Textual Personality Prediction in the Context of NLP and ML

Aurchi Roy, Mahmuda Junainah, Md. Ehtesham-ur-rahman aurid, Readhwana Reaz Adrin, Rifah Tasnia
Abid Hossain, Md Humaion Kabir Mehedi, Ehsanur Rahman Rhythm and Annajiat Alim Rasel
Department of Computer Science and Engineering (CSE)
School of Data and Sciences (SDS)
Brac University
66 Mohakhali, Dhaka - 1212, Bangladesh
{aurchi.roy, mahmuda.junainah, ehtesham.ur.rahman, readhwana.reaz.adrin, rifah.tasnia2
abid.hossain, humaion.kabir.mehedi, ehsanur.rahman.rhythm}@g.bracu.ac.bd, annajiat@gmail.com

Abstract—**The combination of Natural Language Processing (NLP) and machine learning has affected many aspects of our lives in our ever-changing digital landscape. NLP often subtly affects our relationships in everything from voice commands to complex online interactions. Our study investigates personality prediction based on textual analysis of interest in this context. Our study uses advanced datasets and methods, such as multinomial naive Bayes, convolutional Neural Networks, Random Forest, and Support Vector Machines (SVM) Our modeling process starts with painstaking data preprocessing, where transcription reduces noise by removing stopwords The validation loss and accuracy plots shown clearly show the evolution of the model and provide information about its learning process. Furthermore, a thorough analysis using the illusion matrix reveals the robustness of model performance. The predictive ability of the models is demonstrated through measures of precision, recall, and F1-score, and reflects the process of extracting personality traits from text. Our research essentially bridges the gap between linguistics and machine learning, encouraging readers to explore the fascinating interplay between language and personality prediction. This discovery provides a way to better understand the patterns in our language as the digital age advances.**

Index Terms—**Natural Language Processing, Text Analysis, Personality Prediction, Machine Learning Models, Convolutional Neural Network, Naive Bayes, Random Forest, Support Vector Machine**

I. Introduction

Nowadays, Natural Language Processing has become an effective tool not just in theoretical science but also in practical life. From voice typing to text auto-correction, along with Google Translate, we are using NLP without even realizing it. Ai chatbots can not function or communicate with the user without pre-programmed NLP. Moreover, as mankind is moving fast towards globalization and digitalization, we have to use smart devices a lot. For example, in the past, the work which were done offline with pen and paper is now being done by computer with text. Moreover, almost every adult is now somehow connected to any kind of social media. Social media is not just being used for personal reasons. It is also being used for professional reasons such as marketing, building connections, virtual meetings, etc. Therefore, as a human nowadays spends a noticeable amount of time texting, it is a possibility to predict a person's sentiment or personality by analyzing texts, using nlp and machine learning models. In order to make our research a success, we used a large dataset from Kegel, and various machine learning models were used in order to increase accuracy. Lastly, we have a vision to extend our research and add more versatile features to our work.

II. Literature Review

In the research by Jafari and Far [1], multiple methods were processed to predict Towards behavior and Away behavior by collecting the data using questionnaires and data provided by the experts. Later on, they applied five methods. These methods are IBM emotion analysis, ACS sentiment analysis, vector-based method, logistic regression, and lastly bag-of-words method. Five models were used during this analysis which are Logistic Regression, Random Forest, Multinomial Naive Bayes, Complement Naive Bayes, and Gaussian Naive Bayes. Among all of them, IBM emotional analysis predicted the unseen data better than other methods although it has the lowest AUC score and accuracy. The average sentiment score for toward data is 0.78 and for away, it is -0.36. However, the study is limited due to its lack of data and imbalanced data, as well as time-related constraints that led to incomplete labeling of words in the datasets.

In another paper [2], the authors Anas, Gunavathi, Kirubasri used machine learning models (ML) to identify customer personality types from tweets and group them with similar personalities for carpooling. The objective of this study was to group individuals with similar personalities in the same vehicle, bridge the gap between people who are uncomfortable in traveling alongside those with different personalities, and make carpooling easier. In this paper, the authors used the Myers-Briggs Personality Type dataset from Kaggle and preprocessed the dataset using different NLP techniques before feeding the dataset data into the machine learning models. The preprocessing techniques used by the authors include removing unnecessary contents such as removing URL, '#', '@', removing all non-characters, etc, encoding each personality category using Sklearn, LabelEncoder and then applying the different NLP techniques such as tokenization, stopping words removal, stemming and lemmatization. After the data was preprocessed, the authors passed the preprocessed data into the ML models to predict the personality of the individuals. The ML algorithms that were used in this paper are SVM, XG Boost, Decision Tree Classifier, and Stochastic gradient. The output of the machine learning algorithms was passed into a matching algorithm and vectorized using a count vectorizer, then the vectorized text was again passed into the pre-trained machine learning models to predict the personality type of the individuals. Finally, the output was stored in a database. In this paper. The XG Boost ML model outperformed the other ML models by achieving the highest accuracy of 68%. However, the limitation of this study is that the proposed method may not be optimal for small customer numbers with diverse personality types.

