Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.049254

ARTICLE

Predicting Age and Gender in Author Profiling: A Multi-Feature Exploration

Aiman[1], Muhammad Arshad[1,*], Bilal Khan[1], Sadique Ahmad[2,*] and Muhammad Asim[2,3]

[1]Department of Computer Science, City University of Science and Information Technology, Peshawar, 25000, Pakistan

[2]EIAS: Data Science and Blockchain Laboratory, College of Computer and Information Sciences, Prince Sultan University, Riyadh, 11586, Saudi Arabia

[3]School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, 510006, China

*Corresponding Authors: Muhammad Arshad. Email: m.arshad@cusit.edu.pk; Sadique Ahmad. Email: ahmad01.shah@ieee.org

Received: 01 January 2024 Accepted: 09 March 2024 Published: 15 May 2024

ABSTRACT

Author Profiling (AP) is a subsection of digital forensics that focuses on the detection of the author's personal information, such as age, gender, occupation, and education, based on various linguistic features, e.g., stylistic, semantic, and syntactic. The importance of AP lies in various fields, including forensics, security, medicine, and marketing. In previous studies, many works have been done using different languages, e.g., English, Arabic, French, etc. However, the research on Roman Urdu is not up to the mark. Hence, this study focuses on detecting the author's age and gender based on Roman Urdu text messages. The dataset used in this study is Fire'18-MaponSMS. This study proposed an ensemble model based on AdaBoostM1 and Random Forest (AMBRF) for AP using multiple linguistic features that are stylistic, character-based, word-based, and sentence-based. The proposed model is contrasted with several of the well-known models from the literature, including J48-Decision Tree (J48), Naïve Bays (NB), K Nearest Neighbor (KNN), and Composite Hypercube on Random Projection (CHIRP), NB-Updatable, RF, and AdaboostM1. The overall outcome shows the better performance of the proposed AdaboostM1 with Random Forest (ABMRF) with an accuracy of 54.2857% for age prediction and 71.1429% for gender prediction calculated on stylistic features. Regarding word-based features, age and gender were considered in 50.5714% and 60%, respectively. On the other hand, KNN and CHIRP show the weakest performance using all the linguistic features for age and gender prediction.

KEYWORDS

Digital forensics; author profiling for security; AdaBoostM1; random forest; ensemble learning

1 Introduction

Author Profiling (AP) evaluates author's personal information based on various linguistic features, e.g., stylistic, semantic, and syntactic [1]. AP is considered an important task in digital security forensics. Also, it plays a key role in anti-harassment, targeted advertisement, analysis of text for personality identification, medicine, and marketing [2,3]. AP has multiple applications for educational cognitive purposes. Such as, researchers can find out the level of knowledge of exceptionally talented students by evaluating their writing skills. Also, AP can be used to identify various author's attributes from a given text [4]. Scammers and hackers employ false names, gender, age, and location on social

media to conceal one's identity [5]. Law enforcement organizations can use AP to track individuals using forged identities. Likewise, people exhibit vandalism, harassment, and Cyberbullying using alternative or fake identities to ensure that they never get caught [6,7].

AP could help to identify such cases using different natural language processing (NLP) algorithms [8]. From a marketing point of view, organizations will get to know each other by analyzing online websites and items. What kind of people comment on their items, and accordingly, they will promote their efforts towards a specific gender or age limit. It can also determine which age group, gender, occupation, etc., use certain products or comment on certain posts. This information can be used for targeted advertising and increasing revenue for different companies. Research into the AP domain shows that the language features of Facebook, status updates, tweets, messages, and blog posts allow us to evaluate the age and gender of authors accurately [9]. Deep Learning and Machine Learning significantly contribute to modeling various factors to optimize AP systems. Also, few traditional classification methods classify authors based on demographic factors like age, gender, and personality traits [1,10]. Additionally, researchers have explored linguistic features such as stylometric analysis, semantic and syntactic, and, more recently, deep learning techniques to improve classification accuracy. However, while existing studies have made significant strides in this field, there remains a research gap in effectively combining ensemble models like Random Forest and Adaboost for potentially enhanced performance in author profiling tasks, especially regarding age and gender classification.

