IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 August 2024, accepted 16 September 2024, date of publication 18 September 2024, date of current version 30 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3463400

TOPICAL REVIEW

The Applicability of LLMs in Generating Textual Samples for Analysis of Imbalanced Datasets

SAROJ GOPALI[1], **FARANAK ABRI**[2], **AKBAR SIAMI NAMIN**[1], **AND KEITH S. JONES**[3]
[1]Department of Computer Science, Texas Tech University, Lubbock, TX 79409, USA
[2]Department of Computer Science, San Jose State University, San Jose, CA 95192, USA
[3]Department of Psychological Sciences, Texas Tech University, Lubbock, TX 79409, USA

Corresponding author: Saroj Gopali (saroj.gopali@ttu.edu)

This work was supported in part by U.S. National Science Foundation under Award 2319802 and Award 2319803, and in part by U.S. Office of Naval Research under Award N00014-21-1-2007.

ABSTRACT In machine learning class imbalance is a pressing issue, where the model is biased towards the majority classes and underperforms in the minority classes. In textual data, the natural language processing (NLP) model bias significantly reduces overall accuracy, along with poor performance in minority classes. This paper investigates and compares the performance of transformer-based models, such as Multi-head Attention with the data levels and algorithmic levels approaches and BERT (Bidirectional Encoder Representations from Transformers) with LLM-based data augmentation. The research utilized the approaches, such as Random Over Sampler, Synthetic Minority Over-sampling Technique (SMOTE), SMOTEENN, data augmentation at word level, class weights, L2 regularization and leveraging GPT-3.5-Turbo's for data augmentation to create additional data samples in imbalance dataset. The results from the experiment demonstrate that the LLM-based data augmentation with Multi-head Attention and BERT in the Myers-Briggs Type Indicator (MBTI) dataset (a highly skewed dataset) achieves the highest precision, recall and F1 score of 0.76 across terms. It indicates that the LLM-based data augmentation has significant improvements in dealing with class imbalance and improves the model's accuracy in minority class types in the MBTI dataset.

INDEX TERMS Multi-head attention, BERT, LLM, GPT 3.5-turbo, imbalance dataset, Myers-Briggs type indicators.

I. INTRODUCTION

In real-world scenarios, the working datasets do not have the balance nor uniform distribution of classes, which is one of the challenges that machine learning models often deal. The lack of a balanced dataset effects machine-learning models and their performance across various application domains including medical diagnosis, fraud detection, and text classification. The main issue with the imbalanced dataset is that training based on an imbalanced dataset might result in a biased or inaccurate model prediction in minority classes. The models more likely tend to favor the majority class, which can result in elevating the model performance superficially. However, the minority classes are overlooked during training. This will mislead the model's overall performance with the minority classes exhibiting very poor accuracy.

In the evolving field of artificial intelligence along with the rapid growth in deep learning algorithms and Large Language Models (LLMs), balanced datasets play a critical role in fair and unbiased model development. The highly skewed imbalance dataset leads to the representation of the distribution of classes which has a severe impact on the performance of the model, particularly in tasks such as text classification [37], natural language processing [24], and predictive analysis. The challenge of training machine learning models based on imbalanced data becomes more daunting when dealing with textual data, where nuance in language and expression has a major impact on the context of the learning process and the ultimate accuracy of models.

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad J. Abdel-Rahman.

The existing data resampling techniques address the imbalance in datasets using approaches to oversampling the minority class or undersampling the majority class. The approaches of oversampling and undersampling such as Synthetic Minority Over-sampling Technique (SMOTE) [31], SOMOTEEN [5], and Random OverSampler [10] have limitations when applied to textual data, due to the complexity and variability of natural languages. At the algorithmic level such as Class weights [27], and L2 regularization [8] are employed to address issues from the model level rather than the data level. These methods are combined to improve the robustness and performance by treating all classes equally in weight and penalizing large weights. Both approaches are essential in improving the performance and generalizability of machine learning models.

The recent advancement of Large Language Models (LLMs) such as GPT (Generative Pre-trained Transformer) [38], transformer based model, including BERT (Bidirectional Encoder Representations from Transformers) [17], and their derivatives have revolutionized ways to approach language-related tasks in deep learning. In the context of LLMs, these models are trained in vast corpora of data, which can understand and maintain the context across the text segments. The capacity to generate textual context, which is diverse, cohesive, and contextually relevant, presents a potential solution to address the issue of imbalanced datasets in textual contexts.

In addition, the transformer-based [42] models are powerful NLP techniques beneficial for handling imbalanced datasets due to their unique capabilities. Multi-head attention [42] enables learning diverse contextual information, allowing the model to identify patterns specific to minority classes. BERT's contextualized word representations consider bidirectional understanding, providing deep insights into the classification of minority class samples. BERT's transfer learning and fine-tuning capabilities enable adaptation to specific tasks with fewer examples, addressing the imbalance by leveraging pre-trained knowledge.

This research explores the enhancement of natural language processing (NLP) models, specifically focusing on addressing the challenges posed by imbalanced datasets in text classification. We investigate the integration of advanced data level including data augmentation and algorithmic level techniques with Multi-head Attention. In addition, we fine-tuned the BERT Model leveraging the pre-trained knowledge of the model for the text classification. Similarly, we leverage the power of LLMs (i.e GPT 3.5 turbo) for data augmentation with a combination of both models (i.e. Multi-head Attention and BERT). The models are chosen based on their abilities to understand the context in the textual data. The combination of these models with the LLM-based augmentation can significantly improve the representation and classification of minority classes, often underrepresented in textual data.

As a case study, we conduct several experiments to validate these theoretical assumptions in the textual contexts. Personality trait detection in the textual contexts is an important case study for the application of LLMs in imbalanced datasets. The dataset context of various textual sources such as tweets, social media posts or interview transcripts which are labeled with personality traits based on the frameworks like Big Five [33], Myers-Briggs Type Indicator [6].

The purpose of this research is to utilize deep learning models along with LLM-based data augmentation approaches to classifier Myers-Briggs Type Indicator (MBTI) of personality traits while dealing with class imbalance. We focus on conducting MBTI multi-class classification where our expected outcome will be a predicted MBTI label for textual data in this experiment tweet's posts. The key contributions of this paper are as follows:

1) We used LLM-based data augmentation (i.e. GPT 3.5 Turbo) with control prompt, combined with BERT and Multi-Head Attention models to enhance performance, demonstrating their potential benefits in the problem studied in this paper (i.e., the MBTI dataset). The experiments demonstrated that LLM-based data augmentation with BERT achieves the highest model accuracy, with precision, recall, and F1-score reaching 0.76, outperforming other techniques (i.e., data level, algorithmic level, and data augmentation at word level).
2) We conducted empirical studies in building and training Multi-head Attention models with existing techniques such as data-level approaches (i.e., SMOTE, SMOTEENN, and Random Over Sampler), algorithmic-level approaches (i.e., Class weights, L2 regularization) and Data Augmentation with nlpaug [1] at word-level to deal with class imbalance. The results showed these approaches improved in minority class representation. However, our proposed approach in implementing LLM-based augmentation with BERT achieved higher performance, validating the efficacy of the introduced technique.
3) We reported the model architecture comparison such as Mult-Head Attention and BERT, and the results of with or without data augmentation. Hence, the utilization of LLMs for data augmentation improved not only accuracy but also shed lights on the strengths and limitations of such models in the context of highly imbalanced datasets.
4) We tested the proposed approach on the highly imbalanced MBTI dataset which proved our proposed methods are feasible and effective. The empirical studies show that the proposed LLM-based data augmentation addresses a major problem in text classification tasks of improving the predictive accuracy of models especially in underrepresented classes.

The rest of the research organized in sections. Section II contains the case study on personality trait detection. The section III describes the research objective. The section IV presents the existing related works. Sections V and VI contain the technical background and method employed in this study. The section VII contain experimental procedures. The section VIII presents the results and section IX explains textual data augmentation from the experiments. Sections X and XI present the discussion and conclusion of the research.

II. THE CASE STUDY: PERSONALITY TRAIT DETECTION—A HIGHLY IMBALANCED APPLICATION DATA

In this evolving technological era, personality traits play vital role to understand humans, through the information about a person's emotions, preferences, areas of interest, and motives in a methodical manner. Personality describes a distinct pattern of ideas, actions, and feelings that identifies a person. The personality type prediction in psychology has significant application in many areas such as businesses [23], [30] health care [12], [44] mental health screening exams, screening during job interviews and so on. Such analyses are also applicable in education sectors, which also offers educators crucial knowledge that helps them better improve the personalities of Students [21].

Self-evaluation and question answers are examples of common methods used in psychology to measure personality. The method is trustworthy if it can confirm consistency in measured values with a reasonable variation. However, because this includes surveys and highly skilled individuals, conventional procedures in psychology for personality type evaluation are time-consuming and expensive. As a result, even though there are excellent reasons to think, personality tests should be helpful in various research. There has been little concrete evidence of their effectiveness.

