million people use it each year for business purposes [13]. However, there is doubt regarding MBTI instrument validity [14], [15], as there is an objection regarding the MBTI instrument because it lacks the stability-neuroticism trait. In addition, some studies confirm a correlation between the MBTI model and the Big Five model, where extroversion dimensions correlate strongly, and J/P correlates with conscientiousness. In addition, the study shows that the MBTI components are more complex for prediction than the Big Five components [16]. Research [17] also reports that one can obtain better performance with algorithms trained on MBTI than Big Five, and that Big Five offers more information and significant variability depending on the algorithm used.

Jung introduced the terms attitude and function in the description of personality. Attitude defines orientation as external or internal. Cognitive functions are essential in Jung's theory in developing personality types. However, their impact on the MBTI was not the focus of this study. Today, we can find synonyms for the term function in mental processes, cognitive processes, and cognitive functions. It is crucial for the MBTI model that each function can have external or internal aspects. Finally, Jung described functions according to perception (sensation or intuition) and judgment (thinking or feeling). In summary, in the MBTI model, there are four dimensions or dichotomies, each consisting of two mutually exclusive categories.

Going deeper into the MBTI dimensions, the first one is Extrovert (E) vs. Introvert (I). This indicates that a person is more outgoing, talkative, or reserved. In other words, it defines how a person's orientation toward the external or internal world is its primary energy motivation. The second is sensation (S) vs. intuition (N). It defines how a person perceives the information. For example, a person with a more sensing approach processes more facts, while a person with a higher intuitive approach tries to interpret information and find deeper meanings. The third dimension is thinking (T) vs. feeling (F). This dimension describes how a person makes decisions. For example, a person with a thinking approach uses logic and consistency in reasoning and making decisions, while a person with a more feeling approach uses

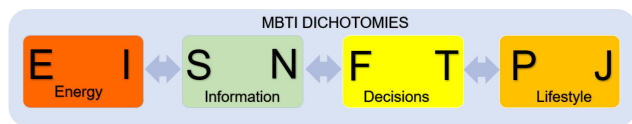


FIGURE 1. MBTI dichotomies.

empathy and focuses on people and particular circumstances. The last dimension is judgment (J) vs. perception (P). This dimension describes a person's orientation to the outer world and how a person lives daily; in other words, a person's lifestyle. For example, a person with judging preference opts for an organized daily life, compared to a person who prefers flexibility. This led us to 16 possible combinations of MBTI personality types. Because each class has four labels, it is evident that these labels are compound. For example, a person who generally prefers being alone (I), trust their intuition in perceiving and interpreting information (N), uses logic in reasoning (T), and lives a kind of spontaneous life (P) would mostly belong to MBTI type INTP. Figure 1 gives an overview of the four MBTI dichotomies, with driving forces for each of them.

III. THE METHOD FOR APPROACHING COMPOUND CLASS LABELS AND LOSS FUNCTION

Solving the MBTI classification problem involves two common approaches in supervised machine learning. First, one can take personality type classification according to MBTI as a multiclass classification into 16 classes. The second stage divides the problems into four binary classification problems.

When we tried to solve the MBTI classification as a binary classification problem, we divided the problem into four binary classifications. First, we included a new column for the first dichotomy and assigned values of 0 and 1. In this way, we mapped the 'E' and the 'I' dimensions by conducting binary classification for the first two dichotomies. This approach simplifies the problem since each row belongs to either the 'E' class or 'I' class. Similarly, we repeated the process for the other three dichotomies. Finally, the overall success of the four binary classifications was calculated by combining the results of the individual components. However, ensemble binary classifications were not the subject of interest in this study.

On the other hand, the multiclass approach must handle multiple problems in the MBTI dataset, such as imbalance and overlapping between classes. For example, we expected that the chosen model would treat classes ESTP and ESTJ as distinct classes, even though they have a majority of their parts as overlapping and slightly different in the last part, in addition to the small number of examples of both classes. This case is an excellent example of the motivation for our method, which can access parts of the compound class labels.

Because the standard multiclass approach does not allow flexibility like the binary approach and gives lower results in MBTI classification, the binary approach to four dichotomies

TABLE 1. Encoding MBTI labels.

| MBTI | Encoding |
|------|----------|
| ENFJ | 0 |
| ENFP | 1 |
| ENTJ | 2 |
| ENTP | 3 |
| ESFJ | 4 |
| ESFP | 5 |
| ESTJ | 6 |
| ESTP | 7 |
| INFJ | 8 |
| INFP | 9 |
| INTJ | 10 |
| INTP | 11 |
| ISFJ | 12 |
| ISFP | 13 |
| ISTJ | 14 |
| ISTP | 15 |

is a natural choice. With this approach, it is possible to obtain dichotomies that are easier to separate because we treat only two of them in each classification, keeping in mind that we can modify each classification if needed for actual dichotomies, leading to better accuracy. Noticeably, this approach also leads to more extensive training data for each classification and more balanced data. However, even though this approach is well known, we wanted to improve the multiclass approach.

The motivation for this research was to include the impact of compound class components in compound labels in the algorithm for MBTI multiclass classification. Thus, we can mitigate or reinforce the effects of misclassified elements, and consequently, misclassified compound classes. Furthermore, this approach also has potential for future research, including cognitive functions, because the present methods lack that direction.

We explain this method in two ways. In the first part, we describe the technique of approaching the compound class labels because it is the first problem we have to solve. The second part describes how we can use the resulting label to calculate the probability for that dimension and then how to use it in the proposed loss function.

A. METHOD OF APPROACHING THE COMPOUND CLASS-LABELS

The starting challenge in including the impact of compound class components is approaching these components because the standard approach converts starting compound class labels to integer values, usually in the range of 0 to 15. We found the encoding approach to be a solution to this challenge.

First, we decided to sort string classes in ascending English alphabetic order. Then, for such sorted classes, we assigned integer values for class encoding. The results of this approach are presented in Table 1.

We get the 'E' label at the first position in the first eight labels and the 'I' label at the first position in the last eight

labels with this approach. Similarly, we can recognize the patterns for the second, third, and fourth labels.

These patterns have two essential roles: calculating the probability for each component and determining the loss according to the correct element and position in the compound class.

1) CALCULATING COMPONENT PROBABILITY

The typical result from the model in a neural network has a final output of raw values – logits. The next level is usually softmax, which converts logits into probabilities. For example, the softmax function for MBTI classification is expressed as follows:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=0}^{15} e^{z_j}} \quad (1)$$

We note the raw output vector with \vec{z} , and the probability of the i^{th} component of the vector with $\sigma(\vec{z})_i$. The sum of the probabilities for all 16 elements was equal to 1.

$$\sum_{i=0}^{15} \sigma(\vec{z})_i = 1 \quad (2)$$

Because our model classifies compound labels, the softmax probabilities are the probabilities of the compound labels. Therefore, considering the encoded labels, we can calculate the probability for each component by summarizing all softmax probabilities with the appearance of that component. We provide an example for the 'E' and 'I' components:

$$\begin{aligned} P(E) = & P(ENFJ) + P(ENFP) + P(ENTJ) + P(ENTP) \\ & + P(ESFJ) + P(ESFP) + P(ESTJ) + P(ESTP) \end{aligned} \quad (3)$$

$$P(E) = \sum_{i=0}^7 P_i(MBTI) \quad (4)$$

$$\begin{aligned} P(I) = & P(INFJ) + P(INFP) + P(INTJ) + P(INTP) \\ & + P(ISFJ) + P(ISFP) + P(ISTJ) + P(ISTP) \end{aligned} \quad (5)$$

$$P(I) = \sum_{i=8}^{15} P_i(MBTI) \quad (6)$$

In addition, the sum of probabilities for labels 'E' and 'I' must be equal to one.

$$P(E) + P(I) = 1 \quad (7)$$

Similarly, we calculated the probabilities of other class components. It should be noted that this calculation must follow the chosen encoding scheme.

2) DETERMINING THE CORRECT COMPONENT AND POSITION

Keeping in mind that this method penalizes the prediction of the wrong component in the compound class, it is essential to include the loss for the correct element. For example, if the ground-true label is ENFJ and that model predicts INFJ, we would like to penalize the model to make a mistake at the dichotomy E/I. In other words, to allow the model to learn better the component that the model missed in classifying the whole MBTI type.

First, we decide whether to take a softmax probability or not:

$$(\text{sgn}(y_i - \hat{y}))^2 \quad (8)$$

For any difference between the target and predicted labels, the expression will have a value of 1.

Second, we must check whether there is a difference between the labels for the ground truth and the predicted label for each position. For this purpose, we used a starting encoding scheme. For example, we can check the first label with (div is an integer division):

$$(\text{sgn}(\text{div}(y_i, 8) - \text{div}(\hat{y}, 8)))^2 \quad (9)$$

For any difference in the first position between the target and predicted labels, the expression will have a value of 1.

Third, when there is a difference between some label components, the next step is to decide which probability between two possibilities at that position to choose. For that purpose, we again use the encoding scheme in Table 1. For example, if there is a difference at the first label, we can calculate the corresponding probability as follows:

$$(1 - \text{div}(y_i, 8)) * P(E) + \text{div}(y_i, 8) * P(I) \quad (10)$$

It is evident that for the first eight labels, we have $P(E)$ and $P(I)$ for the last eight labels.

With similar steps and slightly different pattern recognition, we can determine the probability components for each label. Finally, the next step involves transforming the calculated probabilities into weighted parts of the loss function.

B. PROPOSED LOSS FUNCTION

The standard approach in multiclass classification uses cross-entropy loss as a cost function when optimizing the classification models. Cross-entropy evaluates the difference between two probability distributions and has origins in information theory [18].

The definition of cross-entropy (CE) for a discrete probability distribution with N events gives:

$$CE(y, p) = - \sum_{i=1}^N y_i \log(p_i) = - \log(p_k) \quad (11)$$

An accurate probability distribution is y_i as the truth label, and p_i is the estimated softmax probability distribution for the i^{th} class. The probability related to the ground truth is equal to one for one-hot encoding. In other words, we encode the target probability distribution with values of 1 for index k and 0 for others. The classification model approximates the target probability distribution, and the cross-entropy calculates the total entropy between two distributions.

Imbalanced datasets, such as naturally imbalanced MBTI datasets, have skewed probability distributions and low entropy because the most likely classes prevail. Considering that multiclass classification models intensively implement CE because of the fast calculation, it is essential to note that CE considers only the actual class probability. In other words, the CE does not carry the probability among the other classes.

However, our proposed method considers some misclassified classes by approaching a missed portion of the compound class label.

We propose a novel loss function, the cross-entropy compound class-label impact (CECI) loss, with tunable weight parameters. This loss function includes a weighted penalty for misclassified class label compounds and penalizes misclassified compound classes as well as misclassified components.

$$\begin{aligned} CECI(y, \mathbf{p}) &= CE(y, \mathbf{p}) + \alpha * CE(y(E|I), \mathbf{p}(E|I)) \\ &+ \beta * CE(y(N|S), \mathbf{p}(N|S)) + \gamma * CE(y(F|T), \mathbf{p}(F|T)) \\ &+ \delta * CE(y(P|J), \mathbf{p}(P|J)) \end{aligned} \quad (12)$$

α , β , γ , and δ are weights, with the corresponding cross-entropy loss for each component, according to the corresponding dichotomy position. Regarding the values for weights, we conducted intensive testing and obtained the best results in terms of relevant metrics of F1-score and recall with values larger than 0 and slightly around 1.

IV. RELATED WORK

Quantitatively comparing the available research on personality trait classification from text, the most significant research covers the Big Five approach [19]. In addition, reviews of personality detection from the text confirm that most studies cover the Big Five instruments [20]. Because our research focuses on the MBTI classification, we mainly present work related to this problem.