Researchers have also used different NLP and deep learning techniques to analyze sentiments. In this paper the [3] author focuses on analyzing text using NLP and predicting public sentiments regarding news, especially financial news related to the stock market. This helps businessmen a lot as public opinions have a huge impact on trading in both the stock market and the price hike or downfall of a certain product. Thus trading firms try to use public reactions regarding news in order to make a profit (Ruiz-Martínez et al. 2012). The author used multiple datasets, especially from the DJIA Database [3], which is from Kaggle and contains 25 daily articles with financial news from 2008 to 2016 that the author pulled from the most popular articles on Reddit WorldNews. After pre-processing the datasets with NLP, the author used deep learning to analyze public sentiment. He used a deep learning function which is similar to the human brain's process to find patterns. The function is :- $(y \log(p) + (1 - y) \log(1 - p))$. Lastly, it is one of very few papers closely related to our topic and helped us in improving our work.

In the research conducted by Kanchana, Zoraida and Evelin, [4], the Big Five personality model is used by the authors' framework to forecast an individual's personality using social media data. Openness, conscientiousness, extraversion, agreeableness, and neuroticism are 5 major personality traits included in this paradigm. The authors assess the sentiment contained in social media comments and forecast the Big Five personality traits using NLP, NLTK, and LSTM neural networks. The study finds that personality traits can be accurately predicted from users' tweets, achieving an accuracy of 92%. The three key components of the presented methodology are- collection of data, pre-processing, and model construction. The authors utilize a dataset of approximately 14,183 tweets labeled with 11 parameters related to emotions, such as anger, joy, fear, and trust. Unfortunately, the paper does not provide explicit information about the source or origin of the dataset. After pre-processing the dataset with NLP and NLTK, the LSTM model is trained to forecast the characteristics of the Big Five. The authors highlight that the association between textual data and traits of personality alongside the inclusion of contextual information from images, can contribute to accurate personality prediction. The study shows that certain emotions and sentiments are closely associated with specific personality traits. The LSTM model's accuracy surpasses that of other techniques, indicating its capability to accurately predict user personality traits. By utilizing NLP, sentiment analysis, and LSTM neural networks, the authors achieve a high accuracy rate in predicting the characteristics of the Big Five.The authors suggest that this approach could be extended to incorporate more complex interactions and contextual factors for even more accurate predictions.

The paper [5] is focused on the importance of textual prediction of personality, highlighting different methods of accurate prediction Once the language used in interviews is established in response to a deeper understanding of the main emphasis, this research goes to the investigation of different patterns and processes that predict personality traits. The article also discusses the dynamic development of personality prediction and highlights its many applications in psychology and computer science. One of the most important studies in this area uses the Big Five framework to build a personality prediction engine using knowledge derived from Facebook interaction data. The study uses techniques such as word processing and N-grams to efficiently examine predictive accuracy across multiple data sets. The study considers the potential implications of these revelations, also moving into areas such as the criminal justice system, and concludes with recommendations for improving current predictive tools The current state of personality prediction based on online data is also discussed in this literature review, emphasizing the need for creative regression algorithms and rich data sets to increase the accuracy of predictions What notably, the study identifies a wide range of potential applications of the criminal internal justice system, from specific integration to computerization The entire article deftly navigates analyzes, emphasizing the importance of predicting identity from data, the process of development, and several promising practical approaches for this young topic.