The use of data and leveraging machine learning or deep learning models that can predict and classify personality characteristics based on textual documents has been one of the most interesting applications of psychology studies. The utilization of machine learning techniques enables the classification of personality traits by gathering a larger number of data to find a pattern to classify these characteristics based on various factors [45]. Machine learning methodologies provide an opportunity to advance and classify personality assessments that would otherwise be harder or impossible to achieve using traditional approaches [47]. The Myers-Briggs Type Indicator, a personality assessment tool developed by Myers-Briggs, is based on Carl Jung's idea of psychological types from 1921 [6].

Myers-Briggs assesses personality types and preferences using four characteristics of personality based on the psychological kinds identified by the Jungian system: 1) Extraversion (E) vs. Introversion (I), 2) Sensing (S) vs. Intuition (N), 3) Thinking (T) vs. Feeling (F), 4) Judging (J) vs. Perceiving (P).

III. THE RESEARCH OBJECTIVES OF THIS STUDY

The fundamental goal of this research is to improve the performance and fairness of natural language processing (NLP) model algorithms that deal with imbalanced text datasets. In the imbalanced datasets, certain classes are underrepresented, providing substantial issues in training deep learning models, which leads to bias and poor performance in minority classes. This research aims to address these challenges by combining LLM-based data augmentation to ensure more training and better model prediction in minority classes. Specifically, we explore the effectiveness of Multi-head Attention and BERT models in capturing subtle textual features and their interaction with various data and algorithmic levels techniques to enhance minority class representation and model accuracy.

In addition, one of the objectives is to systematically evaluate the effect of various data-level approaches on the model performance trained on imbalanced datasets. The approaches include such as SMOTE, SMOTEENN, and Random Over Sampler. It also includes more recent methods such as word-level augmentation with nlpaug and synthetic data generation with GPT-3.5-Turbo. Furthermore, in this study we intend to fine-tune BERT models on the MBTI dataset, to understand the complexities of model performance across diverse domains and imbalance scenarios. The research questions that this paper aims to address include:

1) How do different data level techniques, particularly those leveraging advanced NLP and LLM tools like nlpaug and GPT-3.5-Turbo, impact the balance of datasets and the performance of NLP models on minority classes?
2) What is the impact of using Multi-head Attention model and fine tuned BERT models on the classification accuracy in imbalanced text datasets?
3) How do algorithm-level techniques, such as class weights and L2 regularization, compare with data-level augmentation approaches in addressing imbalance?
4) How does the combination of Multi-head Attention and BERT models with LLM-based data augmentation impact in handling imbalanced datasets and improving the representation of minority classes?

IV. RELATED WORK

A. MACHINE LEARNING APPROACHES

Komiisin et al. [29] conduct an experiment test made of MBTI if a person's personality type can be determined from their word choice using probabilistic and non-probabilistic classifiers. Linguistic Inquiry and Word Count (LIWC), a third-party text analysis tool, is used to extract emotional, social, cognitive, and psychological characteristics for classification. For the dataset, the information was gathered over the period of three semesters in 2010 and 2011 as a graduate-level conflict management course. The findings of the Myers-Briggs Type Indicator Step II (MBTI) plus the essays from the Best Possible Future Self (BPFS) make up the dataset. For binary MBTI classification, the study employed naive Bayes and support vector machine (SVM) techniques. The SVM performed worse than the naive Bayes technique, which had precision and recall values higher than 75%.

Zeeshan et al. [34] propose to adopt K-Means clustering and gradient boosting, two well-known machine learning methods, along with the traditional Term Frequency-Inverse Document Frequency (TF-IDF) approach to predict the MBTI personality type. The authors utilized the MBTI data set obtained from Kaggle which has two columns and 8675 rows. The final accuracy of each classifier after the hyper-parameter was high at 89.01%. The provided model had a greater performance with an average accuracy of 86.3%. However, the approach is unable to predict the extroversion nature in the context of classification.

Abidin et al. [3], a random forest classifier is the most effective and useful method for categorizing MBTI binary classification. Word2vec was utilized for word vector representations and extra features, such as words per comment. The dataset for this study has two columns and over 8675 rows of data which was gathered from the Kaggle data archive. Using the testing dataset, the accuracy of the Random Forest and three other models Linear Regression, KNN neighbor, and Support Vector Machine (SVM) was evaluated. All dichotomies reported an accuracy of 100% using a random forest algorithm. The author used 90% of the data for training and the remaining 10% for testing. However, our interest is more toward a multi-class approach and this paper does not include other model evaluation criteria that are crucial for classifying imbalanced datasets.

B. DEEP LEARNING APPROACHES

Cui and Qi [16] present an experiment to classify MBTI using both a deep learning approach and traditional supervised machine learning. The Myers-Briggs Personality Type Dataset (MBTI), has 8,600 individuals with MBTI personality types and the text of their 45-50 social media postings. Additionally, both multi-class and binary classification strategies have been adopted in the study. A multi-layer long-short-term memory (LSTM) recurrent neural network was utilized as the encoder in the encoding system, with rectified linear units (ReLU) being employed for each activation. The last layer used a softmax function to output the probability of each class. With the test accuracy of 23% for multi-class classification, they recorded the best LSTM network result. The maximum test accuracy for the binary strategy was 38%.

Amirhosseini et al. [4], propose a novel machine-learning technique for automating the meta-program identification and personality type prediction based on the MBTI personality type indicator. In this study, 8675 rows of the Myers-Briggs personality type dataset from Kaggle were used. Each MBTI type is composed of four binary classes, the classification problem was divided into 16 classes and then into four binary classification tasks. The MBTI-type indicators were binarized and the term frequency-inverse document frequency (TF-IDF) was used. A training set of 70% of the data and a test set of 30% of the data were utilized in the study. Here, the development method employs machine learning techniques within the Gradient Boosting framework which had an accuracy of 86.06% compared to 62% from Recurrent Neural Network.

Chen et al. [11] offers a unique strategy that uses contrastive samples to solve data imbalance in text classification. The authors process produces high-quality positive samples by using a Label-indicative Component (LIC) that improves performance and reduces distributional skews. The author conducted tests on four text classification datasets, including three simulated benchmark datasets based on THUCNews, AG's News, 20NG, and the imbalanced FDCNews dataset. The first three datasets are balanced, thus the author constructed them with some imbalanced labels. As an encoder model, the author employed the CNN and BERT pre-training models. Encoder (CNN or BERT)+LIC model studies indicate that producing positive samples for the minority class can increase classification task accuracy and Macro-F1 by up to 1%. However, the author's works require further improvement in the quality of constructed samples. Also, the data set used was not imbalanced initially and needs exploration with real data with imbalanced data.

In financial few-shot text classification tasks, Loukas et al. [32] present that fine-tuned LLMs can beat fine-tuned Masked Language Models (MLMs), even when given fewer instances. The author did a thorough investigation on the trade-offs between stand performance and LLMs and MLMs in a few-shot text classification in Banking77, a financial intent detection dataset. The dataset is divided into two subsets: train (10,003 examples) and test (3,080 samples), containing a total of 13,083 annotated customer service questions labeled with 77 intents. In the training subset, the label distribution is significantly imbalanced. The research demonstrated the effectiveness and efficiency of in-context learning using conversational LLMs.

C. SOME OTHER APPROACHES

There exists some other approaches [18], [19] for addressing challenges in machine learning and optimization in classification problems. As an example, Gong et al. [19] presents a quantum K-nearest neighbor (QKNN) algorithm that is a divide-and-conquer operation to increase the rate of classification in a higher dimensionality. The QKNN algorithm achieves the average classification accuracy of approximately 97.04% on the IRIS dataset by utilizing quantum circuits. Another research work by Gong et al. [18] presents the Diversity Migration Quantum Particle Swarm Optimization (DM-QPSO) algorithm that is an improvement of the QPSO with the aim of resolving highly complicated optimization problems than the standard PSO algorithms. Both methods outperform their constructive classical counterparts in performance and point out that quantum computing in advancing these fields.

V. TECHNICAL BACKGROUND

A. MULTI HEAD ATTENTION

The transformer approach is a type of neural network architecture that has been used for Natural Language Processing

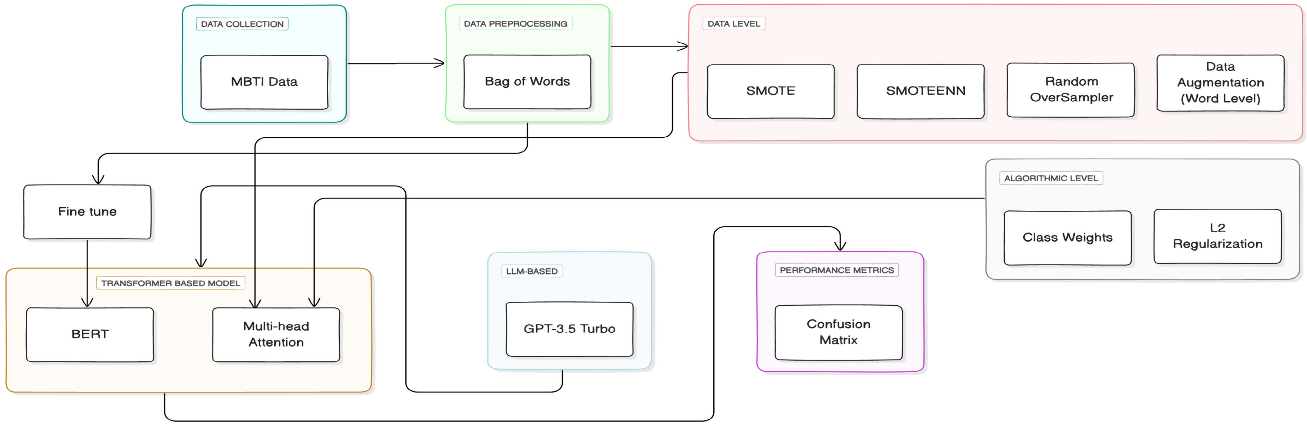


FIGURE 1. Methodology combining data preprocessing, transformer-based models and performance metrics.