Since they were created before deep learning, standard machine learning algorithms were the first options for MBTI classification. For example, in [21], the authors implemented extreme gradient boosting as a machine learning approach for individual training for each pair of dichotomies. The authors used accuracy as the only metric. The highest value presented for accuracy was the N/S dichotomy (86.06%) and the lowest for the J/P dichotomy (65.70%). They also used the recurrent neural network, and the highest accuracy was 77.8% for the F/T dichotomy, and the lowest was 62% for N/S. In this approach, a binary classification was used.

In [22], the authors also used binary classification across MBTI dichotomies using a simple neighbor classifier. The presented results were the best for the E/I dichotomy, and metrics such as recall and precision were between 80% and 95%, while other metrics were between 40% and 70%. However, the dichotomy of J/P has the lowest accuracy.

This paper [23] proposes ensemble learning models for binary MBTI classification, namely bagging, boosting, and stacking. The authors reported that stacking showed the best performance with a 97.53% accuracy for the S/N dichotomy. The authors also reported other metrics for model evaluation, and the highest precision was also shown by the stacking model, with the highest recall showing the boosting model. Finally, the highest F1-score (97.42%) was received from the stacking model again.

The authors in [24] used SVM, naïve Bayes, and neural net classifiers for binary MBTI classification. The best accuracies were obtained using SVM for E/I of 84.9%, S/N of 88.4%, T/F of 87%, and J/P of 78.8%. For the semantic and emotional representation of the text, the authors used linguistic inquiry and word count (LIWC), EmoSenticNet (EmoLex), and ConceptNet in combination with TF-IDF for each row and singular value decomposition (SVD).

The study [25] used an MBTI dataset created by 40 graduate students based on in-class writing samples. The authors used naïve Bayes and support vector machine (SVM) approaches for binary MBTI classification. The naïve Bayes approach with a precision and recall higher than 75% yielded better results than the SVM.

Some researchers have reported random forest classifier as a valuable and the best solution for MBTI binary classification. The authors used Word2vec for word vector representations and additional features, namely words per comment. The reported accuracy for all dichotomies was 100%. However, other model evaluation metrics, which are essential for imbalanced dataset classification, are not presented in this report [26].

Gradient boosting for prediction and K-means clustering with traditional TF-IDF for clustering is an approach proposed in [27] for binary MBTI classification. The proposed architecture achieved the best accuracy of 89.01% for E/I, and the dichotomy F/T showed the lowest accuracy of 81.19%.

On the other hand, the study [28] used classical supervised machine learning and a deep learning approach for MBTI classification. In addition, researchers have used multiclass and binary classification approaches. The baseline method was the softmax classifier, and the reported results were accurate. They reported the best result for the LSTM network with an accuracy of 23% for multiclass classification. Regarding the binary approach, the highest accuracy was 38%, again, with the LSTM network.

In [29], the authors compared an extra tree classifier, naïve Bayes, logistic regression, and SVM as corresponding machine learning algorithms for MBTI classification. They reported the best results for logistic regression, where the original accuracy and F-score were 66.59%. The authors stated that they chose the accuracy of the classifier as the most important metric, which is doubtful because of the dataset imbalance that research does not take into account. After parameter tuning, they reported an improvement of 1%. However, this study did not cite quantitative details regarding parameter tuning.

A review of recent trends in deep-learning approaches to personality detection is provided [30]. The authors conducted research based on the input modality and used text, audio, video, and multimodal sources. From that paper, we can observe the dominance of the Big Five studies using deep learning. In addition, this review reports only one deep learning approach on MBTI, which makes our approach even more valuable for the research community. Finally, the authors expect researchers to explore more accurate and efficient

ways of labeling datasets, which could raise the quality of datasets and increase their number.

The deep learning approach was also used in [31], where the authors provided binary classification according to four dichotomies, using the LSTM recurrent neural network in Keras. The research reported better results for LSTMs compared to RNN, GRU, and bi-LSTM. The accuracy for the user classification was for E/I, N/S, and P/J between 62% and 68%. The best reported result (77.8%) was for F/T. Confusion matrices indicated a similar pattern for N/S and P/J, where the authors reported more false positives and false negatives. However, the overall accuracy was very low, at only 21%. The authors also used the MBTI Kaggle dataset. However, even though other researchers often make reference to this research, to the best of our efforts, we could not find this resource in the official conference repository for the conference noted in this paper.

With an example of 3.62 billion users on Twitter, such an enormous number of visitors creates massive post volume on social networks that grow 20 %–30% daily [32], [33]. Social media are environments with massive interactions between members, and as such, they are a large-scale source of data for open-vocabulary personality prediction.

In [34], the authors contributed to the MBTI classification problem by providing a new, large-scale Reddit dataset labeled with MBTI types by extracting and analyzing a set of features and benchmark models for personality prediction. Three classifiers were used in this study: a three-layer multilayer perceptron (MLP), logistic regression (LR), and support vector machine (SVM). Again, they set up the problem as four binary classification problems. The best results were obtained using LR and MLP approaches. The best macro F1-score for the E/I dimension was 82.8% for the S/N 79.2%, T/F 67.2%, and J/P 74.8%. The authors also provided the MBTI type classification, and the best macro F1-score was 41.7% for MLP. This paper observes that the results show how difficult it is to distinguish INTP from INTJ, that INTJ is more similar to INFJ, and INTP is more similar to INFP. In short, the results show grouping similar MBTI types, similar to the MBTI theory.

The modest data from Twitter social media posts can also predict personality [35]. The authors used the Big Five and the MBTI instruments, and the approach does not rely on a particular lexicon; in other words, it is language independent. The presented results, based only on word counts, showed the highest values for the S/N dichotomy. Furthermore, this study showed significant differences in the results across the selected languages. Therefore, the E/I dichotomy had the best predictions for German, Italian, and Spanish languages. In addition, this work presented potential sources of prediction errors: structural error of the prediction algorithm, changing the text author over time, and using the essays as a baseline.

In [36], the authors used binary word n-grams and gender to predict the MBTI post author type with self-reported labeled tweets. As meta-features, they used followers, tweets,

and retweets, such as the number of favorite tweets. Finally, they used logistic regression as a model, and the authors concluded that E/I and F/T dichotomies have fairly good distinctions, compared to other dimensions where learning was complex and with lower success. The highest reported result was 77% - for the accuracy of the E/I dichotomy.

The public information shared on Twitter can be a relevant source for predicting personality types according to the Big Five instrument [37], where the authors used ZeroR and Gaussian Processes as machine learning algorithms and achieved results for each personality trait between 11% and 18%.

This paper [38] presents experiments on the Twitter dataset for binary MBTI classification and with 12 different algorithms, namely, stochastic gradient descent (SGD), random forest (RF), logistic regression (LR), K-nearest neighbors (KNN), naïve Bayes (NB), multinomial naïve Bayes (MNB), Gaussian naïve Bayes (GNB), support vector machine (SVM), multilayer perceptron (MLP), decision tree (DT), bagging, and extra tree classifier (ET). For axis E/I, the highest accuracy of 78.6% was given by LR and MLP, but the highest F1 score and recall were 38% and 40%, respectively, for SGD. For the second dimension S/N, the highest accuracy of 86.2% was given by MLP, and the highest F1 score and recall were 17% and 18%, respectively, for DT. The third experiment for the dimension F/T achieved the highest accuracy of 64.7% for MLP, and the highest F1 score and recall were 69% and 100%, respectively. Finally, the MLP provided the best accuracy for the P/J axis, with BNB as the classifier with the highest F1 score and recall.

The authors in [39] used a novel dataset for various experiments on the Big Five, MBTI, and Enneagram personality models. A precious fact is that this dataset includes demographic data (age, gender, location, and language). With regard to the MBTI training, the achieved type-level accuracy was 45%. In the experiments, the author used binary classification, linear/logistic regression, and neural networks. The neural network approach has a considerable scope for improvement because there are many comments per user.

Interesting semantic challenges are social networks in languages that are different from English. For example, Chinese semantic analysis is more complex than English. Sina Weibo is one of the most popular sites in China and the leading microblogging service provider in China. As such, Sina Weibo is a rich resource for personality prediction research. However, the number of Sina Weibo users recruited was relatively small (131 of 589 participants). The authors researched personality prediction according to the Big Five dimensions. Pearson's correlation analysis was used to compare the scores for the personality dimensions and all features. In addition, we used the linguistic inquiry and word count (LIWC) dictionary for content analysis, logistic regression, and naïve Bayes. The Naïve Bayes algorithm had better precision results, and both algorithms had similar recall results. The reported mean precision of the five personality traits was 70.7%. Keeping in mind the correlation between

Big Five and MBTI [16], it was an exciting observation that neuroticism was the hardest to predict. In addition, openness and agreeableness were easy to predict, mostly correlated with the MBTI S/N dimension [40].

In addition, research [41] has focused on open-vocabulary binary MBTI personality prediction in Bahasa, Indonesia. Again, Twitter served as the data source. The research used three statistical models, and the machine-learning naïve Bayes classifier outperformed lexicon-based and grammatical-rule-based approaches. The highest accuracy was 80% for the E/I dichotomy and 60% for the other four dichotomies. In addition, the researchers observed that the naïve Bayes classifier was the fastest in classification.

Balancing the MBTI dataset can lead to research that demonstrates how this balancing influences the MBTI classification. The research shows the use of the random over-sampling method and TF-IDF for feature selection. The authors experimented with the following machine learning algorithms: KNN, decision tree, random forest, MLP, LR, SVM, XGBoost, MNB, and SGDC. However, the XGBoost classification showed the best performance - more than 99% for precision and accuracy. This study also reported lower P/J dichotomy results [42].

Since researchers usually report the lowest result in the chosen metrics for the J/P dichotomy in classification according to MBTI, some researchers have focused on better predicting the last dichotomy. The emphasis is also on comparing performance using TF-IDF, character-level TF, TF-IDF, and word-level TF. The research also used the Personality Café MBTI dataset. Interestingly, the authors concluded that previous research on this dataset was overly optimistic. They used five machine learning algorithms and finally suggested using the LightGBM model with a character-level TF as the best model for predicting the P/J dichotomy because of its robustness. The results were compared with those of the SVM, which had similar results. This research used linguistic inquiry and word count (LIWC). The authors reported the best result for the P/J in the F1-Macro score of 80.77% for Kaggle and 65% for Kaggle-Filtered datasets. The authors suggest that the P dichotomy correlates better than the J dichotomy to linguistic markers in communication on social media [43].

Some approaches in predicting the personality of text authors consider that not all posts on social media are equally important and present a model that puts attention at the message-level to learn their relative weight. This study implements the concept of the Big Five dataset. The authors concluded that the last dichotomy is a crucial part of solving MBTI prediction [44].

In addition to proposing a new MBTI-labeled dataset, with personality type and gender for Dutch, German, French, Italian, Portuguese, and Spanish, the author [45] experimented using LinearSVC with 10-fold cross-validation. The authors also used logistic regression to obtain comparable results. The best results were obtained in Dutch, where the research

reported the best improvement compared to the weighted random baseline (WRB) in F1-score from 50.04% to 82.61% for gender prediction. However, the highest result was an F1 score of 79.21% for the S/N dichotomy in Italian regarding MBTI dimensions. The research again reports that the model outperforms the prediction of E/I and F/T dichotomies compared to the other two dimensions.

It is possible to treat text-building hierarchical, vectorial words, and sentence representations in deep learning models. With this method, it is possible to tackle personality prediction in multi-language tasks and achieve high performance. The authors used it on the Big Five dataset and three languages: English, Spanish, and Italian. It would be great to see this approach on the MBTI dataset, as promised in the paper [46].

Because there is a specific correlation between MBTI and the Big Five instruments, it is possible to predict the Big Five dimensions based on the MBTI-labeled dataset. The authors compared six supervised machine learning algorithms and three feature extraction methods (term frequency and inverse document frequency (TF-IDF), bag of words (BOW), and global vector for word representation (GloVe)). Again, they used the binary approach and obtained the best accuracy results for TF-IDF with random forest. For the experiment with BOW, they achieved the best accuracy with XGB. Finally, the authors achieved the best accuracy with GloVe gain with XGB, up to 99.99% [38].