III. Methodology

A. Collected data

For this paper, we have selected the "MBTI Personality Types 500 Dataset" [6]. This specific dataset has been collected from the official kaggle website. The collected Dataset contains a wealth of insights into the human psyche, centered around the Myers-Briggs Type Indicator (MBTI). This dataset shows how people see the world and make decisions based on their personality types. It consists of around 106,000 preprocessed records, each consisting of posts along with the personality type of the author. These posts have been standardized to be of equal size, containing 500 words per sample. The personality types in the dataset encompass a diverse array of 16 unique values, representing the various cognitive functions and preferences individuals possess. The dataset also presents a rich diversity of perspectives, with the following personality type distribution: (1) INTP: 24% (2) INTJ: 21% (3) Other (combining various personality types): 55%. In order to create the dataset, Dylan Storey and Mitchell Jolly collected about 1.7 million posts from Reddit and around 9,000 posts from a forum called Personality Cafe. It basically has 106,067 records with writings and categories of personality in CSV format.

Data Analysis: In this part of the paper, we present a comprehensive analysis of the dataset used in this study, including its characteristics, preprocessing steps, and exploratory data visualizations. The dataset consists of posts collected from individuals' online interactions, with associated personality types as labels. Our analysis aim is to provide insights into the distribution of personality types within the dataset. We basically performed Exploratory Data Analysis (EDA) on the dataset and found out that all the instances are labeled properly with a MBTI personality type. The distribution of personality types within the dataset is shown in Figure 1, illustrating the frequency of each personality type. The distribution reveals varying levels of representation for different personality types, with some types being more prevalent than others.

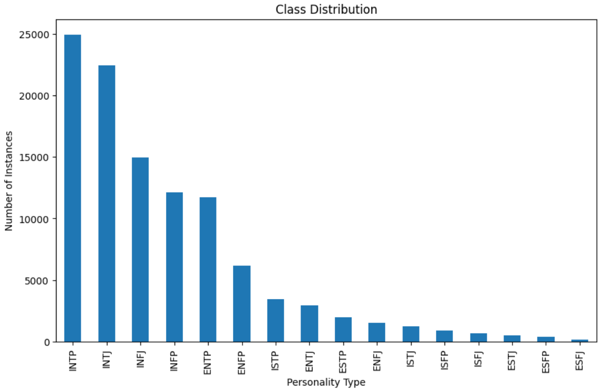

Fig. 1. Class distribution

To better understand the class imbalance, we computed some key imbalance metrics, including the Largest Class Ratio and the Average Class Size. These metrics provide insights into the dataset's relative representation of different personality types. We get the output of 'Largest Class Ratio': 137.91 and 'Average Class Size': 6629.19. Our dataset is a multiclass dataset with 16 different classes and each class contains a certain number of records:

In the dataset, no missing or duplicated values were found inside the classes as well. To ensure data quality and uniformity, the following preprocessing steps were applied:

1) Deletion of stopwords, URLs, and punctuation,
2) Word lemmatization and
3) Reconstruction of samples to ensure consistency in size (500 words)

Class	Record
INTP	24961
INTJ	22427
INFJ	14963
INFP	12134
ENTP	11725
ENFP	6167
ISTP	3424
ENTJ	2955
ESTP	1986
ENFJ	1534
ISTJ	1243
ISFP	875
ISFJ	650
ESTJ	482
ESFP	360
ESFJ	181

TABLE I
DISTRIBUTION OF INSTANCES IN EACH CLASS OF THE DATASET

B. Data Preprocessing

We have preprocessed our dataset by removing stopwords and unnecessary characters such as URL, newline characters and non-alphabetic characters from the dataset. The dataset was randomly shuffled and divided into training and testing set in the ratio of 90% and 10% respectively. Tokenization and label encoding were then applied to the training and testing set. Tokenization involved breaking the paragraphs into smaller units and label encoding involved converting categorical data into numerical data. These modifications allowed the data to be compatible with machine learning models which required numerical input. We have also balanced the classes of our dataset through random oversampling. Random oversampling is a technique used to solve the class imbalance problem in machine learning [7].

C. Proposed Model Architecture

In this paper, we will use different machine-learning techniques to predict personality traits from our dataset. The machine learning techniques that we will use are the convolutional neural network model (CNN) and some supervised machine learning algorithms such as multinomial Naive Bayes classifier, random forest, and SVM. In our proposed methodology we will start by utilizing the preprocessed dataset as an input for our machine learning models. This preprocessed dataset has undergone steps such as the removal of unnecessary characters and stopwords. We will then train this model using the training set and iterate the data for 10 epochs to improve the performance of the model. Once the training phase is complete, we will save the model and assess the performance of the model through training and validation loss and accuracy graphs. This visualization helps us to understand how well the model is learning during training and whether it is susceptible to overfitting or underfitting. Finally, we will generate a confusion matrix and evaluation metrics. This evaluation process will enable us to analyze and compare the accuracies of different models in predicting personality traits based on the given dataset.