(NLP) tasks such as text classifications. The transformer-based [42] model leverages the self-attention mechanism to process input sequences, allowing it to capture long-range dependencies in the data and improve the performance of natural language processing tasks. Multi-head attention enables the transformer to encode multiple relationships and nuances for each word. Multi-head attention enables the neural network to manage the blending of data between elements of an input sequence, creating richer representations and improving performance on machine learning tasks. The attention module in the Transformer performs its computations in parallel. Each of these is known as an Attention Head. The attention module divides its parameters into Query, Key, and Value. The sum of all comparable attention calculations results in a final attention score.

In the context of text classification, a transformer model would take in a sequence of words or tokens as input and output a predicted class label for the text. The self-attention mechanisms in the model would help it to identify important words and their relationships to each other in order to make a more accurate classification.

B. BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS

Bidirectional Encoder Representations from Transformers (BERT) is a natural language processing model developed by Google researchers [17]. BERT is designed to capture the contextual relationships between words in a sentence by utilizing a deep bidirectional representation, allowing it to understand the meaning of words in their context.

One key advantage of BERT is its ability to perform well on a wide range of tasks, including text classification, natural language inference, and question answering [17]. This flexibility is due to BERT's use of a transformer architecture, which allows it to effectively capture long-term dependencies in text.

Another notable feature of BERT is its use of unsupervised pre-training on large corpora of text. This pre-training allows the model to learn general-purpose language representations, which can then be fine-tuned for specific tasks. BERT has achieved state-of-the-art results on a wide range of tasks, including question answering, natural language inference, sentiment analysis, and named entity recognition [46].

C. GPT 3.5 Turbo

OpenAI-developed type of Generative Pre-trained Transformer 3 `GPT-3.5-Turbo` [36], represents an advanced version of Large language models designed to generate human-like text. GPT-3.5 Turbo is a 175B-parameter model containing capabilities including natural language and code comprehension and design. The GPT-3.5 Turbo adjusts 4,096 tokens per interaction as inputs. The model, developed for chat interactions, boasts exceptional capabilities while being remarkably cost-effective, priced at only one-tenth of the cost of the `text-davinci-003` model. In the research, we employ `GPT-3.5-Turbo` with the custom prompt for data augmentation in minority classes.

VI. METHODOLOGY

This section presents the details methodology employed during the experiment in the study. Figure 1 demonstrates the flow chart of the methodology. The process begins with data collection (i.e, MBTI data). The imbalance datasets are then prepared in the data preprocessing stage, where we apply Bag of Words (BoW) [9] to effectively capture the frequency of words. In the data-level stage, considering the class imbalance in the personality classification tasks, we incorporate several techniques such as SMOTE and SMOTEENN that create new synthetic samples for each minority class and eliminate noise. These methods are chosen to enrich the samples belonging to the minor classes and preserve the integrity of the dataset.

Moreover, we apply the Randon OverSampler which oversamples the instances randomly within the minority classes for improving the distribution of the dataset. As a result, we assess word-level data augmentation to enhance the variety and enrichment of the input data and further improve the generalization capability of the model. At the algorithmic-level, to optimize the model performance and address the overfitting problem, we employed class weights and L2 regularization. The class weights are optimized when training the model to keep the model responsive to the minority class thereby preventing overemphasizing the majority class. L2 regularization is used to penalize the coefficients from growing big which helps in maintaining the model to balance between high variance and generalization.

In the LLM-based augmentation stage, we utilized the GPT-3.5 Turbo model to generate additional samples in minority classes. The data-level and algorithmic-level and LLM-based augmentation utilized the Multi Head Attention model. The BERT model is fine tuned without data-level and algorithmic-level approaches. In addition, BERT model trained in the LLM based augmentation. Both the transformer-based model valued using a confusion matrix, which provides metrics such as accuracy, precision, recall and F1 score. The methodology designed for improving the personality classification with increased scalability and reliability particularly under the settings of the MBTI dataset.

TABLE 1. Overview of methodologies employed in studies addressing class imbalance.

Study	Methodology
Komiisin et al. [29]	Naive Bayes and support vector machine (SVM) techniques.
Zeeshan et al. [34]	K-Means clustering and gradient boosting, along with the traditional Term Frequency-Inverse Document Frequency (TF-IDF).
Abidin et al. [3]	Random Forest Linear Regression, KNN neighbor, and Support Vector Machine (SVM), along with Word2vec.
Cui et al. [16]	Long Short- term memory (LSTM), with rectified linear units (ReLU) as activation.
Amirhosseini et al. [4]	Natural language processing toolkit (NLTK) and XGBoost.
Chen et al. [11]	Encoder (CNN or BERT)+LIC model to produce positive samples for the minority class.
Loukas et al. [32]	LLMs and MLMs in a few-shot text classification.
Gong et al. [19]	Quantum K-nearest neighbor (QKNN) algorithm.
Gong et al. [18]	Diversity Migration Quantum Particle Swarm Optimization (DM-QPSO) algorithm.
Our Methodology	Multi Head Attention and BERT with LLM Based Augmentation (i.e GPT 3.5 Turbo) with control prompt.

Table 1 reports the studies that have addressed the problem of class imbalance utilizing various methodologies. The approaches begins with the conventional methods like Naive Bayes and support vector machines (SVM) used by Komiisin et al. [29], and extends up to the K-Means clustering with the gradient boosting with the Term Frequency-Inverse Document Frequency (TF-IDF) by Mushtaq et al. [34]. In addition, the table includes tasks performed with few-shot text classifications with LLMs by Loukas et al. [32], and quantum algorithms like QKNN and DM-QPSO by Gong et al. [18], [19]. The authors of this work propose their novel approach combining Multi Head Attention and BERT associated with LLM Based Augmentation with GPT 3. 5 Turbo, with control prompt to improve the model performance on the MBTI dataset. The diverse set of methodologies presents in table 1 shows of the continually evolving nature of approaches towards class imbalance across domains.

VII. EXPERIMENTAL SETUP

This section describes the data set, data preprocessing, model architecture, and the assessment metrics employed for the experiment.

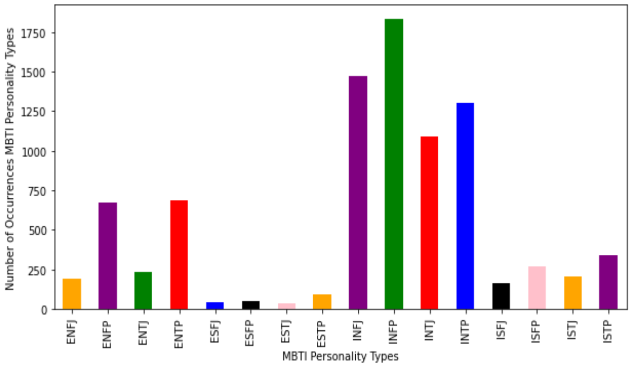


FIGURE 2. The personality types distribution in MBTI dataset.

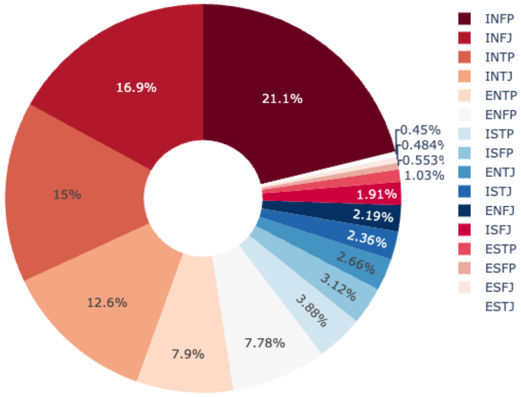


FIGURE 3. MBTI personality types.

A. DATASET

For this experiment, we used the Kaggle MBTI dataset,[1] which consisted of 8,675 rows of posts from the PersonalityCafe forum labeled with 16 MBTI personality types for their authors. The dataset is highly skewed and not evenly distributed across the 16 personality types as demonstrated in Figure 2. The Myers-Briggs Type Indicator (MBTI) [35] is a psychological assessment tool that uses 16 different personality types to categorize people. The classification system is based on four axes and consists of a four-letter code, with each letter referring to the dominant trait in each axis.