In [47], the authors used Naïve Bayes, KNN, and SVM on the Big Five dataset, and according to reported results, Naïve Bayes gave the best overall result with an accuracy of 60%. The authors stated that the experiment failed to improve previous results and that the system had 65% accuracy compared to the survey-based test. However, we included this research because of its overall accuracy.

In [48], the authors used CNN and Mairesse features, and they obtained the best accuracy of 62.68% on the Big Five dataset. However, there is no discussion regarding balance in the dataset, and we cannot conclude whether this metric is the best one. Nevertheless, we emphasized this work because it presented multiclass approach results, one of the rare works to do so.

Some reduction approaches, such as principal component analysis (PCA) and information gain, showed slight improvements, with the highest gain of less than 2% in the Big Five dataset [49].

Predicting personality can be an additional tool for sentiment analysis to analyze email content and create a spam filter. This approach can be beneficial because the number of spam emails is increasing. These studies are examples of research in which the model includes MBTI personality prediction as a web service hosted on *uClassify* [50], [51].

Table 2 presents the research and applied algorithms in MBTI classification using the binary approach. Researchers do not have a unique approach to metrics, especially considering that the MBTI is an imbalanced dataset.

TABLE 2. Related Results of the binary MBTI classification.

| Paper | Approach | Classifier | Metric | The best result |
|-------|--------------|-----------------------------|----------------|---------------------------|
| [21] | Binary | XGBoost | Accuracy | 86.06% (N/S) |
| | | RNN | Accuracy | 77.8% (F/T) |
| [22] | Binary | KNN | Recall | 80%-95% (E/I) |
| | | | Precision | 84%-90% (E/I) |
| [24] | Binary | Naïve Bayes | Accuracy | 86.2 % (S/N) |
| | | SVM | | 88.4 % (S/N) |
| | | Neural Net | | 86.3 % (S/N) |
| [25] | Binary | Naïve Bayes | Precision | > 75% (S/N) |
| | | | Recall | >75% (S/N) |
| | | | Precision | >75% (F/T) |
| | | | Recall | >75% (F/T) |
| | | SVM | Precision | 31%-60% (S/N) |
| | | | Recall | 45% |
| [31] | Binary | LSTM | Mean Accuracy | 77.8% (F/T) |
| | | | Mean Accuracy | 67.6% (E/I) |
| [34] | Binary | LR, MVP | F1 – score | 82.8% (E/I) |
| [35] | Binary | MCCV | Avg accuracy | 92 % (S/N) |
| | | | F1 | 92 % (S/N) |
| | | | Ratio | 55 % (E/I) |
| [36] | Binary | Logistic regression | Avg accuracy | 77 % (E/I) |
| [39] | Binary | NN | Macro-avg F1 | 63.4% (T) |
| | | | | 54.6% (I) |
| | | | | 52.8% (N) |
| | | | | 56.6% (P) |
| | | | | 73.9% (T) |
| | | Linear/Logistic regression | Macro-avg F1 | 64.2% (P) |
| | | | | 65.4% (I) |
| | | | | 60.6% (N) |
| [38] | Binary | MLP, LR | Accuracy | 78.6% (E/I) |
| | | SVM, MLP | Accuracy | 86.2% (S/N) |
| | | MLP | Accuracy | 64.7% (F/T) |
| | | MLP | Accuracy | 59.6% (P/J) |
| | | SGD | F1 score | 38% (E/I) |
| | | DT | F1 score | 17% (S/N) |
| | | SGD | F1-score | 69% (F/T) |
| | | SGD, BNB, MNB, SVM | F1-score | 74% (P/J) |
| [41] | Binary | Naïve Bayes | Accuracy | 80% (E/I) |
| | | | Accuracy | 60% (S/N, F/T, J/P) |
| [45] | Binary | LR, Linear SVC | F1-score | 79.21% (S/N) |
| [43] | Binary (P/J) | LightGBM | F1-macro score | 80.77% (P/J) |
| | | | Accuracy | 82.77% (P/J) |
| | | | AUROC | 90.08% (P/J) |
| [42] | Binary | XGBoost | Accuracy | 99.92% (S/N) |
| | | | F1-score | 99.75% (S/N) |
| [26] | Binary | Random Forest | Accuracy | 100% (E/I, S/N, F/T, P/J) |
| [23] | Binary | Stacking | Accuracy | 95.79% (S/N) |
| | | | F1-score | 97.42% (S/N) |
| | | Boosting | Recall | 96.91% (S/N) |
| [27] | Binary | K-Means Clustering, XGBoost | Accuracy | 89.01% (E/I) |

Table 3 presents the research and applied algorithms in the MBTI classification with the multiclass approach or overall results with the binary approach.

Common supervised machine learning approaches to MBTI classification problems include multiclass classification into 16 classes or four binary classifications. Most MBTI

TABLE 3. Related results of the multiclass MBTI classification.

| Paper | Approach | Classifier | Metric | The best result |
|-------|-----------------------|----------------------------|--------------------|-----------------|
| [31] | Binary | LSTM | Overall accuracy | 21% |
| [34] | Binary | MLP | F1-score (overall) | 47% |
| [39] | Binary | Linear/Logistic regression | Accuracy (Overall) | 45% |
| [28] | Binary | Naïve Bayes | Accuracy (Overall) | 26% |
| | | Reg. SVM | Accuracy (Overall) | 33% |
| | | LSTM | Accuracy (Overall) | 38% |
| [28] | Multiclass | Softmax | Accuracy | 17% |
| | | LSTM | Accuracy | 23% |
| [48] | Multiclass (Big Five) | CNN | Accuracy | 62.68 % |
| [47] | Multiclass (Big Five) | Naïve Bayes | Accuracy | 60% |
| [29] | Multiclass | LR | Accuracy | 66.59% |

classification research uses a binary classification approach because it provides more flexibility than multiclass classification and provides higher values for classification metrics than multiclass classification based on standard CE. In addition, classes for the binary classification approach are more balanced, which allows for higher accuracy. From the perspective of our approach, binary classification results can provide insights for decisions regarding weight factors in CECI.

We wanted to separate the approaches and results for a more accurate benchmark in our approach. Therefore, we decided to summarize the results of the related work into two tables for binary and multiclass approaches. We left out a few Big Five multiclass classification results because of the number of multiclass classification studies for MBTI classification. For each table, we summarize the best-reported results and the algorithm applied to these results.

V. EXPERIMENTAL SETUP AND RESULTS

Figure 2. provides an overview of the pipeline of the proposed method. In the first part, we clean and preprocess the dataset. An essential part of this step is the encoding of MBTI labels, according to Table 1. Then, we conduct feature engineering, which results in embedding vectors. After that, we create two models using Bi-LSTM and CNN architectures. Our goal was not to find the optimal architecture for MBTI classification, as in [54], but to prove that the proposed method improves results with different architectures. In addition, since LSTM architectures are trained to recognize patterns across time, and CNN architectures recognize patterns across space, weighting parameters could lead to insights into the behavior of compound class labels. Finally, we trained and evaluated the models, applying the CECI loss function. For each phase, we provide more details in the following sections in this chapter.

1) TOOLS AND RESOURCES

This study was conducted on two platforms and setup environments. First, we used Windows 10, Python 3.8.5 as a

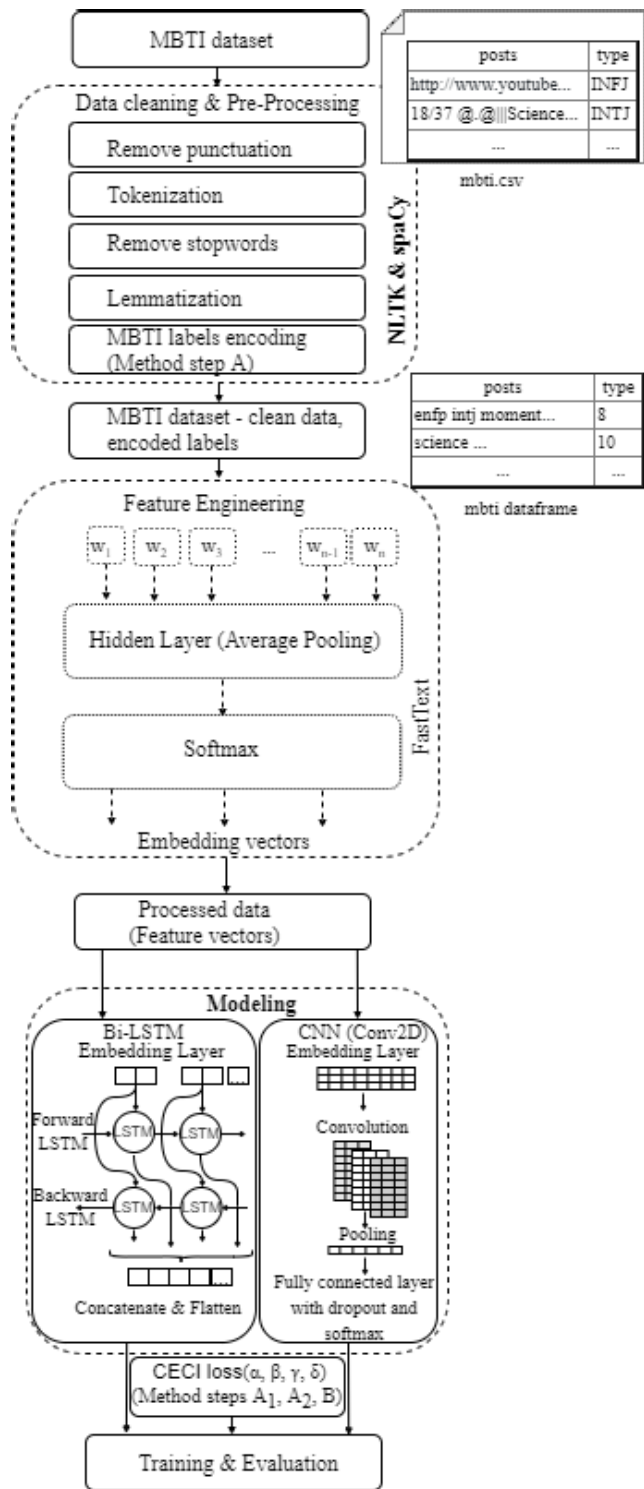


FIGURE 2. Overview of the pipeline of the method.

scripting language, Jupyter Notebook, and Python scripts. The essential libraries and Cuda versions were torch 1.8.1, cuda10.2, and torchtext 0.9.1. The graphical device for the GPU was a GeForce GTX 1050. This environment was used for prototyping and preliminary testing. Second, we used

| type | posts |
|------|--|
| INFJ | 'http://www.youtube.com/watch?v=qsXHcwe3krw ... |
| ENTP | 'I'm finding the lack of me in these posts ver... |
| INTP | 'Good one ____ https://www.youtube.com/wat... |
| INTJ | 'Dear INTP, I enjoyed our conversation the o... |
| ENTJ | 'You're fired. That's another silly misconce... |
| INTJ | '18/37 @.@ Science is not perfect. No scien... |
| INFJ | 'No, I can't draw on my own nails (haha). Thos... |

FIGURE 3. The Personality Cafe forum MBTI dataset.

NVIDIA DGX-1, with 8x NVIDIA Tesla V100 for final testing, and we presented the final results obtained from this DGX-1 environment.

2) DATASET AND TEXT PREPROCESSING

There is no unique standard dataset for machine-learning techniques based on the MBTI instrument. In [36], the authors proposed a corpus of 1.2M English tweets from 1.500 users and annotated it with self-reported MBTI personality type and gender. In [34] and [39], the authors proposed Reddit datasets MBTI9k, and PANDORA labeled with MBTI types. The PANDORA dataset is worth emphasizing because it is the first large-scale dataset covering multiple personality models (Big 5, MBTI, Enneagram) and includes demographic data, which most datasets lack.