IV. Prototype implementation

A. Random Forest

Random Forest is a machine learning technique used for classification and regression tasks. In order to build a robust and accurate model, random forest combines multiple decision trees. Each decision tree in the random forest is trained on a subset of the data [8]. Voting or averaging is used for combining the predictions from each individual tree to create the final classification or regression prediction. During the implementation of the random forest classifier, we imported the necessary libraries such as we imported the scikit-learn library. After importing the necessary libraries we took the train_sentences, train_labels, and test_sentences as parameters in the random forest classifier and created a pipeline. The pipeline consists of 3 steps which are a bag of words (bow), TF-IDF transformer, and classifier step. In the bag of words (bow) step the text is vectorized and converted into a matrix of token counts using the count vectorizer, then the TF-IDF transformer is applied on the matrix of token counts, and the matrix token counts are converted into TF-IDF values, and finally the random forest classifier is applied with 600 decision tree (estimators) in the ensemble as the n_estimators parameter of the RandomForestClassifier is set to 600. Lastly, the pipeline is fit to the training data, and the final predictions are obtained.

B. Naive Bayes

Naive Bayes is used for classification and, regression and it is a machine learning algorithm. [9]. For this task, first, we started with a dataset containing text samples and labels. Then we split the data into training and testing sets with a ratio of 90% and 10%. Later on for vectorization, we converted the text into numerical features using Count Vectorization. Afterward, we trained a Multinomial Naive Bayes classifier on the vectorized training data. For prediction, we used the trained model to predict the test data. Lastly, we calculated accuracy. Generated a classification report with metrics like precision and recall, and visualized performance with a confusion matrix.

C. Support Vector Machine

Support vector machine (SVM) is an algorithm that is used in machine learning. It is usually used for classification and regression problems [10]. It figures out a hyperlane to separate the data into different classes and also maximize the margin between the data. We used an SVM classifier with scikit-learn for the construction of a text classification pipeline. At first, we imported necessary library functions such as SVC for SVM classification and Pipeline for creating a sequential data processing workflow. The pipeline comprises three key steps: first, a 'Bag-of-words' transformation using CountVectorizer with custom text cleaning; second a TF-IDF transformation to weigh the importance of words in documents; and third, the inclusion of the SVM classifier. The pipeline is then trained on the training data, enabling it to prepossess text and classify it effectively. Subsequently, this trained pipeline is utilized to make predictions on test data. Finally, the code calculates and prints the accuracy of the SVM classifier's predictions, demonstrating a basic framework for text classification tasks. Users are encouraged to adapt and modify the code to suit their specific data and project requirements.

D. Convolutional Neural Network

Convolutional neural network (CNN) is a deep learning architecture mostly used for image and sequence data processing [11]. It is effective in capturing hierarchical patterns in data using convolutional layers, pooling layers, and fully connected layers. During the implementation of the CNN model we at first imported the necessary libraries such as Tensorflow Keras and then we built the CNN model. While building the CNN model we at first created a sequential model using the 'Sequential' class from Keras where this model will allow us to stack layers on top of each other sequentially. After the sequential model has been built we created the embedding layers, convolutional layers, max-pooling layers, dropout layer, flatten layer, and dense layer. The embedding layer embeds the input sequences and the convolutional layer consists of two 1D convolutional layers where the first layer has 64 filters with a kernel size of 3 and ReLU activation and the second layer has 128 filters with a kernel size of 3 and ReLU activation. The max pooling layers consist of two 1D maxpool layers with a poolsize of 2, the drop out layer has a dropout rate of 0.2, the flatten layer converts the output of the convolutional and pooling layers into a flat vector and the dense layer consists of two fully connected dense layer where the first layer consists of 128 units with ReLU activation and the second layer consists of 16 units with softmax activation for multiclass classification. After the CNN model was built the model was compiled and trained on the training data for 10 epochs.

V. Result Analysis

For each of the models that we have implemented in this research work, we have calculated the accuracy of the model and generated a confusion matrix and a classification report. The classification report of each of the models contains the F1 score, precision score, and recall score.

A. Random Forest

After training the model on the training set using the Random Forest Classifier we achieved an accuracy of 35.57% and generated a confusion matrix and classification report. The confusion matrix is given below.

B. Naive Bayes

After training the model on the training set using the multinomial naive bayes classifier we achieved an accuracy of 23.24% and generated a confusion matrix and classification report. The confusion matrix is given below.