- Extroversion (E) / Introversion (I)
- Sensing (S) / Intuition (I)
- Thinking (T) / Feeling (F)
- Judging(J) / Perceiving (P)

This dataset was chosen because it has exceptionally imbalanced characteristics and provides well-defined classes for multiclass text classification. For instance from Figure 2 and Figure 3, we can observe that the classes INFP, INFJ, INTP, and INTJ have the highest percentages and type counts, 21.1% or 1832, 16.9% or 1470, 15% or 1304 and 12.6% or 1091, respectively. The classes, ESTP, ESFP, ESTJ, and ESFJ have the lowest type counts and percentages, 89 or 1.03%, 48 or 0.55%, 39 or 0.48% and 42 or 0.45%, respectively.

[1] https://www.kaggle.com/datasets/datasnaek/mbti-type

B. DATA PREPARATION

In the Data Preparation Phase, the preprocessing was done on the textual data in column 'posts' to improve feature extraction. Using NLTK, spaCy python libraries, lowercase was applied to the textual data. The stopwords, and special characters were removed during the pre-process stage. Textual data was similarly split and stripped. To create a bag of words representation, we employed sklearn library CountVectorizer to tokenize the 'posts' column in the dataset. The goal is to convert the text (character strings) document into a sequence of unique integer values.

C. TOOLS AND LIBRARIES USED

During the experiment, we used several Python libraries, including Keras.Preprocessing, spaCy [22] and Natural Language Toolkit (NLTK) [7] for pre-processing the text. For visualization we used matlibplot [25], seaborn [43] and ploty [26]. To deal with the imbalanced dataset, we have implemented SMOTE - Synthetic Minority Over-sampling Technique [10] imbalanced-learn is a Python package [31] that provides a SMOTEENN, Ramdon Oversampling. For the LLM argumentation we utilized GPT 3.5 Turbo [36]. The keras [13] with tensorflow [2] libraries utilized in the model building. Keras library is used to build the model architecture of deep learning models (i.e., MultiHead Attention). To compute performance metrics, we used the sklearn library.

D. MODEL ARCHITECTURE

For the experiment, the multi-head attention deep learning model was implemented using the Keras library with TensorFlow as the backend. In this model, the token embedding size was set to 120, the number of attention heads was 2, and the hidden layer size in the feedforward network inside the transformer was set to 8. The batch size during training was 64 and the number of training epochs was 30. The maximum length of the input sequence and the size of the vocabulary were both set to 40,000.

The input layer for the model has a shape of max_seq_length of 482, where max_seq_length is the maximum length of the input sequence. It then applies a TokenAndPositionEmbedding layer to the input, which learns an embedding for each token in the input sequence.

The output of the embedding layer is then passed through a TransformerBlock layer, which applies the transformer algorithm to the input sequence. The output of the transformer block is then passed through a global average pooling layer, which averages the values for each token in the input sequence. The resulting tensor is then passed through a dropout layer and a dense layer, which applies a softmax activation function to the tensor to produce the final outputs for the model. Finally, the model uses the Adam optimizer, the categorical cross-entropy loss function, and the accuracy metric. The configuration of transformer models is set based on observations during the model training period to enhance the model robustness.

E. PERFORMANCE METRICS

Model performance is evaluated using precision, recall, and the $F1$ score measure, which are common classification metrics. These metrics are calculated using a confusion matrix that contains counts of samples classified as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

F. ENTROPY

Entropy is a concept used in information theory [39] as well as machine learning. Entropy is a measure of the uncertainty or randomness of a dataset or signals in information theory. It is frequently used as a loss function in machine learning to measure the difference between predicted and true probabilities. Entropy represents the average amount of information conveyed by each symbol in a dataset in the context of information theory. The entropy formula.

$$H = -\sum_{x} p(x) \log_2 p(x) \tag{1}$$

where H is the entropy, p(x) is the probability of symbol x appearing in the dataset, and the sum is calculated across all possible symbols in the dataset.

In handling imbalanced datasets, entropy can be used to select features of data by identifying the most informative attributes that distinguish between classes. The features selected through this entropy-based method can improve the performance of the model as it focuses on the variable that best separates the minority class from the majority class. The model can be more effective in identifying underrepresented classes when trained with reduced dimensional features.

VIII. RESULTS

In the experiment, 16 different personality types from a total dataset of 8,675 have taken. The training dataset contains 80% and the remaining 20% of the entire dataset is used for testing. Table 2 reports the results of the original dataset with the Multi-Head attention base model for each category of imbalanced data, where the model has a weighted average F1-score of 0.67 and a macro average F1-score of 0.53.

The experiment contains the data level technique of sampling including random oversampler, SMOTE, SMOTEENN, and textual data augmentation using LLMs. At the algorithmic level (i.e., model level), the class weights and L2 regularization technique are employed.

A. IMPROVING RESULTS WITH DATA LEVEL IN IMBALANCED DATASETS

1) RANDOM OVERSAMPLER

Random Oversampling is a data preprocessing method used to balance class distributions in a dataset by oversampling the minority class. It involves randomly selecting examples from the minority class and duplicating them until the minority

TABLE 2. Classification report multi-head attention base model.

Class	Precision	Recall	F1-score	Support
INFJ	0.62	0.66	0.64	38
ENTP	0.55	0.65	0.60	135
INTP	0.71	0.37	0.49	46
INTJ	0.62	0.69	0.66	137
ENTJ	0.50	0.11	0.18	9
ENFJ	0.00	0.00	0.00	10
INFP	0.43	0.38	0.40	8
ENFP	0.50	0.39	0.44	18
ISFP	0.71	0.74	0.72	294
ISTP	0.77	0.73	0.75	366
ISFJ	0.64	0.71	0.67	218
ISTJ	0.77	0.65	0.70	261
ESTP	0.64	0.55	0.59	33
ESFP	0.45	0.54	0.49	54
ESTJ	0.36	0.59	0.45	41
ESFJ	0.74	0.64	0.69	67
macro avg	0.56	0.52	0.53	1735
weight avg	0.68	0.67	0.67	1735

class has the same number of examples as the majority class. Random Oversampling has been shown to be effective in improving the performance of machine learning models on imbalanced datasets [10].

TABLE 3. Classification report multi-head attention model with random oversampler.

Class	Precision	Recall	F1-score	Support
INFJ	0.48	0.61	0.53	38
ENTP	0.53	0.61	0.56	135
INTP	0.39	0.48	0.43	46
INTJ	0.61	0.57	0.59	137
ENTJ	0.75	0.33	0.46	9
ENFJ	0.00	0.00	0.00	10
INFP	0.33	0.25	0.29	8
ENFP	0.20	0.28	0.23	18
ISFP	0.65	0.75	0.69	294
ISTP	0.73	0.68	0.70	366
ISFJ	0.74	0.49	0.59	218
ISTJ	0.62	0.72	0.67	261
ESTP	0.58	0.55	0.56	33
ESFP	0.47	0.50	0.49	54
ESTJ	0.37	0.37	0.37	41
ESFJ	0.63	0.51	0.56	67
macro avg	0.51	0.48	0.48	1735
weight avg	0.63	0.62	0.62	1735

Table 3 reports the performance of a Random Oversampler that achieves an overall F1-score of 0.62, where the type ENFJ has an F1-score of 0 and the type ISTP has the highest F1-score 0.70 across types. The type ENTJ has the highest precision of 0.75 and the type ISFP has the highest recall of 0.75. The weighted average scores of precision, recall, and f1-scores are 0.63, 0.62, and 0.62, respectively. In comparison to table 2 with Multi-head Attention base model the approach with Random Oversampling did not improve precision, recall and F1-score.

2) SMOTE

Synthetic Minority Oversampling Technique (SMOTE) is an oversampling method used in the field of machine learning to address the problem of imbalanced datasets [10]. SMOTE works by generating synthetic samples of the minority class using a K-nearest neighbor approach. This involves selecting a sample from the minority class and then finding its K nearest neighbors. The synthetic sample is then created by selecting a random point along the line connecting the selected sample and one of its K nearest neighbors. By generating synthetic samples in this way, SMOTE can effectively increase the number of samples in the minority class, balancing the dataset and improving the performance of machine learning models.

TABLE 4. Classification report multi-head attention model with SMOTE.

Class	Precision	Recall	F1-score	Support
INFJ	0.49	0.55	0.52	38
ENTP	0.59	0.62	0.61	135
INTP	0.61	0.43	0.51	46
INTJ	0.73	0.55	0.62	137
ENTJ	0.00	0.00	0.00	9
ENFJ	0.00	0.00	0.00	10
INFP	0.40	0.25	0.31	8
ENFP	0.50	0.28	0.36	18
ISFP	0.67	0.77	0.71	294
ISTP	0.75	0.75	0.75	366
ISFJ	0.78	0.59	0.67	218
ISTJ	0.62	0.79	0.70	261
ESTP	0.50	0.61	0.55	33
ESFP	0.51	0.57	0.54	54
ESTJ	0.49	0.44	0.46	41
ESFJ	0.66	0.60	0.62	67
macro avg	0.52	0.49	0.50	1735
weight avg	0.66	0.66	0.66	1735

Table 4 reports the classification results obtained by applying SMOTE. It shows that the type ENTP has slightly higher scores with a precision of 0.59, a recall of 0.62, and an F1-score of 0.61. The types ENTJ and ENFJ have the lowest scores of precision of 0, recall of 0, and an F1-score of 0. The macro averages for precision, recall, and F1-score were 0.52, 0.49, and 0.50 respectively, indicating average performance across all classes. However, the weighted averages demonstrate slightly better performance with values all at 0.66. SMOTE outperformed the Random OverSampler 3 in terms of precision, recall and F1-score.