Despite the existing research efforts in author profiling using various machine learning techniques and stylistic features, a significant research gap emerges in the lack of exploration and comparison of ensemble models combining Random Forest (RF) and AdaboostM1 (AB). While individual models have been examined extensively, more comprehensive investigations must be conducted into the potential performance enhancements and synergies that can be achieved by integrating these two powerful techniques [10]. Furthermore, most prior studies have focused on specific languages or datasets, neglecting the potential generalizability of ensemble approaches across diverse linguistic and textual domains [11]. This research gap warrants a thorough examination of the effectiveness of ensemble techniques in author profiling tasks and the potential improvements they can offer over traditional single-model approaches while also considering broader applicability across linguistic and demographic variations [12]. This research contributes to the advancement of author profiling methodologies by offering a novel ensemble-based approach that outperforms traditional single-model methods, thereby expanding the toolset available for researchers and practitioners. The key contributions of the paper are:

1. Investigate and improve the accuracy of AP for Roman Urdu text messages while evaluating authors' attributes (such as age and gender) based on linguistic features.
2. Analyze the FIRE'18-MAPonSMS dataset for AP purposes and propose an AdaBoostM1 and Random Forest (AMBRF) ensemble model that leverages multiple linguistic features.
3. Examine the effectiveness of this ensemble model compared to existing models. Perform a comprehensive comparative analysis of the proposed AMBRF model against well-known AP models from the literature, including J48 Decision Tree (J48), Naive Bayes (NB), K Nearest Neighbor (KNN), and Composite Hypercube on Random Projection (CHIRP), NB-Updatable, RF, and AdaboostM1.
4. Examine the advantages of ensemble models over single models regarding predictive accuracy and performance. Focus on Boosting as an ensemble technique for constructing a robust classifier from weak classifiers, aiming to enhance overall predictive performance.

The rest of the paper is organized: Section 2 presents the brief literature, and Section 3 discusses the proposed model. Section 4 presents the experimental setup, while Section 5 illustrates the results achieved and the discussion. Finally, Section 6 concludes the study.

2 State of the Art

The FIRE'18-MAPonSMS dataset is used in [13] for AP. They used a multilingual (English and Roman Urdu) SMS-based document and conducted different experimental analyses. They concluded that RF achieved the best accuracy of 73.714% for gender while using all 14 language-independent features together and an accuracy of 58.571% for the age group by using all 29 features together. For the best result the authors concluded that they obtained the best accuracy of 58.571% for the age. The overall result was compared with the baseline technique.

In [14], the authors focused on the ML techniques such as KNN, Support Vector Machine (SVM), Logistic Regression (LR), Radial basis Support Vector Machine function (RBFSVM), SVM Linear (SVML), (Convolution Neural Network (CNN). Seventeen stylometric features were used to train the model. The best accuracy for both detections was 92.45%, achieved from the English dataset, and 90.36 for gender classification. For the Spanish dataset, 89.68% and 88.88% for bot detection. The authors in [15] used various ML algorithms, which are SVM, LR, deep pyramid Convolution Neural Network (DPCNN), NB, Gaussian Naive Bays (GNB), NB complement (NBC), LR, and RF (Random Forest), Region Based Convolution Neural Network (RCNNs) for AP. They used stylistic and word-level features. The models do not predict unfrequented demographics, i.e., non, binary gender, and work that is not single-topic (e.g., manager, professional, and science). Predicting the date of birth year works best between 1980 and 2000 for that age is 20–40, but it is not good for older personalities.

The author in [16] proposed a technique for predicting Age and Gender where they used Multilingual (English and Spanish) corpus (PAN-2018) datasets and applied RF to classify age and gender. They used different features, i.e., Lexical, Grammatical category, Close words, Suffixes, and Signs. Precision, recall, and F1 scores are used as evaluation measures. The results obtained using only the training set show a better classification in gender than in age; however, in neither case, it exceeded 55% of the F1 measure. Besides, this measure lowers when the two classifiers are joined, reaching an F1 value between 40% and 44%. In [17] authors have used a combination of Semantic, Syntactic, and NLP as a feature, and then all these combinations are fed as an ensemble model that classifies age and gender. They employed a supervised Random Forest ensemble classifier for the AP using the PAN2014 dataset.