There is also a corpus with the text author's MBTI personality type and gender for six Western European languages [45]. We used the MBTI dataset from Kaggle to demonstrate the proposed approach [52].

This dataset is a well-known dataset with 8.675 rows representing the self-reported MBTI personality type. The dataset originated from the Personality Café forum in 2017, and it contains all posts in English, with an approximate corpus of 11.2 million words in more than 420.000 labelled points. Each row represents the last 50 posts of each user.

Figure 3 shows a few rows in the MBTI dataset containing two feature columns: string values of compound labels and textual posts for each user.

Thus, users' discussions on the Personality Café determined the MBTI type [22]. Figure 4 shows the distribution of the classes in this dataset. The distribution of classes in the MBTI dataset indicates that we must deal with a highly imbalanced dataset.

Table 4 shows the number of occurrences for each type and the percentage of events compared to the total number of examples. For example, the first four classes account for 65.67% of all categories, and these classes can be considered the majority classes. Table 4 also gives the estimated relative

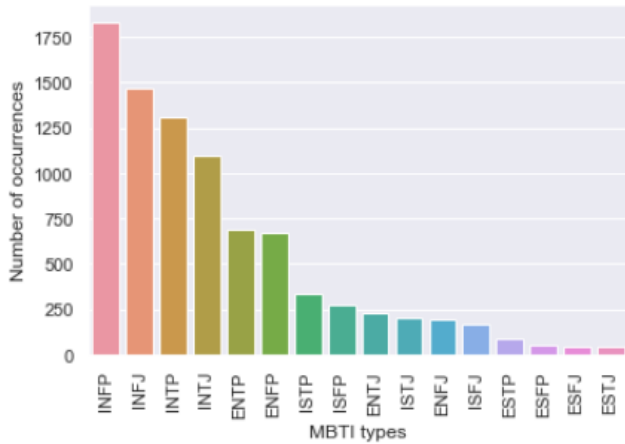


FIGURE 4. Distribution of classes in Personality Cafe MBTI dataset.

frequency of each of the 16 types in the United States population [53].

It is essential to note that the MBTI types are self-reported, and that data is limited to a particular forum that can influence the sample of the actual population. In addition, we noticed a significant difference between the distribution in the dataset and the general population for some classes. This observation could be the subject of interest for further research. However, it is helpful to compare the distributions of the four dichotomies. This information can be used as a guide for experiments with the weight factors for each component. For example, we can correlate the weighted dichotomy factor depending on the frequency in the population. The data is listed in Table 5. This information will be a direction for future research using the proposed method.

The MBTI dataset has 16 distinct labels, each consisting of four labels. The first place in compound labels corresponds to values E (extrovert) or I (Introvert), the second place corresponds to values N (intuitive) or S (sensitive), and the third position corresponds to values P (perceive) or J (judging). With such a structure, classification tasks on the MBTI dataset can have multiclass classification, multilabel, or four binary classification approaches. Therefore, along with the occurrence of MBTI types, it is helpful to analyze the number of words per post and the MBTI type. This data is presented in Table 6 and Figure 5. In this paper [26], we can find an analysis of the Pearson correlation between words per comment and ellipses per comment, concluding that there is a high correlation of 0.69 between words per comment and ellipses per comment for the overall dataset and that the highest correlation is for MBTI types ENFP, INFJ, and INTP.

Interestingly, the MBTI type with the second-lowest number of occurrences has the maximum average number of words per post.

We used standard data preprocessing steps before constructing the neural-network models. For example, we removed the numbers, special characters, links, and punctuation. Then we ensured that all tokens were lowercase;

TABLE 4. Distribution of classes.

| MBTI | Number of values | % of values | Estim./ US pop. |
|------|------------------|-------------|-----------------|
| INFP | 1832 | 21.12 | 4.4% (4-5%) |
| INFJ | 1470 | 16.95 | 1.5% (1-3%) |
| INTP | 1304 | 15.03 | 3.3% (3-5%) |
| INTJ | 1091 | 12.58 | 2.1% (2-4%) |
| ENTP | 685 | 7.90 | 3.2% (2-5%) |
| ENFP | 675 | 7.78 | 8.1% (6-8%) |
| ISTP | 337 | 3.89 | 5.4% (4-6%) |
| ISFP | 271 | 3.12 | 8.8% (5-9%) |
| ENTJ | 231 | 2.66 | 1.8% (2-5%) |
| ISTJ | 205 | 2.36 | 11.6% (11-14%) |
| ENFJ | 190 | 2.19 | 2.5% (2-5%) |
| ISFJ | 166 | 1.91 | 13.8% (9-14%) |
| ESTP | 89 | 1.03 | 4.3% (4-5%) |
| ESFP | 48 | 0.55 | 8.5% (4-9%) |
| ESFJ | 42 | 0.48 | 12.3% (9-13%) |
| ESTJ | 39 | 0.45 | 8.7% (8-12%) |

TABLE 5. Distribution of dichotomies.

| Dichotomy. | Occurrences | % | Estim./ US pop. |
|------------|-------------|-------|-----------------|
| E | 6676 | 76.96 | 49.3% (45-53%) |
| I | 1999 | 23.04 | 50.7% (47-55%) |
| S | 7478 | 86.20 | 73.3% (66-74%) |
| N | 1197 | 13.80 | 26.7% (26-34%) |
| T | 4694 | 54.11 | 40.2% (40-50%) |
| F | 3981 | 45.89 | 59.8% (50-60%) |
| J | 5241 | 60.41 | 54.1% (54-60%) |
| P | 3434 | 39.59 | 45.9% (40-46%) |

TABLE 6. Statistics of the dataset.

| Statistic | Value |
|--|--------|
| Maximum words per post – ENFP | 37.62 |
| Minimum words per post – INFP | 0.08 |
| Maximum average words per post - ESFJ | 25.81 |
| Minimum average words per count – ESFP | 20.44 |
| The average number of words per post | 24.52 |
| The average variance of word counts | 137.21 |

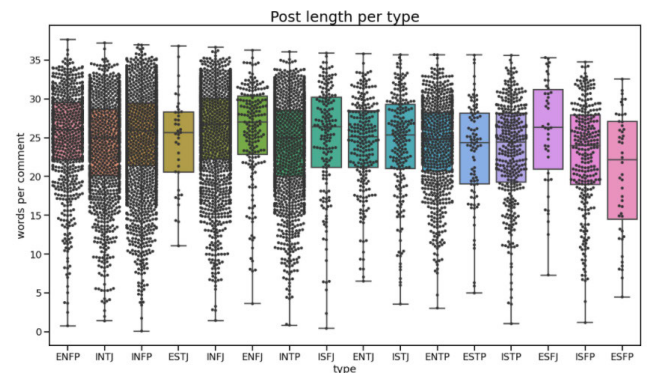


FIGURE 5. Post length per MBTI type.

we removed stop words, one-letter words, and transformed tokens into a list of words; finally, we converted the text into word embeddings using FastText.

```
LSTM(
  (embedding): Embedding(25002, 25, padding_idx=1)
  (rnn): LSTM(25, 25, num_layers=2, dropout=0.4, bidirectional=True)
  (fc1): Linear(in_features=50, out_features=25, bias=True)
  (fc2): Linear(in_features=25, out_features=16, bias=True)
  (dropout): Dropout(p=0.4, inplace=False)
)
```

FIGURE 6. LSTM model.

```
CNN(
  (embedding): Embedding(25002, 300)
  (convs): ModuleList(
    (0): Conv2d(1, 100, kernel_size=(2, 300), stride=(1, 1))
    (1): Conv2d(1, 100, kernel_size=(3, 300), stride=(1, 1))
    (2): Conv2d(1, 100, kernel_size=(4, 300), stride=(1, 1))
  )
  (fc): Linear(in_features=300, out_features=16, bias=True)
  (dropout): Dropout(p=0.2, inplace=False)
)
```

FIGURE 7. CNN model.

3) TRAINING AND VALIDATION SETUP

We divided the initial dataset into a training dataset and a validation dataset with a ratio of 4:1. In addition, we used stratification options. Initially, we set up seed, deterministic, and benchmark options to ensure that the training had repeatable results for the chosen platform. The training batch size was 256 and the validation batch size was 64. We used a BucketIterator with a False value for the sort option and the True for the sort_within_batch option as an iterator.

We used a bidirectional long short-term memory network (Bi-LSTM) and a 2-dimensional convolutional model (CNN). Using these two models, we verified that the model works on both common model types for NLP classification problems. For LSTM, we used two layers with 25 neurons. The dropout value was 0.4. We trained both models through 40 epochs. In addition, we trained all models with CE and CECI and experimented with the values for the weight parameters.

An overview of the architecture of the LSTM model is given in Figure 6.

Our experiment wanted to keep the comparison explicit so that the impact of using CECI compared to CE is easy to measure. Therefore, as values for weights α , β , γ , and δ , we used an experimental approach with values between 0 and 1; we chose the step for changing the value of 0.05 to limit the computational workload.

An overview of the architecture of the CNN model is given in Figure 7.

Finally, we evaluated the results by comparing multiple metrics, such as F1-score, accuracy, precision, recall, and confusion matrix, as metrics suitable for imbalanced datasets. In comparison to the results, we have in mind that the F1-score measures the balance between recall and precision, which is essential for imbalanced datasets. In the next section, we present the experimental results.

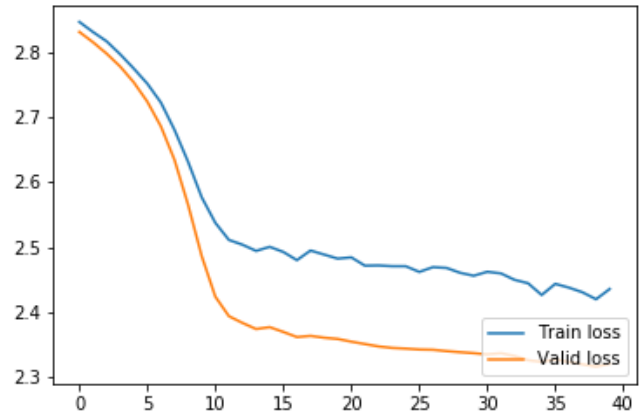


FIGURE 8. Training/Validation loss for LSTM - CE.

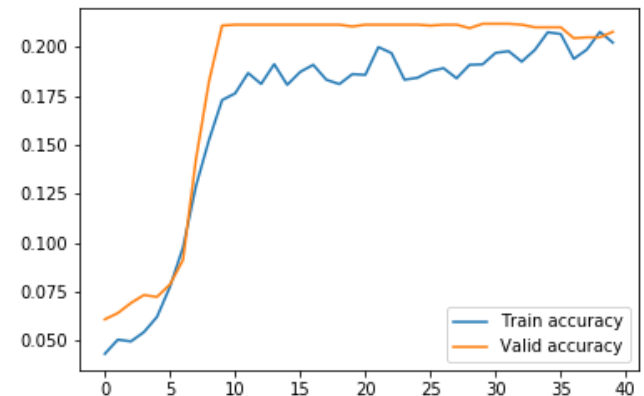


FIGURE 9. Training/Validation accuracy for LSTM - CE.

4) RESULTS AND DISCUSSIONS

First, we trained the LSTM model with standard CE as a baseline because our approach should first show improvement to the standard CE and obtained the following results. Figure 8 shows the results for multiclass training and validation loss. Figure 9 shows the training and validation accuracies.

These results were within the expected range for such an imbalanced dataset. In addition, these results are in the range and are comparable to other reported results in Table 3 using similar architectures. Unfortunately, this LSTM model with standard CE learns poorly and is thus prone to overfitting.