3) SMOTEENN

Synthetic Minority Oversampling Technique with Edited Nearest Neighbors (SMOTEENN) is a data preprocessing method used to balance class distributions in a dataset by oversampling the minority class. After oversampling, SMOTEENN uses the edited nearest neighbors method to clean the oversampled dataset by removing any synthetic examples that are too close to the boundary between the two classes. This helps to reduce the risk of overfitting and improve the generalization performance of the model. SMOTEENN is effective in improving the performance of machine learning models on imbalanced datasets [5].

The result from Table 5 of SMOTEENN shows the types like ENTJ has a high precision of 0.75, INTJ has a high recall of 0.72, and ISTJ F1-score of 0.56. The types ENFJ,

TABLE 5. Classification report multi-head attention model with SMOTEENN.

Class	Precision	Recall	F1-score	Support
INFJ	0.21	0.63	0.31	38
ENTP	0.40	0.56	0.47	135
INTP	0.15	0.70	0.25	46
INTJ	0.42	0.72	0.53	137
ENTJ	0.75	0.33	0.46	9
ENFJ	0.00	0.00	0.00	10
INFP	0.22	0.50	0.31	8
ENFP	0.12	0.44	0.19	18
ISFP	0.66	0.41	0.51	294
ISTP	1.00	0.04	0.07	366
ISFJ	0.56	0.51	0.53	218
ISTJ	0.65	0.50	0.56	261
ESTP	0.27	0.55	0.36	33
ESFP	0.17	0.69	0.28	54
ESTJ	0.00	0.00	0.00	41
ESFJ	0.00	0.00	0.00	67
macro avg	0.35	0.41	0.30	1735
weight avg	0.58	0.39	0.37	1735

TABLE 6. Classification report multi-head attention model with class weights.

Class	Precision	Recall	F1-score	Support
INFJ	0.31	0.76	0.44	38
ENTP	0.57	0.60	0.59	135
INTP	0.49	0.43	0.46	46
INTJ	0.58	0.64	0.61	137
ENTJ	0.13	0.44	0.20	9
ENFJ	0.05	0.20	0.08	10
INFP	0.20	0.62	0.30	8
ENFP	0.20	0.61	0.30	18
ISFP	0.76	0.53	0.62	294
ISTP	0.79	0.63	0.70	366
ISFJ	0.67	0.54	0.60	218
ISTJ	0.72	0.54	0.62	261
ESTP	0.43	0.61	0.50	33
ESFP	0.38	0.65	0.48	54
ESTJ	0.26	0.59	0.36	41
ESFJ	0.66	0.55	0.60	67
macro avg	0.45	0.56	0.47	1735
weight avg	0.65	0.58	0.60	1735

ESTJ, and ESFJ have the lowest precision, recall, and F1-score of 0 across. The weighted average of the approach reported 0.58, 0.29, and 0.37 for precision, recall, and F1-score simultaneously. However, in comparison to SMOTE 4 the approach did not improve in terms of precision, recall, and F1-score.

B. IMPROVING RESULTS WITH ALGORITHMIC LEVEL IN IMBALANCED DATASETS

1) CLASS WEIGHTS

Class weights [27] are used in machine learning models to specify the importance of each class in a multi-class classification problem, which is provided during the model training. In mathematical terms, class weights can be used to modify the loss function of a machine learning algorithm. This can be done by multiplying the loss associated with each class by the weight assigned to that class. The modified loss function would be as follows:

$$L = \frac{1}{N}\sum_{i=1}^{N} w_i \cdot \text{loss}(y_i, \hat{y}_i)$$

where N is the number of examples, w_i is the weight assigned to the i-th example, $\text{loss}(y_i, \hat{y}_i)$ is the loss associated with the i-th example, y_i is the true label of the i-th example, and $\hat{y}_i$ is the predicted label of the i-th example.

We used Class Weight [27] in this experiment where the weights for the classes were computed by the compute_class_weight function to balance for the difference in number of instances of the two classes in the dataset. These weights were then put into a dictionary called class_final where each class label was assigned to its calculated weight. These weights were applied during the training which improves the model's capacity to predict a minority class while increasing the overall performance of the model.

Table 6 represents the result obtained by the multi-head attention model at the algorithmic level using the class weights approach. The classification report shows the types ENFJ, ESTJ, and ESFJ have improved compared to the approach SMOTEENN. The type ISTP achieves precision, recall, and F1-score of 0.79,0.63 and 0.70 respectively. The type ENFJ has the lowest precision of 0.05, recall of 0.20, and F1-score of 0.08. The weighted average precision of 0.65, recall of 0.58, and F1-score of 0.60 have been achieved with the class weights approach. The approach outperformed the SMOTEENN 5 across all terms.

2) REGULARIZATION

In machine learning and statistics, regularization is a technique used to prevent overfitting by adding a penalty term to the objective function [8]. This penalty term, which is typically a parameter that is multiplied by the magnitude of the coefficients, helps to reduce the complexity of the model by penalizing large coefficients [20]. As a result, regularization can help to improve the generalization of the model, making it better at making predictions on unseen data.

One common form of regularization is called `L1` regularization as known as Lasso regularization [8], which adds a penalty term that is proportional to the absolute value of the coefficients. This type of regularization can be used to perform feature selection by forcing some of the coefficients to be zero, effectively removing those features from the model. `L2` regularization, also known as Ridge regularization [8], on the other hand, adds a penalty term that is proportional to the square of the coefficients. This type of regularization can help to reduce overfitting by shrinking the coefficients, but it does not perform feature selection [20]. In our experiment, we utilized `L2` Regularization [8], $tf.keras.regularizers.l2(l = 0.0001)$ adding a penalty to the loss function based on the sum of the squared weight values. This helps prevent overfitting by discouraging large weights. The regularization strength is controlled by $l = 0.0001$, ensuring a subtle effect.

TABLE 7. Classification report multi-head attention model with L2 regularization.

Class	Precision	Recall	F1-score	Support
INFJ	0.62	0.47	0.54	38
ENTP	0.57	0.59	0.58	135
INTP	0.69	0.39	0.50	46
INTJ	0.62	0.68	0.65	137
ENTJ	0.50	0.11	0.18	9
ENFJ	0.00	0.00	0.00	10
INFP	0.67	0.50	0.57	8
ENFP	0.43	0.33	0.38	18
ISFP	0.67	0.73	0.70	294
ISTP	0.76	0.75	0.75	366
ISFJ	0.66	0.61	0.64	218
ISTJ	0.67	0.78	0.72	261
ESTP	0.72	0.55	0.62	33
ESFP	0.54	0.54	0.54	54
ESTJ	0.55	0.41	0.47	41
ESFJ	0.64	0.69	0.66	67
macro avg	0.58	0.51	0.53	1735
weight avg	0.66	0.67	0.66	1735

TABLE 8. Classification report BERT base model.

Class	Precision	Recall	F1-score	Support
INFJ	0.74	0.53	0.62	38
ENTP	0.57	0.71	0.63	135
INTP	0.57	0.50	0.53	46
INTJ	0.63	0.71	0.67	137
ENTJ	0.00	0.00	0.00	9
ENFJ	0.00	0.00	0.00	10
INFP	1.00	0.25	0.40	8
ENFP	0.29	0.11	0.16	18
ISFP	0.70	0.75	0.72	294
ISTP	0.75	0.75	0.75	366
ISFJ	0.74	0.70	0.72	218
ISTJ	0.75	0.71	0.73	261
ESTP	0.80	0.61	0.69	33
ESFP	0.53	0.69	0.60	54
ESTJ	0.57	0.49	0.53	41
ESFJ	0.67	0.69	0.68	67
macro avg	0.58	0.51	0.53	1735
weight avg	0.69	0.69	0.68	1735

Table 7 shows the L2 Regularization approach achieves the weighted average precision, recall, and F1-score of 0.66, 0.67, and 0.66 respectively. The type ISTP has the highest precision, recall and F1-score of 0.76, 0.75 and 0.75 respectively. The type ENFJ has the lowest precision, recall, and F1-score of 0 throughout. The L2 Regularization improved precision, recall, and F1-score compared to class weights 6.

C. BERT MODEL

The model architecture consists of the pre-trained BERT `'bert-base-uncased'` model which we fined tuned with our custom feed-forward classifier. The Bert pre-trained model had 768 hidden dimensions, which capture and interpret the linguistic nuances of the input. In the fine-tuned model the model is constructed with a linear layer. The model is fine-tuned with 768 hidden dimensions with an output of 50 dimensions. The output of the model is set to 16 with the activation Relu activation to match the final output size. During the training configuration, the batch size is set to 16 and the epoch is set to 30.