C. Support Vector Machine

After training the model on the training set using the SVM classifier we achieved an accuracy of 35.32% and generated a confusion matrix and classification report. The confusion matrix is given below.

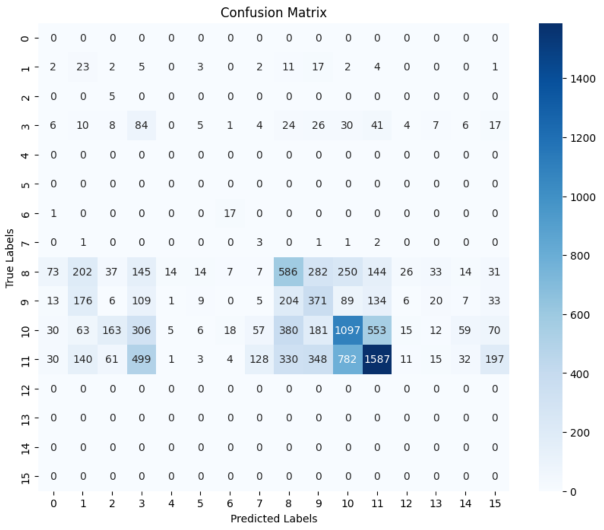

Fig. 2. Random forest confusion matrix

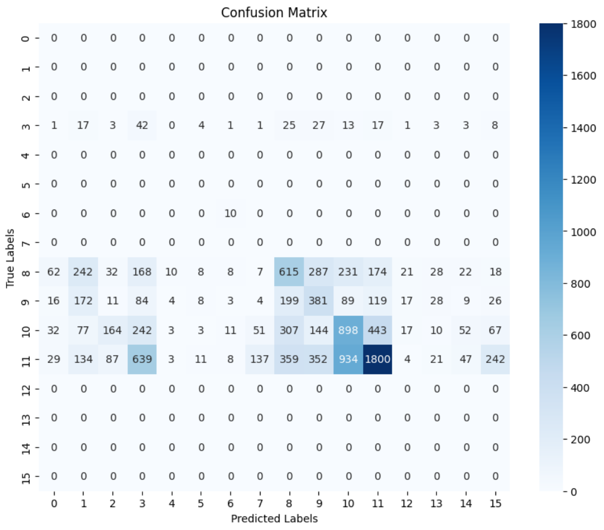

Fig. 4. SVM confusion matrix

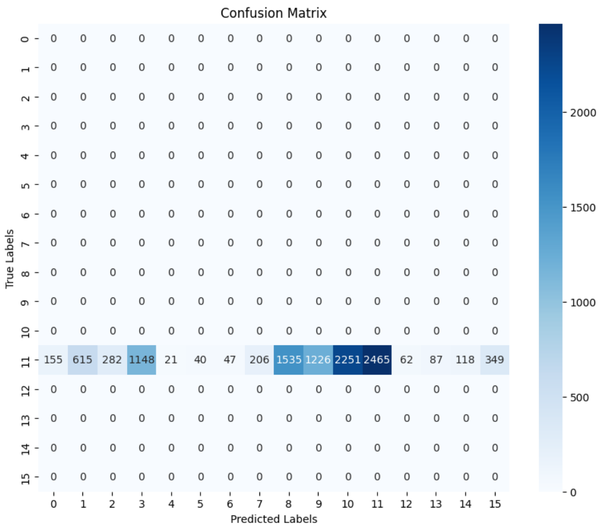

Fig. 3. Naive Bayes confusion matrix

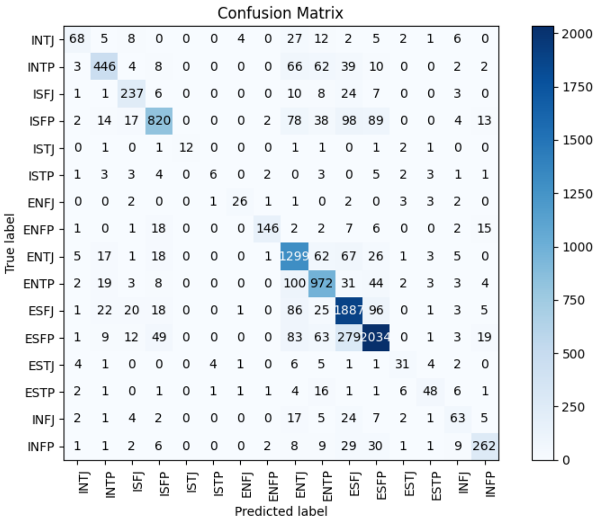

Fig. 5. CNN confusion matrix

D. *Convolutional neural network*

After training the model on the training set for 10 epochs using CNN we achieved an accuracy of 78.79% and generated a confusion matrix and classification report. The confusion matrix is given below.