Table 7 presents a classification report of the CE approach. In training models on such an imbalanced dataset, we focus on metrics like the F1-score. The results of 14% for the weighted average F1-score and 4% for the macro F1-score were again in the expected range. Figure 10 presents the confusion matrix for the CE approach. Again, the model learns the best with the majority classes INFP and INFJ, which is the range of expected results because majority classes prevail, and with standard CE, the model prefers majority classes and has low generalization.

In our training for weight parameters in LSTM CECI, we obtained the best results for α , β , γ , and δ using 0.7, 0.5,

TABLE 7. Classification report for LSTM CE.

| MBTI | precision | recall | F1-score | support |
|--------------|-------------|-------------|-------------|---------|
| ENFJ | 0.00 | 0.00 | 0.00 | 40 |
| ENFP | 0.15 | 0.10 | 0.12 | 320 |
| ENTJ | 0.00 | 0.00 | 0.00 | 40 |
| ENTP | 0.05 | 0.04 | 0.05 | 160 |
| ESFJ | 0.00 | 0.00 | 0.00 | 80 |
| ESFP | 0.00 | 0.00 | 0.00 | 0 |
| ESTJ | 0.00 | 0.00 | 0.00 | 0 |
| ESTP | 0.00 | 0.00 | 0.00 | 0 |
| INFJ | 0.46 | 0.03 | 0.05 | 480 |
| INFP | 0.26 | 0.78 | 0.39 | 600 |
| INTJ | 0.03 | 0.02 | 0.03 | 80 |
| INTP | 0.26 | 0.02 | 0.32 | 320 |
| ISFJ | 0.00 | 0.00 | 0.00 | 120 |
| ISFP | 0.00 | 0.00 | 0.00 | 0 |
| ISTJ | 0.00 | 0.00 | 0.00 | 0 |
| ISTP | 0.00 | 0.00 | 0.00 | 40 |
| accuracy | | | 0.23 | 2280 |
| macro avg | 0.07 | 0.06 | 0.04 | 2280 |
| weighted avg | 0.23 | 0.23 | 0.14 | 2280 |

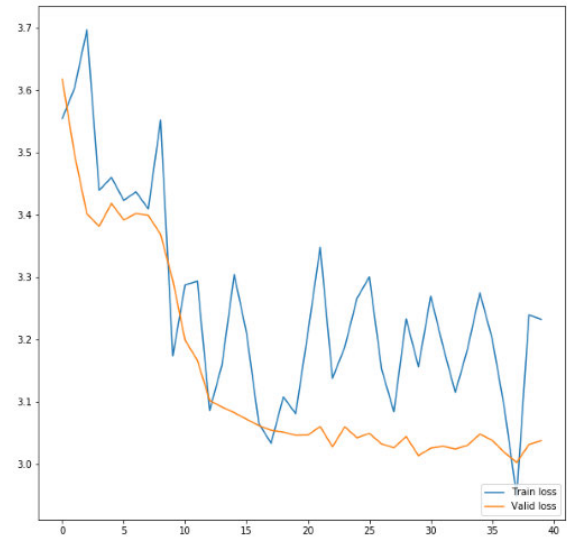


FIGURE 11. Training/Validation loss for LSTM - CECI(0.7, 0.5, 0.7, 0.6).

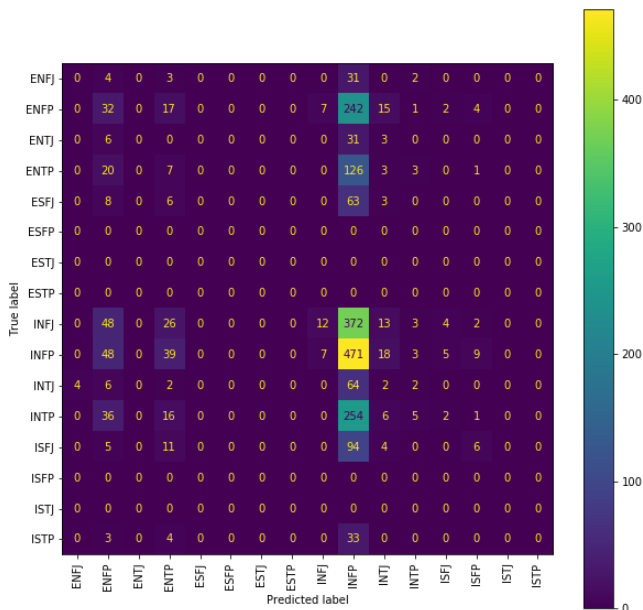


FIGURE 10. Confusion matrix for LSTM - CE.

0.7, and 0.6. Figure 11 shows the results for the training and validation losses for the best CECI combination.

The training and validation accuracy behaviors are in the CE approach range, as shown in Figure 9 and Figure 12. Therefore, the model learns slightly better, and then goes to overfitting. The validation accuracy has a better stability than the CE approach. Figure 12 shows the training and validation accuracies of the best CECI combinations. We conclude that the CECI method improves the training results, but still, the model has space to improve the impact of the imbalanced dataset and internal relationships among MBTI classes.

Table 8 presents a classification report of the CECI approach. The result of 20% for the weighted average F1-score outperformed the CE approach. The accuracy of the

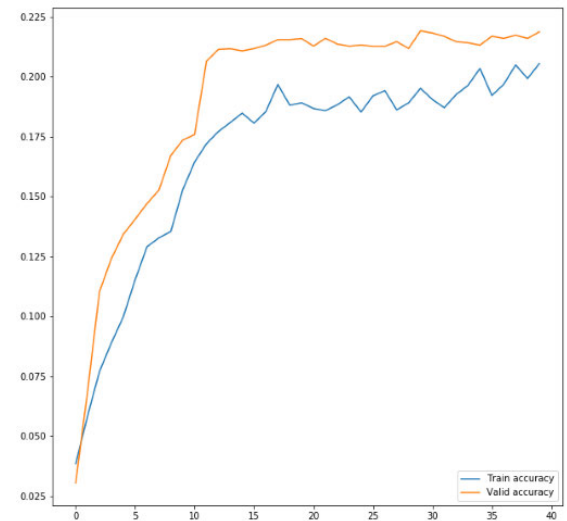


FIGURE 12. Training/Validation accuracy for LSTM-CECI(0.7, 0.5, 0.7, 0.6).

CECI approach was 27%, which also outperformed the CE approach. Regarding the macro F1-score metric, the CECI model shows an improvement from 4% to 7%. In addition, the model learned to classify the class ENFJ, the class that the CE approach missed, and missed the class INTJ. The CECI model also improved recall for INFJ from 0.03 to 0.26.

Figure 13 presents the confusion matrix for the CECI approach. Again, the model learns the best with majority classes, INFP and INFJ. However, the model showed improvement in all predicted classes compared to the CE approach. Comparing the base LSTM and LSTM-CECI models showed that our approach significantly improved the base LSTM model. Moreover, compared to other reported results in Table 3, the model outperforms the reported results in [28] and [31].

TABLE 8. Classification report for LSTM - CECI(0.7, 0.5, 0.7, 0.6).

| MBTI | precision | recall | F1-score | support |
|--------------|-------------|-------------|-------------|---------|
| ENFJ | 0.17 | 0.03 | 0.04 | 40 |
| ENFP | 0.24 | 0.14 | 0.18 | 320 |
| ENTJ | 0.00 | 0.00 | 0.00 | 40 |
| ENTP | 0.14 | 0.10 | 0.11 | 160 |
| ESFJ | 0.00 | 0.00 | 0.00 | 80 |
| ESFP | 0.00 | 0.00 | 0.00 | 0 |
| ESTJ | 0.00 | 0.00 | 0.00 | 0 |
| ESTP | 0.00 | 0.00 | 0.00 | 0 |
| INFJ | 0.43 | 0.26 | 0.32 | 480 |
| INFP | 0.28 | 0.68 | 0.40 | 600 |
| INTJ | 0.00 | 0.00 | 0.00 | 80 |
| INTP | 0.17 | 0.18 | 0.17 | 320 |
| ISFJ | 0.00 | 0.00 | 0.00 | 120 |
| ISFP | 0.00 | 0.00 | 0.00 | 0 |
| ISTJ | 0.00 | 0.00 | 0.00 | 0 |
| ISTP | 0.00 | 0.00 | 0.00 | 40 |
| accuracy | | | 0.27 | 2280 |
| macro avg | 0.07 | 0.07 | 0.07 | 2280 |
| weighted avg | 0.20 | 0.27 | 0.20 | 2280 |

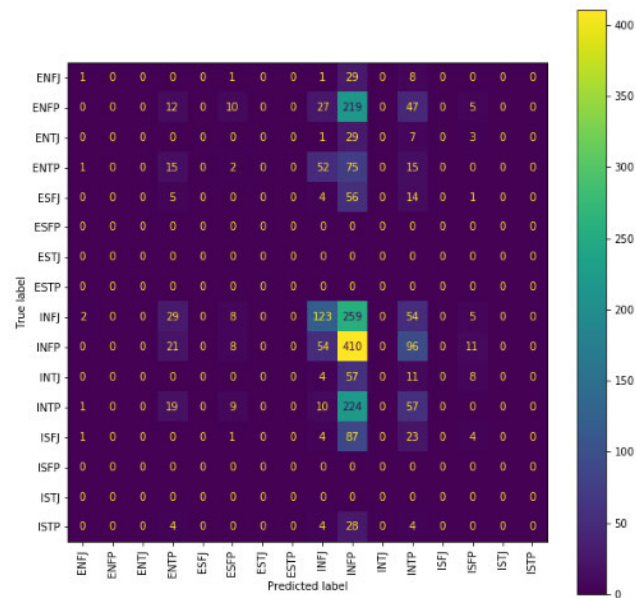


FIGURE 13. Confusion matrix for LSTM - CECI(0.7,0.5,0.7,0.6).

Second, we wanted to improve our model on other architectures and trained the CNN model with standard CE and then with CECI and obtained the following results. Figure 14 shows the results of the training and validation losses for the CE approach. In addition, Figure 15 shows the results for the training and validation accuracies of the CNN with the CE. Finally, Figure 16 shows the confusion matrix for the CNN CE approach. Again, these results are in the range expected for such an imbalanced dataset. However, the CNN results were significantly better than those obtained using both approaches with LSTM. For example, the weighted average F1-score (Table 9) for the CNN CE approach is 57%, compared to 14% (LSTM CE) and 20% (LSTM CECI), and

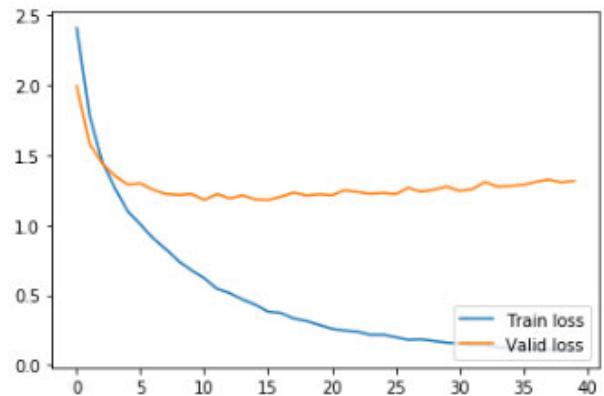


FIGURE 14. Training/Validation loss for CNN - CE.

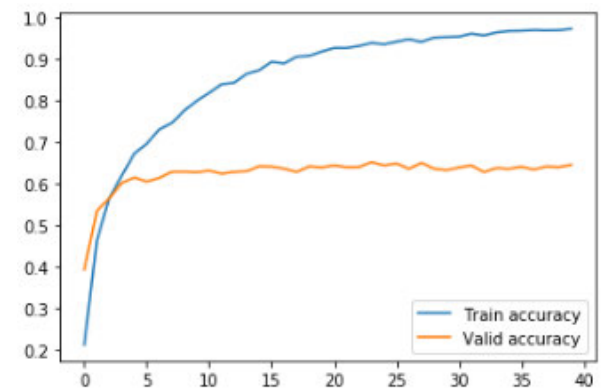


FIGURE 15. Training/Validation accuracy for CNN - CE.

the maximum F1-score is 27%, which is much better than the 4% and 7% for LSTM CE and LSTM-CECI.