Table 8 represents the classification report of the base BERT model. The model has the highest weighted average of precision, recall, and F1-score of 0.67, 0.68, and 0.67 respectively along with Multi-head attention model with data level and algorithmic level approaches. However, the types ENTJ and ENFJ have scores of 0 throughout precision, recall, and F1-score.

Overall, The BERT model demonstrates significant improvement by achieving the highest precession, recall and F1-score compared to the base Mulit-head attention model. The BERT model improvements bring the notion of contextual techniques better than traditional resampling methods. In the imbalanced MBTI datasets, our fine-tuned BERT model improves precision, recall and classification accuracy.

IX. TEXTUAL DATA AUGMENTATION

A. DATA AUGMENTATION AT WORD LEVEL

Data augmentation is a technique used to artificially increase the size of a dataset by generating new data samples from the existing ones. This is often done in machine learning to improve the performance of a model by providing it with more diverse and representative data to learn from [14]. There are various ways to perform data augmentation, including changing the color or brightness of an image, adding noise to a signal, or generating new text by paraphrasing or translating existing text [40]. Data augmentation can also involve synthesizing new data samples using techniques such as Generative Adversarial Networks (GANs) [15] or Variational Autoencoders (VAEs) [28].

One of the main benefits of data augmentation is that it allows for the expansion of a dataset without the need to manually collect and label additional data, which can be time-consuming and costly [41]. This can be especially useful in situations where it is difficult to obtain a large enough dataset, or when the cost of collecting more data is prohibitive.

In the research, we have utilized Python library nlpaug [1]. The NLPaug library provides a set of tools and functions for augmenting text data in a variety of ways, including adding noise, inserting and deleting words or characters, and modifying the structure of sentences. `nlpaug.augmenter.word.SynonymAug` is a function from the NLPaug library that performs data augmentation by replacing words in the input text with their synonyms. The data augmentation added 4,855 samples to the original dataset during the experiment.

The classification report using data augmentation in Table 9 shows that INFJ, ENTJ, ENFJ, and ENFP have scores of 0 throughout all types in precision, recall, and F1-score. The highest precision of 0.62 at type ISTP, the highest recall of 0.75 at type ISFP, and the highest f1-score

TABLE 9. Classification report multi-head attention model with data augmentation.

Class	Precision	Recall	F1-score	Support
INFJ	0.00	0.00	0.00	39
ENTP	0.09	0.02	0.03	142
INTP	0.05	0.02	0.03	52
INTJ	0.14	0.05	0.07	150
ENTJ	0.00	0.00	0.00	13
ENFJ	0.00	0.00	0.00	8
INFP	0.00	0.00	0.00	4
ENFP	0.00	0.00	0.00	18
ISFP	0.42	0.75	0.54	266
ISTP	0.62	0.59	0.60	373
ISFJ	0.61	0.55	0.58	212
ISTJ	0.51	0.54	0.52	264
ESTP	0.04	0.09	0.05	32
ESFP	0.02	0.02	0.02	58
ESTJ	0.03	0.05	0.04	38
ESFJ	0.12	0.09	0.10	66
macro avg	0.17	0.17	0.16	1735
weight avg	0.38	0.40	0.38	1735

of 0.60 at type ISTP are reported in Table 9. Even though, the data augmentation increased the sample data the model improved in training accuracy. However, the testing accuracy did not improve. The approach data augmentation (word level) achieves a precision of 0.38, recall of 0.40, and f1-score of 0.38, which is the lowest weighted average across all other approaches including BERT and Multi-head Attention with data level and algorithmic level.

B. LLM-BASED AUGMENTATION

To perform data augmentation using LLMs in the experiment, we utilized the `GPT 3.5 Turbo` with the prompt to generate synthetic data for the minority classes. In the original dataset the type "ENTJ, ENFJ, ISFP, ISTP, ISFJ, ISTJ, ESTP, ESFP, ESTJ, ESFJ" contains less than 3.8 % of the dataset. In the LLM data augmentation analysis, we have chosen all the types that have less than 5.0 % of the dataset and increased to 5.0 % of the dataset. Figure 4 demonstrates the change in each class after using LLM data augmentation. We have utilized the prompt engineering technique along with the few shots example to generate sample data using the LLM model (i.e. `GPT 3.5 Turbo`). The *temperature* = 0.5, *top_p* = 0.7 were used as the configuration during the experiment for the model. The temperature is a parameter in the model that controls the randomness of the model's output. The top_p value is the concentrate probability threshold that controls the model's output.

1) THE PROMPT

The prompt instruction contains the input {**data**} as the posts and type for the original text. The instruction has the task to generate the additional specific number of samples in the {**type**} given which will be added to the original dataset.

"Instruction Prompt"*: You are an expert in the personality trait analysis in text. For your reference use the provided posts and their label (i.e. personality traits) examples as a reference to understand the context and style of each personality type. Create five new and distinct text posts for a given {**type**} category using the style and context of the existing samples in {**data**}. Ensure that each new text posts correspond with the intended personality types's.*

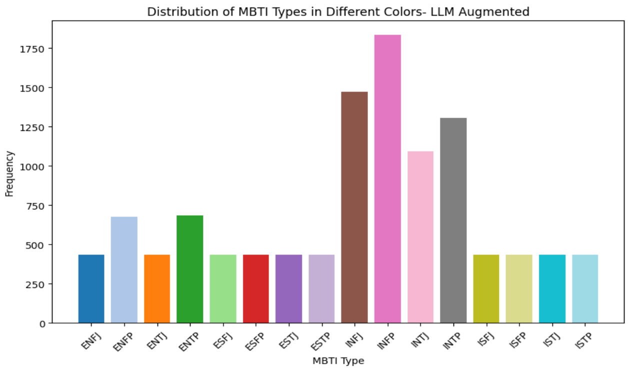


FIGURE 4. Personality types distribution in MBTI dataset with LLM augmated.

The prompt has been chosen with various iterations of experiment and observation. The prompt engineering criteria have shown the categorical approach which includes 1) clarity and specificity, 2) objectives and intents, 3) contextual information, 4) format and style, and 5) optimality along with sensitive levels of high.

Tables 10, 11 display classification reports for two different models applied to an LLM-based data augmentation dataset: Multi-Head Attention Model with LLM-Data Augmentation 10, and the BERT model with LLM-Data Augmentation 11. The table 10 presents the classification report for the Multi-head Attention model with LLM-based data augmentation which has improved results compared to 8. The approach achieves a weighted average in terms of precision of 0.70, recall of 0.70, and F1-score of 0.69. The result from the data augmentation at word level 9 has been suppressed by the LLM-based data augmentation.

TABLE 10. Classification report multi-head attention model with LLM-based data augmentation.

Class	Precision	Recall	F1-score	Support
INFJ	0.78	0.64	0.70	87
ENTP	0.57	0.64	0.60	135
INTP	0.54	0.69	0.61	87
INTJ	0.66	0.46	0.54	137
ENTJ	0.96	1.00	0.98	87
ENFJ	0.91	1.00	0.95	87
INFP	0.80	1.00	0.89	87
ENFP	0.81	0.92	0.86	86
ISFP	0.70	0.66	0.68	294
ISTP	0.74	0.73	0.73	366
ISFJ	0.60	0.67	0.63	218
ISTJ	0.66	0.68	0.67	261
ESTP	0.74	0.78	0.76	87
ESFP	0.68	0.42	0.52	86
ESTJ	0.81	0.56	0.66	86
ESFJ	0.51	0.56	0.53	87
macro avg	0.72	0.71	0.71	2278
weight avg	0.70	0.70	0.69	2278

In table 11 the classification report demonstrates the BERT model with LLM-Data Augmented Dataset which achieved the highest weighted average of 0.76 throughout precision, recall, and F1-score. This model shows particularly strong performance in classes like ENTJ, ENFJ, INFP, and ENFP, where it achieves nearly perfect precision and recall. The model effectively leveraging the LLM-based data augmentation to achieve better overall classification results shows higher performance metrics across most classes compared to the multi-head attention Model.

TABLE 11. Classification report BERT LLM-based data augmentation.

Class	Precision	Recall	F1-score	Support
INFJ	0.80	0.92	0.86	87
ENTP	0.64	0.61	0.63	135
INTP	0.82	0.78	0.80	87
INTJ	0.64	0.61	0.63	137
ENTJ	0.96	1.00	0.98	87
ENFJ	0.96	1.00	0.98	87
INFP	0.98	1.00	0.99	87
ENFP	0.91	1.00	0.96	86
ISFP	0.70	0.68	0.69	294
ISTP	0.75	0.72	0.74	366
ISFJ	0.70	0.64	0.67	218
ISTJ	0.67	0.72	0.69	261
ESTP	0.84	0.93	0.89	87
ESFP	0.77	0.71	0.74	86
ESTJ	0.81	0.90	0.85	86
ESFJ	0.77	0.74	0.75	87
SMOTEine macro avg	0.80	0.81	0.80	2278
weighted avg	0.76	0.76	0.76	2278

X. DISCUSSION

Table 12 presents a detailed comparison of model test accuracies from the experiment. For instance, the INFJ personality type shows a Multi-head attention base model accuracy of 42.11%, the Multi-head attention with Data augmentation accuracy of 73.68% of which significantly improves to 97.75% when augmented with LLM data in BERT model. Similarly, the ENTP and INTP types have an accuracy of 63.70% and 67.39% respectively with Multi-head attention SMOTEENN. The ENTP and INTP types achieve 90.16% with base BERT and 94.02% with LLM data augmentation respectively.