E. *Comparison table*

TABLE II
MODEL PERFORMANCE METRICS

Model (%)	Random forest	Multinomial Naïve bayes	SVM	CNN
Accuracy	35.57	23.24	35.32	78.79
Precision	26.99	1.452	16.42	75.79
Recall	14.50	6.25	13.30	65.09
F1 Score	14.49	2.36	12.55	69.09

Table: Comparison of Random Forest Classifier, Multinomial Naive Bayes classifier, Support Vector Machine classifier, and CNN on performance evaluation metrics

The table II shows the comparative analysis between Random forest, Multinomial Naive Bayes, SVM and CNN based on the performance evaluation metrics such as accuracy, recall score, precision score, and F1 score. According to the performance evaluation metrics of the table it can be seen that CNN has the best performance overall compared to the other models. It has the highest accuracy of 78.79% and also the highest precision score, recall score and F1 score of 75.79%, 65.09% and 69.09% respectively. The Random Forest classifier has the second best performance as it achieved an accuracy of 35.57%. However, the SVM and Naive bayes classifier has the worst performance as it achieved low accuracy of 35.32% and 23.24% respectively.

VI. Conclusion

NLP has integrated into our daily lives. As our world becomes more interconnected and digitalized, NLP plays a vital role in communication, analysis, and prediction. Our research used NLP to figure out how people feel and what kind of person they are based on what they write. We used different models and tried to get the best accuracy. We plan to do more with this as NLP continues to change how we communicate in out digital world. Our work can be revolutionary as it can be implemented industrially and has industrial value. For example, in various service based app or website our project can increase user friendly experience.Dating or matrimony cite, online gaming, digital entertainment sites etc can analyze user personality and give them the best experience according to their personality.

References

[1] R. Jafari and B. H. Far, "Behavioral mapping, using nlp to predict individual behavior : Focusing on towards/away behavior," in *2022 International Conference on Advanced Enterprise Information System (AEIS)*, 2022, pp. 120–126.

[2] M. Anas, G. C, and K. G, "Machine learning based personality classification for carpooling application," in *2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS)*, 2023, pp. 77–82.

[3] P. William, Y. N, V. M. Tidake, S. Sumit Gondkar, C. R, and K. Vengatesan, "Framework for implementation of personality inventory model on natural language processing with personality traits analysis," in *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, 2023, pp. 625–628.

[4] T. Kanchana and B. S. E. Zoraida, "A framework for automated personality prediction from social media tweets," in *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*, 2022, pp. 698–701.

[5] "Analysis of news sentiments using natural language processing and deep learning - AI SOCIETY — link.springer.com," https://link.springer.com/article/10.1007/s00146-020-01111-x, [Accessed 04-09-2023].

[6] "MBTI Personality Types 500 Dataset — kaggle.com," https://www.kaggle.com/datasets/zeyadkhalid/mbti-personality-types-500-dataset, [Accessed 04-09-2023].

[7] P. Barak Or, "Solving The Class Imbalance Problem — towardsdatascience.com," https://towardsdatascience.com/solving-the-class-imbalance-problem-58cb926b5a0f#:~:text=There%20are%20several%20methods%20that,the%20balance%20of%20the%20dataset., [Accessed 05-09-2023].

[8] "Machine Learning Random Forest Algorithm - Javatpoint — javatpoint.com," https://www.javatpoint.com/machine-learning-random-forest-algorithm, [Accessed 04-09-2023].

[9] "Multinomial Naive Bayes Explained: Function, Advantages Disadvantages, Applications in 2023 — upGrad blog — upgrad.com," https://www.upgrad.com/blog/multinomial-naive-bayes-explained/, [Accessed 04-09-2023].

[10] "What is a support vector machine? — Definition from WhatIs — techtarget.com," https://www.techtarget.com/whatis/definition/support-vector-machine-SVM, [Accessed 04-09-2023].

[11] "What are Convolutional Neural Networks? — IBM — ibm.com," https://www.ibm.com/topics/convolutional-neural-networks, [Accessed 04-09-2023].