Comparing these results to the reported results in Table 3, this model outperforms most models, except for the LR model in [29] and the MLP model in [34]. However, the reported metric in [29] for the LR model is the highest accuracy result that should be considered carefully because of the imbalanced dataset. In [34], the reported overall F1-score was relating to the binary-based approach. Compared to the binary-based approaches in [28] and [39] with our base CNN model, our model outperforms all the presented models. However, because we use the standard CNN model, these results indicate more about the performance of that architecture compared to other reported architectures. Again, the results with reported accuracy results should be considered carefully because of the imbalanced dataset. We emphasize the results in Table 9 that highly outperform the results compared to the LSTM models used in training. In addition, this model learned how to train classes INTJ and ISFJ, compared to previous models.

After CNN with CE, we trained the CNN model with CECI and obtained the best results for the values 0.1, 0.2, 0.7, and 0.1, respectively, for weights α , β , γ , and δ , and obtained the following results. Figure 17 shows the results

TABLE 9. Classification report for CNN - CE.

| MBTI | precision | recall | F1-score | support |
|--------------|-----------|-------------|-------------|---------|
| ENFJ | 0.00 | 0.00 | 0.00 | 40 |
| ENFP | 0.69 | 0.38 | 0.49 | 320 |
| ENTJ | 0.00 | 0.00 | 0.00 | 40 |
| ENTP | 0.64 | 0.59 | 0.62 | 160 |
| ESFJ | 0.00 | 0.00 | 0.00 | 80 |
| ESFP | 0.00 | 0.00 | 0.00 | 0 |
| ESTJ | 0.00 | 0.00 | 0.00 | 0 |
| ESTP | 0.00 | 0.00 | 0.00 | 0 |
| INFJ | 0.61 | 0.69 | 0.64 | 480 |
| INFP | 0.55 | 0.71 | 0.62 | 600 |
| INTJ | 0.50 | 0.99 | 0.66 | 80 |
| INTP | 0.78 | 0.96 | 0.86 | 320 |
| ISFJ | 1.00 | 0.22 | 0.36 | 120 |
| ISFP | 0.00 | 0.00 | 0.00 | 0 |
| ISTJ | 0.00 | 0.00 | 0.00 | 0 |
| ISTP | 0.00 | 0.00 | 0.00 | 40 |
| accuracy | | | 0.61 | 2280 |
| macro avg | 0.30 | 0.28 | 0.27 | 2280 |
| weighted avg | 0.59 | 0.61 | 0.57 | 2280 |

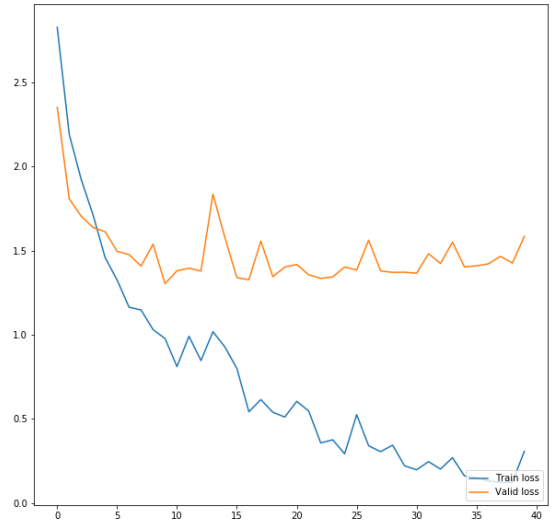


FIGURE 17. Training/Validation loss for CNN - CE (0.1, 0.2, 0.7, 0.1).

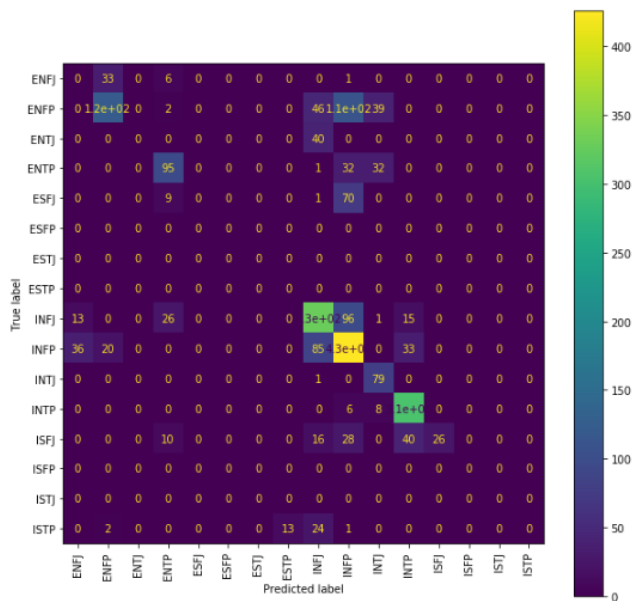


FIGURE 16. Confusion matrix for CNN - CE.

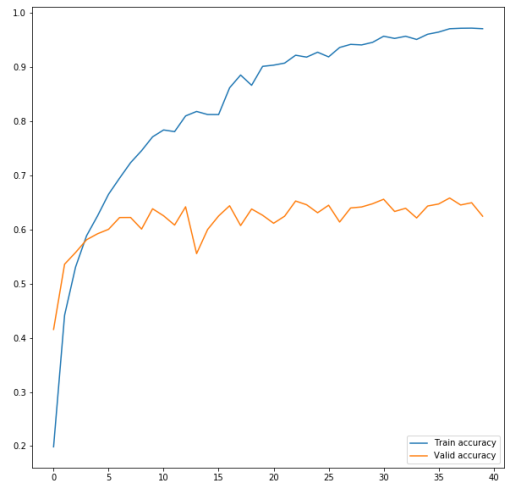


FIGURE 18. Training/Validation acc. for CNN - CE (0.1, 0.2, 0.7, 0.1).

of the training and validation losses. Figure 18 shows the training and validation accuracy results.

These results are much better than those obtained using both LSTM approaches and the basic CNN CE approach. For example, CNN CE (0.1, 0.2, 0.7, 0.1) approved macro F1-score from 27% to 33%, and the model learned how to classify the class ISTP, the class that the CNN CE approach missed.

The CNN CE approach achieved an 86% F1-score and 75% recall for the ISTP class. In addition, it shows that the CNN CE approach has considerable potential for modeling this type of classification problem.

In addition, we would like to note that with the LSTM CE approach, the best result included the highest penalization for the first two MBTI dichotomies and with CNN CE the third dichotomy. We also present summary results for the other CE weights. This observation could be a direction for future research.

Table 10 presents a classification report of the CE approach. Figure 19 shows the confusion matrix for the CNN CE approach, and the result of 63% for the weighted F1-score outperformed the CE approach. However, the accuracy of the CE approach was 66%, which also outperformed the CE approach. In addition, the CNN CE model improved the recall for ENTP from 59% to 65%. Moreover, there is a class (INTJ) where the CE model had a slightly better F1-score of 66% compared to 63% for the CNN CE approach.

TABLE 10. Classification report for CNN – CECI (0.1, 0.2, 0.7, 0.1).

| MBTI | precision | recall | F1-score | support |
|--------------|-------------|-------------|-------------|---------|
| ENFJ | 0.00 | 0.00 | 0.00 | 40 |
| ENFP | 0.75 | 0.36 | 0.49 | 320 |
| ENTJ | 0.00 | 0.00 | 0.00 | 40 |
| ENTP | 0.56 | 0.65 | 0.60 | 160 |
| ESFJ | 0.00 | 0.00 | 0.00 | 80 |
| ESFP | 0.00 | 0.00 | 0.00 | 0 |
| ESTJ | 0.00 | 0.00 | 0.00 | 0 |
| ESTP | 0.00 | 0.00 | 0.00 | 0 |
| INFJ | 0.66 | 0.79 | 0.72 | 480 |
| INFP | 0.66 | 0.79 | 0.72 | 600 |
| INTJ | 0.48 | 0.95 | 0.63 | 80 |
| INTP | 0.79 | 0.97 | 0.87 | 320 |
| ISFJ | 0.96 | 0.22 | 0.35 | 120 |
| ISFP | 0.00 | 0.00 | 0.00 | 0 |
| ISTJ | 0.00 | 0.00 | 0.00 | 0 |
| ISTP | 1.00 | 0.75 | 0.86 | 40 |
| accuracy | | | 0.66 | 2280 |
| macro avg | 0.37 | 0.34 | 0.33 | 2280 |
| weighted avg | 0.65 | 0.66 | 0.63 | 2280 |

TABLE 11. Classification report for CNN – CECI (0.0, 0.2, 0.0, 0.5).

| MBTI | precision | recall | F1-score | support |
|--------------|-------------|-------------|-------------|---------|
| ENFJ | 0.00 | 0.00 | 0.00 | 40 |
| ENFP | 0.71 | 0.32 | 0.44 | 320 |
| ENTJ | 0.00 | 0.00 | 0.00 | 40 |
| ENTP | 0.75 | 0.83 | 0.79 | 160 |
| ESFJ | 0.00 | 0.00 | 0.00 | 80 |
| ESFP | 0.00 | 0.00 | 0.00 | 0 |
| ESTJ | 0.00 | 0.00 | 0.00 | 0 |
| ESTP | 0.00 | 0.00 | 0.00 | 0 |
| INFJ | 0.64 | 0.77 | 0.70 | 480 |
| INFP | 0.59 | 0.77 | 0.67 | 600 |
| INTJ | 0.51 | 1.00 | 0.68 | 80 |
| INTP | 0.85 | 0.96 | 0.90 | 320 |
| ISFJ | 1.00 | 0.19 | 0.32 | 120 |
| ISFP | 0.00 | 0.00 | 0.00 | 0 |
| ISTJ | 0.00 | 0.00 | 0.00 | 0 |
| ISTP | 1.00 | 0.55 | 0.71 | 40 |
| accuracy | | | 0.66 | 2280 |
| macro avg | 0.38 | 0.34 | 0.33 | 2280 |
| weighted avg | 0.65 | 0.66 | 0.62 | 2280 |

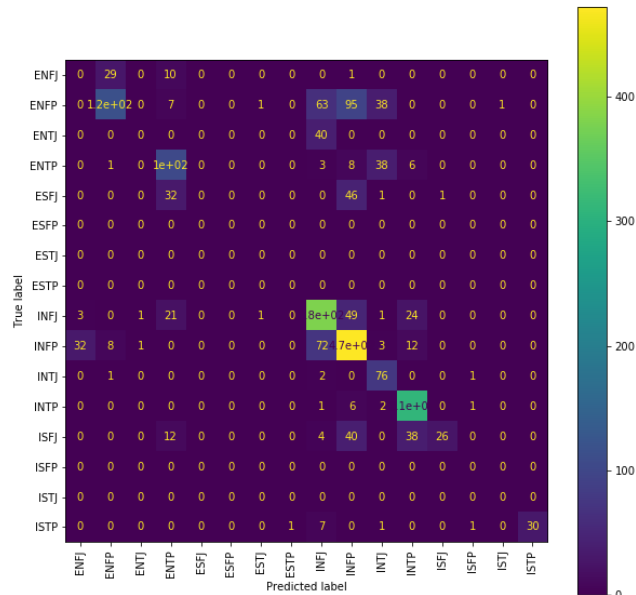


FIGURE 19. Confusion matrix for CNN – CECI (0.1, 0.2, 0.7, 0.1).