Notably, the ENFJ type demonstrates an improvement, 0.00% in the Multi-head attention base model and achieving 100.00% accuracy with both Multi-head attention and BERT with LLM Data Augmentation, illustrating the impact of technique in the model performance. The ISTP type shows improvement, from 74.59% with L2 regularization to 93.40% with the BERT base model.

For the ESTJ type, Multi-head attention with Data augmentation accuracy of 51.22% is reported, which increases to 98.40% with LLM Data Augmentation in the BERT model. Lastly, the ESFJ type, with L2 regularization and Data augmentation in Multi-head attention accuracy of 68.66%, achieves an accuracy of 92.63% with LLM Data Augmentation in BERT model.

These accuracies of type in model test highlight the Multi-head attention with Data augmentation the has highest accuracies in 7 types. The others technique like SMOTEENN has the highest accuracies of 4 types and L2 regularization which a the algorithmic level (i.e. model level) has the highest accuracies of 5 types. The methods like SMOTE, Random over sampler and Class weights have significantly optimized the predictive capabilities of the Multi-head attention model. However, the LLM Data Augmentation demonstrates the drastic improvement in MBTI types with the Multi-head attention and BERT model. The analysis shows the leverage of LLM for advanced data augmentation to improve the accuracy and reliability of the model to deal with imbalanced data in personality prediction tasks.

Table 13 demonstrates that Multi-head attention with the random oversampling technique has the highest training accuracy of 90 %. The test Bert model with the LLM-based data augmentation has the highest accuracy of 76.21%. The Bert with the original dataset also shows the highest testing accuracy of 69.05%, Even though the Multi-Head attention with the LLM-based data augmentation has the testing accuracy of 69.75%.

In addition, Table 13 has the entropy scores of each model, where the higher entropy score refers to the higher level of uncertainty in model prediction. The Multi-Head attention with Data augmentation has the lowest entropy score of 2.816 whereas the model has the lowest testing accuracy of 35.91%. The BERT Model with LLM data augmentation has the highest entropy score of 3.766 with the highest training accuracy amongst the models of 76.21%.

Table 12 provides a comparative analysis for methods such as SMOTE, SMOTEENN, Random Over Sampler, class weights, L2 regularization, Data augmentation technique, and LLM Data augmentation technique. An extension of this discussion could be exploring why such techniques exhibited being more effective than the other techniques. For example, the performance of LLM Data Augmentation used the highest accuracy almost for every class especially when it has been combined with BERT. This may be because LLMs are able to produce more contextually relevant and balanced training samples, enhancing the possibility of generalized performance of the model. Conversely, the traditional approaches such as SMOTEENN, while effective in some instances such as the INTP and ENTP classes, may not fully capture the contextual features, leading to lower accuracy in other classes.

The lower accuracy of the techniques, such as Random Over Sampler, and L2 Regularization, could be affected by their inefficiency in managing the high dimensionality and distinct variations between personality classes. For instance, these methods may lead to issues of overfitting or underfitting especially when working on small sample data sets or when dealing with highly imbalanced classes like ENFJ and ENTJ or any class with very few samples this will lead to a serious drop in accuracy.

Altogether, this research shows the advantage of LLM-based data augmentation, especially when used in combination with BERT highlighting the value of advanced

TABLE 12. **Model test accuracy of each class.**

Class	Multi-Head Attention with								Bert with	
	Base	SMOTE	SMOTEENN	Random Over Sampler	Class Weights	L2 Regularization	Data Augmentation	LLM Data Augmentation	Base	LLM Data Augmentation
INFJ	42.11	47.37	63.16	28.95	0.00	47.37	73.68	64.37	94.54	97.75
ENTP	57.78	60.74	63.70	62.22	4.23	58.52	61.48	63.70	90.16	87.50
INTP	43.48	45.65	67.39	41.30	3.85	39.13	45.65	68.97	84.40	94.02
INTJ	66.42	62.77	66.42	56.93	8.67	67.88	62.77	45.99	93.37	89.18
ENTJ	11.11	0.00	33.33	11.11	0.00	11.11	11.11	100.00	81.77	100.00
ENFJ	0.00	0.00	0.00	0.00	12.50	0.00	20.00	100.00	53.27	99.98
INFP	50.00	0.00	25.00	0.00	0.00	50.00	50.00	100.00	95.02	100.00
ENFP	22.22	0.00	16.67	5.56	0.00	33.33	55.56	91.86	94.71	99.94
ISFP	76.53	72.79	39.80	69.05	86.47	72.79	58.50	65.99	90.29	89.92
ISTP	74.04	70.22	6.01	68.58	42.09	74.59	62.30	72.68	93.40	90.45
ISFJ	65.60	56.88	60.55	55.96	46.23	61.47	53.21	66.97	90.96	87.63
ISTJ	75.86	70.50	60.15	63.98	40.53	78.16	68.20	67.82	92.04	90.89
ESTP	57.58	39.39	66.67	36.36	0.00	54.55	63.64	78.16	89.42	97.95
ESFP	51.85	42.59	48.15	38.89	5.17	53.70	66.67	41.86	90.25	88.06
ESTJ	36.59	41.46	0.00	24.39	0.00	41.46	51.22	55.81	78.43	98.40
ESFJ	59.70	59.70	0.00	62.69	9.09	68.66	68.66	56.32	91.64	92.63

TABLE 13. **Model training and testing accuracy.**

Model	Training Accuracy	Testing Accuracy	Entropy Score
Multi-Head Attention	77.10%	66.46%	3.200
Multi-Head Attention with SMOTE	80.93%	62.19%	3.184
Multi-Head Attention with SMOTEENN	82.92%	41.27%	3.370
Multi-Head Attention with Random Over Sampler	90.84%	58.90%	3.174
Multi-Head Attention with Class Weights	66.15%	60.69%	3.648
Multi-Head Attention with L2 Regularization	76.60%	66.51%	3.217
Multi-Head Attention with Data Augmentation	87.69%	35.91%	2.816
BERT	73.16%	69.05%	3.380
Multi-Head Attention with LLM Data Augmentation	80.90%	69.75%	3.754
BERT with LLM Data Augmentation	82.80%	76.21%	3.766

context-aware techniques. The context-aware approach is particularly effective for solving such multi-class imbalance problems as personality prediction. It is possible to continue refining these techniques to achieve better performance in all the classes in future studies.

XI. CONCLUSION AND FUTURE WORK

In this research, we conduct experiments to classify highly imbalanced Myers-Briggs personality types data using transformer-based deep learning models and LLM-based data augmentation techniques. Our approach in combining Multi-head Attention, BERT, and GPT-3.5-Turbo has shown enhanced model performance, especially in obtaining higher precision, recall, and F1 scores in the MBTI dataset. By creating synthetic samples using techniques such as SMOTE, SMOTEENN, we reduce the problem of having few samples of minority classes, which in turn makes the model more generalized. The proposed method can be used in various domains, including vision and speech domains which make this contribution relevant to the advancement of machine learning to deal with class imbalance.

From the experiments, the original MBIT dataset with various data levels and algorithmic techniques including random over-sampling, SMOTE, SMOTEENN, Data augmentation, Class weights and L2 Regularization with Multi-head attention and BERT models have the highest F1-scores 0.69 for most personality types. The oversampling methods like random over sampler improved the training accuracy but did not perform during the model testing. The algorithmic approaches including class weights and regularization had only a limited impact on training and testing accuracy. The data augmentation in word level improved training accuracy. However, the model test has the lowest F1 score of 0.38. The BERT fine-tuned model performance well on the original imbalanced data with the highest testing accuracy of 69.05%.

In addition, the LLM data augmentation using GPT-3.5 Turbo further improved BERT's performance, achieving test accuracy of 76.21% and the highest F1-score of 0.76. The multi-head attention model with LLM data augmentation using GPT-3.5 Turbo has a testing accuracy of 69.75% making it second highest accuracy across other approaches. The results demonstrate that the LLM data argumentation using GPT-3.5 Turbo performs better in both the Multi-head attention and BERT model. The experiments demonstrate that the potential of leveraging large language models like GPT-3.5 for data augmentation can utilized to handle class imbalance and increase model performance in real-world scenarios.

The use of data from the MBTI is a limitation and a threat to validity in terms of the range and variability of the data, which limits the research into how well the results can be applied and generalized to other data sets. The issue of overfitting or underfitting is another major issue that could dramatically affect the performance of models while dealing with small samples. The complication that comes with the advanced procedures for LLM-based data augmentation such as the computational cost and the need for fine-tuning for specified domains make it difficult to implement these concepts in a large-scale environment.