We can see that with different values for α , β , γ , and δ , we can manage the training goals. For example, on an imbalanced dataset, such as the MBTI dataset, we focused on the macro average F1-score as a more informative metric, and these weights helped us improve the results compared to the baseline (CE). In addition, with CECI(0.0, 0.2, 0.0, 0.5) and CNN CECI(0.1,0.2,0.7,0.1) we raised the F1-score and recall for the class ISTP compared to CNN CE, and with CECI(0.1, 0.2, 0.0, 0.25), we obtained the highest weighted F1-score.

Keeping in mind that we have a highly imbalanced dataset and that we would likely want to maximize our maximum F1-score as a measure of equally paid attention to all classes, we also achieved improved results for CECI (0.0, 0.2, 0.0,

TABLE 12. Overview of results.

| Model | Loss | $\alpha, \beta, \gamma, \delta$ | macro F1 | weighted F1 |
|-------|------|---------------------------------|-------------|-------------|
| LSTM | CE | 0.0,0.0,0.0,0.0 | 0.04 | 0.14 |
| LSTM | CECI | 0.7, 0.5, 0.7, 0.6 | 0.07 | 0.20 |
| CNN | CE | 0.0,0.0,0.0,0.0 | 0.27 | 0.57 |
| CNN | CECI | 0.1, 0.2, 0.0, 0.25 | 0.30 | 0.65 |
| CNN | CECI | 0.0, 0.2, 0.0, 0.5 | 0.33 | 0.62 |
| CNN | CECI | 0.1, 0.2, 0.7, 0.1 | 0.33 | 0.63 |

0.5) and we present the classification report in Table 11 for these weights.

Table 12 summarizes the best results achieved in experiments with LSTM and CNN models.

We researched two typical neural network models, LSTM and CNN, and in both, we obtained improvements with the CECI approach compared to the standard CE objective function.

Our LSTM model with CE predicted MBTI types poorly; the macro F1-score was 4% and the weighted F1-score was 14%. In contrast, the LSTM with CECI (0.7, 0.5, 0.7, 0.6) gave a macro F1-score of 7% and a weighted F1-score of 20%. On the other hand, the CNN model with CE had a macro F1-score of 27% and a weighted F1-score of 57%. Finally, the CNN model with CECI (0.0, 0.2, 0.7, 0.1) had a macro F1-score of 33% and a weighted F1-score of 63%. In addition, with CECI, we had a better prediction for some classes, which base models missed. For example, LSTM with the CECI model learned how to predict the class ENFJ but missed the class INTJ. The CNN model with the CECI learned to predict the ISTP. Thus, comparing the CECI approach based on the CE models improved both with the LSTM and CNN models.

Before comparing the CNN approach with other reported results in Tables 2 and 3, we would like to emphasize how the

right metrics are essential because of the imbalanced MBTI dataset.

Because the MBTI dataset is imbalanced, using accuracy as a metric is doubtful and misleading [55]. This is especially true when we perform multiclass classification with a highly imbalanced dataset and binary classification if an imbalance exists. For example, in the Personal Café MBTI dataset, there was a high imbalance in the first two dichotomies (Table 5). Introverts account for 76.96% of the first dichotomy, and intuition accounts for 86.20% of the second. Therefore, having high accuracy does not validate a model, with a binary or multiclass approach, as a successful model because the high accuracy on an imbalanced dataset usually means that the model predicts majority classes but misses minority classes.

The precision or positive predictive value calculates the fraction of true positives divided by the number of positively predicted classes. In this way, precision gives a classifier exactness because it provides information on how much we can trust the model when it predicts a class as positive. Hence, it is also called the positive predictive value. On the other hand, recall or sensitivity measures the completeness of classifiers because it calculates the fraction of true positives and the total number of positively classified classes. Hence, this is known as the true-positive rate. Finally, the F1-score or F-Score conveys a balance between precision and recall as a weighted average. Because macro-averaging pays attention equally to all classes, it is more reliable than accuracy in an imbalanced dataset.

Keeping that in mind, comparing CNN CECI with the multiclass approaches in Table 3, CNN CECI (0.1,0.2,0.7,0.1) outperforms LSTM multiclass approaches. For example, [31] the reported accuracy was between 21% and 23%, and 66% and a macro F1-score of 33%, respectively. In addition, this model outperformed the models in [28], both for multiclass and overall accuracy in the binary approach. The paper [34] reported a higher overall F1-score of 47%. However, that research used a binary approach.

Comparing CNN CECI with binary approaches in Table 3 and considering that the overall metric includes all four dimensions, we conclude that the results reported in [23] with the stacking and boosting approaches, random forest in [26], and XGBoost in [42] outperform CNN CECI for weight combination we obtained in experiments. However, for all other approaches, the CNN CECI has higher or comparable results in the metrics of the presented research. In addition, it is vital to emphasize again that some metrics are not consistent across studies, and some, such as accuracy, are not the best metrics for imbalanced datasets for comparison.

Table 13 provides an overview of the comparison of the F1-score results in our research and the results of related studies. Only the reported overall results are included in this table. We can see that the CNN CECI approach outperforms the best binary approach with regard to the F1-score.

Table 14 provides an overview of the comparison of accuracy results in our research and the results of related studies. Again, we included only the reported overall results in this

TABLE 13. Comparing results to related work – f1 core.

| THIS PAPER | | | | |
|--------------------------------|---------------------------------|------------|--------------------|-----------------|
| Model | $\alpha, \beta, \gamma, \delta$ | | macro F1 | weighted F1 |
| LSTM | 0.7, 0.5, 0.7, 0.6 | | 7% | 20% |
| CNN | 0.1, 0.2, 0.0, 0.25 | | 30% | 65% |
| CNN | 0.1, 0.2, 0.7, 0.1 | | 33% | 63% |
| RELATED WORK (OVERALL RESULTS) | | | | |
| Paper | Approach | Classifier | Metric | The best result |
| [34] | Binary | MLP | F1-score (overall) | 47% |

TABLE 14. Comparing results to related work - accuracy.

| THIS PAPER | | | | |
|--------------------------------|---------------------------------|----------------------------|--------------------|-----------------|
| Model | $\alpha, \beta, \gamma, \delta$ | | | Accuracy |
| LSTM | 0.7, 0.5, 0.7, 0.6 | | | 27% |
| CNN | 0.0, 0.15, 0.0, 0.6 | | | 67% |
| CNN | 0.1, 0.2, 0.7, 0.1 | | | 66% |
| RELATED WORK (OVERALL RESULTS) | | | | |
| Paper | Approach | Classifier | Metric | The best result |
| [31] | Binary | LSTM | Overall accuracy | 21% |
| [28] | Multiclass | Softmax | Accuracy | 17% |
| | | LSTM | Accuracy | 23% |
| [29] | Multiclass | LR | Accuracy | 66.59% |
| [39] | Binary | Linear/Logistic regression | Accuracy (Overall) | 45% |
| [28] | Binary | Naïve Bayes | Accuracy (Overall) | 26% |
| | | Reg. SVM | Accuracy (Overall) | 33% |
| | | LSTM | Accuracy (Overall) | 38% |

table. The results show that the CECI approach, especially on the CNN network, outperforms related works with regard to the accuracy metrics.

VI. CONCLUSION AND FUTURE WORK

This research shows how using an encoding scheme for MBTI compound labels and using a method to calculate individual probabilities for MBTI dichotomies can improve MBTI multiclass classification. Furthermore, our research included individual probabilities in a custom loss function of a neural network as a supervised machine-learning approach to achieve better multiclass classification and open new perspectives for research.

Throughout this paper, we have answered the questions we used to define the problem since the CECI method enables us to conduct MBTI multiclass classification while including all compound classes, and it helps to mitigate the overlap and unbalance problem between the compound classes.

In addition, the CECI approach showed improvement in all metrics compared to the baseline LSTM CE and CNN CE approaches. For example, we improved the macro F1-score from 27% to 33% for the CNN model, where the highest weight in CECI was 0.7 for the third dichotomy. We also improved the LSTM model with weights of 0.7 for the first and third dichotomies. In addition, the CECI approach

showed improvement compared to the present multiclass MBTI classification approaches and comparable results to present multiclass and binary approaches to MBTI classification. However, some binary approaches exhibit a slightly better performance.

Regarding the constraints and limitations of our approach, we conducted experiments using the CECI approach on one MBTI dataset. In addition, our dataset comes from one social network and contains only textual data. Therefore, experiments on other MBTI datasets from different sources, and possibly with different data types, will probably help the approach and provide new ideas regarding relations among compound class labels. In addition, in further research, one could experiment with other similar problems with compound class labels and binary values for each component. As well, our experiments were conducted on an English dataset, and the multilanguage approach could also provide new perspectives. To prove the concept, we conducted experiments on two neural network models: bidirectional LSTM and 2-dimensional CNN. Experiments on other architectures and model parameters can provide new insights and improve the method.

Future research using the CECI method will include experiments on a more balanced dataset. Furthermore, we intend to apply different techniques to handle imbalances in the MBTI dataset. In addition, our research will include cognitive functions and other relations between MBTI components and weight factors regarding the implementation of the CECI method on the MBTI dataset.