Future research should attempt to replicate and validate the studied methods in various domains such as engineering, healthcare and finance. It turns out that these areas are as prone to class imbalance as our study here. However, LLM-based augmentation raises the computational difficulty in subsequent processes which is the area that can be improved based on the presented work in the future. They present useful suggestions for further improvement of the model and areas of its further applicability. To our best knowledge, there is no other similar dataset with comparable features that can be studied for the purpose of this research paper. Thus, there is a need to create larger datasets in the future and conduct larger scale replication and analysis. Additionally, It will be also interesting to investigate combining Active Learning techniques with LLMs, playing the role of an oracle, that can be considered as effective in order to reduce the training time and to focus on the most informative samples. Finally, applying LLM prompts through the Chain-of-Thought technique may enhance the reasoning process inside models, which in turn might generate better samples during data augmentation.

REFERENCES

[1] *Nlpaug Documentation*. Accessed: Dec. 28, 2022. [Online]. Available: https://nlpaug.readthedocs.io/

[2] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.

[3] N. H. Z. Abidin, M. Akmal, N. Mohd, D. Nincarean, N. Yusoff, and H. Karimah, "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.

[4] 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.

[5] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 20–29, Jun. 2004.

[6] R. Bayne, *The Myers-Briggs Type Indicator: A Critical Review and Practical Guide*. Cheltenham, U.K.: Nelson Thornes Ltd, 1997.

[7] S. Bird, E. Klein, and E. Loper, *Natural Language Processing With Python: Analyzing Text With the Natural Language Toolkit*. Sebastopol, CA, USA: O'Reilly Media, 2009.

[8] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4. New York, NY, USA: Springer, 2006.

[9] A. Bosch, X. Muñoz, and R. Martí, "Which is the best way to organize/classify images by content?" *Image Vis. Comput.*, vol. 25, no. 6, pp. 778–791, Jun. 2007.

[10] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[11] X. Chen, W. Zhang, S. Pan, and J. Chen, "Solving data imbalance in text classification with constructing contrastive samples," *IEEE Access*, vol. 11, pp. 90554–90562, 2023.

[12] L. W. Choi-Kain and J. G. Gunderson, "Mentalization: Ontogeny, assessment, and application in the treatment of borderline personality disorder," *Amer. J. Psychiatry*, vol. 165, no. 9, pp. 1127–1135, Sep. 2008.

[13] F. Chollet. (2015). *Keras*. [Online]. Available: https://keras.io

[14] F. Chollet, *Deep Learning With Python*. New York, NY, USA: Simon and Schuster, 2021.

[15] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.

[16] B. Cui and C. Qi, "Survey analysis of machine learning methods for natural language processing for MBTI personality type prediction," Rep., 2017.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[18] C. Gong, N. Zhou, S. Xia, and S. Huang, "Quantum particle swarm optimization algorithm based on diversity migration strategy," *Future Gener. Comput. Syst.*, vol. 157, pp. 445–458, Aug. 2024.

[19] L. Gong, W. Ding, Z. Li, Y. Wang, and N. Zhou, "Quantum K-nearest neighbor classification algorithm via a divide-and-conquer strategy," *Adv. Quantum Technol.*, vol. 7, no. 6, Jun. 2024, Art. no. 2300221.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[21] M. S. Halawa, M. E. Shehab, and E. M. R. Hamed, "Predicting student personality based on a data-driven model from student behavior on LMS and social networks," in *Proc. 5th Int. Conf. Digit. Inf. Process. Commun. (ICDIPC)*, Oct. 2015, pp. 294–299.

[22] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "Industrial-strength natural language processing in python," spaCy, Explosion, Berlin, Germany, Rep., 2020, doi: 10.5281/zenodo.1212303.

[23] S. Hosany, Y. Ekinci, and M. Uysal, "Destination image and destination personality: An application of branding theories to tourism places," *J. Bus. Res.*, vol. 59, no. 5, pp. 638–642, May 2006.

[24] D. Hovy and S. Prabhumoye, "Five sources of bias in natural language processing," *Lang. Linguistics Compass*, vol. 15, no. 8, Aug. 2021, Art. no. e12432.

[25] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007.

[26] Plotly Technologies. (2015). *Collaborative Data Science*. [Online]. Available: https://plot.ly

[27] G. King and L. Zeng, "Logistic regression in rare events data," *Political Anal.*, vol. 9, no. 2, pp. 137–163, 2001.

[28] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[29] M. C. Komisin and C. Guinn, "Identifying personality types using document classification methods," Ph.D. thesis, Univ. North Carolina Wilmington, Wilmington, NC, USA, 2011.

[30] S. Lee and D.-Y. Kim, "Brand personality of airbnb: Application of user involvement and gender differences," *J. Travel Tourism Marketing*, vol. 35, no. 1, pp. 32–45, Jan. 2018.

[31] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, no. 17, pp. 1–5, 2017.

[32] L. Loukas, I. Stogiannidis, O. Diamantopoulos, P. Malakasiotis, and S. Vassos, "Making LLMs worth every penny: Resource-limited text classification in banking," in *Proc. 4th ACM Int. Conf. AI Finance*, vol. 35, Nov. 2023, pp. 392–400.

[33] D. P. McAdams and J. L. Pals, "A new big five: Fundamental principles for an integrative science of personality.," *Amer. Psychologist*, vol. 61, no. 3, pp. 204–217, 2006.

[34] Z. Mushtaq, S. Ashraf, and N. Sabahat, "Predicting MBTI personality type with K-means clustering and gradient boosting," in *Proc. IEEE 23rd Int. Multitopic Conf. (INMIC)*, Nov. 2020, pp. 1–5.

[35] I. B. Myers, *The Myers-Briggs Type Indicator: Manual (1962)*. Palo Alto, CA, USA: Consulting Psychologists Press, 1962.

[36] OpenAI. (2023). *OpenAI API*. [Online]. Available: https://platform.openai.com/docs/models/gpt-3-5

[37] C. Padurariu and M. E. Breaban, "Dealing with data imbalance in text classification," *Proc. Comput. Sci.*, vol. 159, pp. 736–745, Jan. 2019.

[38] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI Blog, San Francisco, CA, USA, Rep., 2018.

[39] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[40] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[41] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Proc. ICDAR*, vol. 3, Edinburgh, U.K., Aug. 2003, pp. 958–963.

[42] As. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.* 30, 2017, pp. 5998–6008.

[43] M. Waskom, "Seaborn: Statistical data visualization," *J. Open Source Softw.*, vol. 6, no. 60, p. 3021, Apr. 2021.

[44] T. A. Widiger and T. J. Trull, "Personality and psychopathology: An application of the five-factor model," *J. Personality*, vol. 60, no. 2, pp. 363–393, 1992.

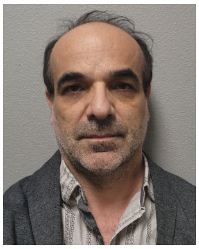
[45] P. William, A. Badholia, B. Patel, and M. Nigam, "Hybrid machine learning technique for personality classification from online text using HEXACO model," in *Proc. Int. Conf. Sustain. Comput. Data Commun. Syst. (ICSCDS)*, Apr. 2022, pp. 253–259.

[46] T. Wolf et al., "HuggingFace's transformers: State-of-the-art natural language processing," 2019, *arXiv:1910.03771*.

[47] W. Youyou, M. Kosinski, and D. Stillwell, "Computer-based personality judgments are more accurate than those made by humans," *Proc. Nat. Acad. Sci. USA*, vol. 112, no. 4, pp. 1036–1040, Jan. 2015.

SAROJ GOPALI received the B.S. degree in computer science, the master's degree in software engineering, and the Ph.D. degree in computer from Texas Tech University, Lubbock, in 2020, 2021, and 2024, respectively. He has research experience in time series, deep learning models, natural language processing, and large language models in cybersecurity. His research interests include time series analysis in cybersecurity problems using deep learning, machine learning, and large language model techniques.

FARANAK ABRI received the master's and Ph.D. degrees in computer science from Texas Tech University, in 2020 and 2022, respectively. She is currently an Assistant Professor in computer science with San Jose State University. She has research experience in a wide range of topics in this area, including malware analysis, cloud security, automated deception detection, social engineering, security comprehension, and usable security. Her research interests include modeling cybersecurity problems using artificial intelligence (AI) and machine learning (ML) techniques.



AKBAR SIAMI NAMIN received the Ph.D. degree in computer science from Western University, London, ON, Canada, in August 2008. He is currently a Professor in computer science with Texas Tech University. He has co-authored over 100 research articles published in premier journals and venues. His research on cyber security research and education is funded by the National Science Foundation and Office of Navy Research (ONR). His research interests include machine learning, natural language processing (NLP), cyber security, and time series analysis.

KEITH S. JONES is currently a Human Factors Psychologist and a Professor with Texas Tech University who specializes in human–computer interaction. To date, he has been awarded over $2.9M in research funding from the National Science Foundation, the Office of Naval Research, the Air Force Office of Scientific Research, and Microsoft; and has published numerous journal articles in peer-reviewed outlets and conference proceeding papers at international venues. His current research interests include human–robot interaction and human factors issues related to cybersecurity.