REFERENCES

- M. Mitchell, K. Hollingshead, and G. Coppersmith, "Quantifying the language of schizophrenia in social media," in *Proc. 2nd Workshop Comput. Linguistics Clin. Psychol., From Linguistic Signal Clin. Reality*, 2015, pp. 11–20.
- D. Liu, Y. Li, and M. A. Thomas, "A roadmap for natural language processing research in information systems," in *Proc. 50th Hawaii Int. Conf. Syst. Sci.*, 2017, pp. 1–10.
- S. R. Maddi, *Personality Theories: A Comparative Analysis*, 6th ed. Long Grove, IL, USA: Waveland Press Inc, 2001.
- L. R. Goldberg, "An alternative 'description of personality': The big-five factor structure," *Personality Personality Disorders, Sci. Mental Health*, vol. 7, no. 34, p. 1216, 2013.
- A. M. Bland, "The enneagram: A review of the empirical and transformational literature," *J. Humanistic Counseling, Educ. Develop.*, vol. 49, no. 1, pp. 16–31, Mar. 2010.
- D. Shaffer, M. Schwab-Stone, P. Fisher, P. Cohen, J. Placentini, M. Davies, C. K. Conners, and D. Regier, "The diagnostic interview schedule for children-revised version (DISC-R): I. Preparation, field testing, interrater reliability, and acceptability," *J. Amer. Acad. Child Adolescent Psychiatry*, vol. 32, no. 3, pp. 643–650, May 1993.
- S. Soto, "Big Five personality traits," in *The SAGE Encyclopedia Lifespan Human Development*, M. H. Bornstein, M. E. Arterberry, K. L. Fingerma, and J. E. Lansford, CA, USA: SAGE Publications, 2018, pp. 240–241.
- M. Papurt, "A study of the woodworth psychoneurotic inventory with suggested revision," *J. Abnormal Social Psychol.*, vol. 25, no. 3, p. 335, 1930.
- O. P. John, R. W. Robins and L. A. Pervin, *Handbook of Personality: Theory and Research*, 3rd ed. New York, NY, USA: Guilford Press, 2008.
- I. B. Myers and P. B. Myers, *Gifts Differing*. Palo Alto, CA, USA: Consulting Psychologists Press, 1990.
- C. G. Jung, *Psychological Types The Collected Works of C. G. Jung*, vol. 6. Princeton, NJ, USA: Princeton Univ. Press, 1976.
- L. V. Berens, *Dynamics of Personality Type: Understanding and Applying Jung's Cognitive Processes*. New York, NY, USA: Telos Publications, 2000.
- S. E. S. Duane and P. Schultz, *Theories of Personality*, 11th ed. Boston, MA, USA: Cengage Learning, 2016.
- T. L. Bess and R. J. Harvey, "Bimodal score distributions and the Myers–Briggs type indicator: Fact or artifact?" *J. Personality Assessment*, vol. 78, no. 1, pp. 176–186, Feb. 2002.
- P. T. Costa, Jr., R. R. McCrae, and D. A. Dye, "Facet scales for agreeableness and conscientiousness: A revision of the NEO personality inventory," *Pers. Individual Differences*, vol. 12, no. 9, pp. 887–898, 1991.
- A. Furnham, "The big five versus the big four: The relationship between the Myers–Briggs type indicator (MBTI) and NEO-PI five factor model of personality," *Personality Individual Differences*, vol. 21, no. 2, pp. 303–307, Aug. 1996.
- F. Celli and B. Lepri, "Is big five better than MBTI? A personality computing challenge using Twitter data," in *Proc. 5th Italian Conf. Comput. Linguistics (CLiC-IT)*, Dec. 2018, pp. 1–6.
- C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul./Oct. 1948.
- H. Ahmad, M. Z. Asghar, A. S. Khan, and A. Habib, "A systematic literature review of personality trait classification from textual content," *Open Comput. Sci.*, vol. 10, no. 1, pp. 175–193, Jul. 2020.
- B. Agarwal, "Personality detection from the text: A review," *Int. J. Comput. Syst.*, vol. 1, vol. 1, no. 1, pp. 1–4, 2014.
- M. H. Amirhosseini and H. Kazemian, "Machine learning approach to personality type prediction based on the Myers–Briggs type indicator," *Multimodal Technol. Interact.*, vol. 4, no. 1, p. 9, Mar. 2020.
- C. Li, M. Hancock, B. Bowles, O. Hancock, L. Perg, P. Brown, A. Burrell, G. Frank, F. Stiers, S. Marshall, and G. Mercado, "Feature extraction from social media posts for psychometric typing of participants," in *Proc. Int. Conf. Augmented Cognition*, 2018, pp. 267–286.
- H. P. K. Das, "Personality identification based on MBTI dimensions using natural language processing," *Int. J. Creative Res. Thoughts*, vol. 8, no. 6, pp. 1653–1657, 2020.
- S. Bharadwaj, S. Sridhar, R. Choudhary, and R. Srinath, "Persona traits identification based on Myers–Briggs type Indicator(MBTI)—A text classification approach," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2018, pp. 1076–1082.
- M. Komisin and C. Guinn, "Identifying personality types using document classification methods," in *Proc. 25th Int. Florida Artif. Intell. Res. Soc. Conf.*, Marco Island, FL, USA, May 2012, pp. 232–237.
- N. H. Z. Abidin, M. A. Remli, N. M. Ali, D. N. E. Phon, N. Yusoff, H. K. Adli, and A. H. Busalim, "Improving intelligent personality prediction using Myers–Briggs type indicator and random forest classifier," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 11, pp. 192–199, 2020.
- Z. Mushtaq, S. Ashraf, and N. Sabahat, "Predicting MBTI personality type with K-means clustering and gradient boosting," in *Proc. 23rd Int. Multitopic Conf. (INMIC)*, Bahawalpur, Pakistan, Nov. 2020, pp. 1–5.
- B. Cui and C. Qi. (2017). *Survey Analysis of Machine Learning Methods for Natural Language Processing for MBTI*. Accessed: Jun. 1, 2021. [Online]. Available: <http://cs229.stanford.edu/proj2017/final-reports/5242471.pdf>
- S. Chaudhary1, R. Singh, S. T. Hasan, and M. I. Kaur, "A comparative study of different classifiers for Myers–Brigg personality prediction model," *Int. Res. J. Eng. Technol.*, vol. 5, no. 5, pp. 1410–1413, 2018.
- Y. Mehta, N. Majumder, A. Gelbukh, and E. Cambria, "Recent trends in deep learning based personality detection," *Artif. Intell. Rev.*, vol. 53, no. 4, pp. 2313–2339, Apr. 2020.
- R. Hernandez and I. S. Knight, "Predicting Myers–Briggs type indicator with text classification," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, long beach, CA, USA, Dec. 2017, pp. 4–9. Accessed: Jul. 15, 2021. [Online]. Available: <https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6839354.pdf>
- N. Patel. (2021). *17 Charts That Show Where Social Media is Heading*. Accessed: Jul. 26, 2021. [Online]. Available: <https://neilpatel.com/blog/social-media-trends/>
- S. Barrett. (May 30, 2021). *How Much Data is Produced Every Day in 2021*. Accessed: Jul. 28, 2021. [Online]. Available: <https://the-tech-trend.com/reviews/how-much-data-is-produced-every-day/>
- M. Gjuroković and J. Šnajder, "Reddit: A gold mine for personality prediction," in *Proc. 2nd Workshop Comput. Model. People's Opinions, Personality, Emotions Social Media*, 2018, pp. 87–97.

- [35] N. Alsadhan and D. Skillicorn, "Estimating personality from social media posts," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2017, pp. 350–356.
- [36] B. Plank and D. Hovy, "Personality traits on Twitter—Or—How to get 1,500 personality tests in a week," in *Proc. 6th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, 2015, pp. 1–7.
- [37] J. Golbeck, C. Robles, M. Edmondson, and K. Turner, "Predicting personality from Twitter," in *Proc. IEEE 3rd Int. Conf. Privacy, Secur., Risk Trust IEEE 3rd Int. Conf. Social Comput.*, Oct. 2011, pp. 149–156.
- [38] S. Garg, P. Kumar, and A. Garg, "Comparison of machine learning algorithms for content-based personality resolution of tweets," *Social Sci. Humanities Open*, vol. 4, no. 1, Jan. 2021, Art. no. 100178, doi: [10.2139/ssrn.3626306](https://doi.org/10.2139/ssrn.3626306).
- [39] M. Gjurković, M. Karan, I. Vukojević, M. Bošnjak, and J. Snajder, "PAN-DORA talks: Personality and demographics on Reddit," in *Proc. 9th Int. Workshop Natural Lang. Process. Social Media*, 2021, pp. 1–15.
- [40] D. Wan, C. Zhang, M. Wu, and Z. An, "Personality prediction based on all characters of user social media information," in *Proc. Chin. Nat. Conf. Social Media Process.*, Beijing, China, 2014, pp. 220–230.
- [41] L. C. Lukito, A. Erwin, J. Purnama, and W. Danokoesoemo, "Social media user personality classification using computational linguistic," in *Proc. 8th Int. Conf. Inf. Technol. Electr. Eng. (ICITEE)*, Oct. 2016, pp. 1–6.
- [42] A. S. Khan, H. Ahmad, M. Zubair, F. Khan, A. Arif, and H. Ali, "Personality classification from online text using machine learning approach," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 3, pp. 460–476, 2020.
- [43] E. J. Choong and K. D. Varathan, "Predicting judging-perceiving of Myers-Briggs type indicator (MBTI) in online social forum," *PeerJ*, vol. 9, Jun. 2021, Art. no. e11382, doi: [10.7717/peerj.11382](https://doi.org/10.7717/peerj.11382).
- [44] V. Lynn, N. Balasubramanian, and H. A. Schwartz, "Hierarchical modeling for user personality prediction: The role of message-level attention," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 5306–5316.
- [45] B. Verhoeven, W. Daelemans, and A. B. Plank, "TwiSty: A multilingual Twitter stylometry corpus for gender and personality profiling," in *Proc. 10th Lang. Resour. Eval. Conf. (LREC)*, Portoroz, Slovenia, May 2016, pp. 1632–1637.
- [46] F. Liu, J. Perez, and S. Nowson, "A language-independent and compositional model for personality trait recognition from short texts," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics, Long Papers*, vol. 1, 2017, pp. 754–764.
- [47] B. Y. Pratama and R. Sarno, "Personality classification based on Twitter text using naive Bayes, KNN and SVM," in *Proc. Int. Conf. Data Softw. Eng. (ICoDSE)*, Nov. 2015, pp. 170–174.
- [48] N. Majumder, S. Poria, A. Gelbukh, and A. E. Cambria, "Deep learning-based document modeling for personality detection from text," *IEEE Intell. Syst.*, vol. 32, no. 2, pp. 74–79, Mar. 2017.
- [49] E. P. Tighe, J. C. Ureta, B. A. L. Pollo, C. K. Cheng, and A. R. D. D. Bulos, "Personality trait classification of essays with the application of feature reduction," in *Proc. 4th Workshop Sentiment Anal. Where AI Meets Psychol. (SAAIP), (IJCAI)*, New York, NY, USA, 2016, pp. 22–28.
- [50] E. Ezpeleta, I. V. de Mendizabal, J. M. G. Hidalgo, and U. Zurutuza, "Novel email spam detection method using sentiment analysis and personality recognition," *Log. J. IGPL*, vol. 28, no. 1, pp. 83–94, Jan. 2020.
- [51] E. Ezpeleta, U. Zurutuza, and J. M. G. Hidalgo, "Using personality recognition techniques to improve Bayesian spam filtering," *Procesamiento del Lenguaje Natural*, vol. 57, pp. 125–132, Sep. 2016.
- [52] *Kaggle MBTI Dataset*. Accessed: Mar. 1, 2021. [Online]. Available: <https://www.kaggle.com/datasnaek/mbti-type/download>
- [53] T. M. B. Foundation. (2021). *The Myers & Briggs Foundation*. Accessed: Jul. 8, 2021. [Online]. Available: <https://www.myersbriggs.org/my-mbti-personality-type/my-mbti-results/how-frequent-is-my-type.htm>
- [54] M. Frkovic, N. Cerkez, B. Vrdoljak, and S. Skansi, "Evaluation of structural hyperparameters for text classification with LSTM networks," in *Proc. 43rd Int. Conv. Inf., Commun. Electron. Technol. (MIPRO)*, Sep. 2020, pp. 1145–1150.
- [55] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.



NINOSLAV CERKEZ (Member, IEEE) was born in Zenica, Bosna and Herzegovina, in 1971. He received the B.S. and M.S. degrees in engineering from the University of Zagreb, Croatia, in 1998 and 2006, respectively, and the M.B.A. degree from IGBS, Indiana University in 2011. He is currently pursuing the Ph.D. degree in computer science with the University of Zagreb.

From 1999 to 2018, he had different technical and HR roles in the IT industry. Also, he was working as a part-time Lecturer at IT colleges in Zagreb. Since 2018, he has been a Lecturer at the IT College in Zagreb. In addition, from 2018 to 2020, he was the Vice-Dean at the IT College. His research interest includes applied data science to the psychology of personality.

Mr. Cerkez became a member of SHRM, in 2014; HRCI, in 2014; and PMI, in 2007. He holds PMP, SHRM-SCP, and HRCI professional certificates.



BORIS VRDOLJAK (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Faculty of Electrical Engineering and Computing (FER), University of Zagreb, Zagreb, Croatia, in 1995, 1999, and 2004, respectively.

He has been at the Faculty of Electrical Engineering and Computing, University of Zagreb, since 1996. He spent three months as a Visiting Researcher at the University of Bologna, Italy, in 2001. From October 2004 to September 2005, he was a Postdoctoral Researcher at the INRIA Institute, France. He is currently a Full Professor with the Faculty of Electrical Engineering and Computing, University of Zagreb. His research interests include data warehousing, big data analytics, automation of ontology matching, text classification, e-business, and information security.



SANDRO SKANSI was born in Zagreb, in 1985. He attained his M.A. in Philosophy and Croatian Culture in 2009 at the University of Zagreb, and completed a thesis in mathematical logic at the University of Zagreb in 2013.

Since 2017, he has been an Assistant Professor in logic with the University of Zagreb. His research interests include the satisfiability of propositional logic, graph theory, fuzzy logic, cybernetics, and artificial neural networks.

Mr. Skansi has been a member of the Association for Symbolic Logic, since 2006, and a Life Member of the Association for the Advancement of Artificial Intelligence, since 2016, and the National Rifle Association.

• • •