The only working language indicates readability criteria, function words, and structural features, which play a vital role in identifying the age and gender of the writer. For predicting an author's profile, researchers [18] proposed a technique for Twitter bots that can only categorize human gender as female and male. Both users were there in 11 Twitter bots, and from their profiles, only a hundred tweets were selected overall, and a hundred tweets were selected randomly. They focused on the Semantic feature category, which is present in the tweets. They joined those semantic features with other stylistic features and POS tags. They used different machine learning methods, having an ensemble model; they said that Adaboost is fruitful, and the F1 score is 0.99. For the English language, the results gain an accuracy of 89.17%. RF technique was used to predict the profile of an author. In [19] describes an approach to working with an author's profile PAN 2013 Challenge. This work is based on the idea of a linguistic method successfully used in other classification tasks, such as document writing. They considered three different features: Syntactic, stylistic, and semantic. Each represents a different aspect of the text. They extracted similarity relationships between attribute vectors in test files and center-specific

modality clusters for each method. The authors [20] presented a clear feature in the form of a group; each group is then together with appropriate pre-processing steps for each group. Structural trigrams, counts of Twitter's most important characteristics, and stylometric grouping were used.

The authors clarified that age and gender prediction as a classification job and character prediction are regression problems employing SVM and Support Vasomotor Rhinitis (SVMR), respectively. Researchers [10] focused on age and gender prediction and carried out experiments using deep learning (DL) methods, i.e., LSTM, Gated recurrent units (GRU), BI-LSTM, and CNN. BAT-AP-19, RUEN-AP-17 corpus, and SMS-AP-18 corpus datasets were used for training and testing. This research focused on POS features. The best accuracy was achieved when the Bi-LSTM classifier was used. Best accuracy was achieved accuracy = 0.882, F1 score = 0.839 and accuracy = 0.735, F1 score = 0.739 for age and gender, respectively, using LSTM. To predict the age, gender, education, language, country, and emotions of AP [21], researchers used SMO, RF, Bagging, and Instance bases learning with parameter k (KNN) and Lib SVM. Character (case word length), lexical (function word correlate), structural (document category Hyper Text Markup Language (HTML)), and POS features are extracted from the English Email dataset in this research. The best result was achieved using RF for the language classification, which is 84.22%. In this research [22], the author proposed a technique for age, gender, and country prediction. They used a linear, SVM, and MLP classifier as an ML technique. Consequently, we proposed the following AdaboostM1 with Random Forest (ABMRF) model to address the highlighted issues.

3 Proposed Methodology

All authors' text messages are fed into the system in .txt format. The proposed system architecture, depicted in Fig. 1, comprises six components. Upon inputting the text documents into the system, preprocessing steps are applied to each document. After the preprocessing stage, in phase 3, a comparative analysis was undertaken using two distinct strategies (feature selection) to predict the author's age and gender. This prediction was accomplished by examining the author's writing behavior through eight learning models: RF, AdaboostM1, CHIRP, J48, NB, NBupdatable, and ABMRF. The comparison was conducted based on the chosen strategy.

During the fifth phase, the model selects a file, typically the top one, and makes predictions for a specific class. Additionally, the class assigned to the test files is determined by the majority votes, with the highest count dictating the chosen class. The system's output is the predicted gender and age class for the test files.

In real-world applications, multi-class problems are prevalent. Therefore, understanding how RF operates in the context of Multi-class AdaBoostM1 becomes crucial. For instance, AdaBoostM1 can execute RF in multi-class scenarios, provided the application of AdaBoostM1 achieved accuracy exceeding 50%. Both algorithms should be used correctly for optimal results [23].

Ensemble models are preferred over single models for two primary interconnected reasons. An ensemble model can yield more accurate predictions and outperform individual contributing models. It can also reduce the variance in predictions and enhance overall model performance. Consequently, our focus has been on Boosting, an ensemble technique aiming to construct a robust classifier from a set of weak classifiers. This approach can improve a model's robustness and reliability. The proposed research methodology is illustrated in Fig. 1.

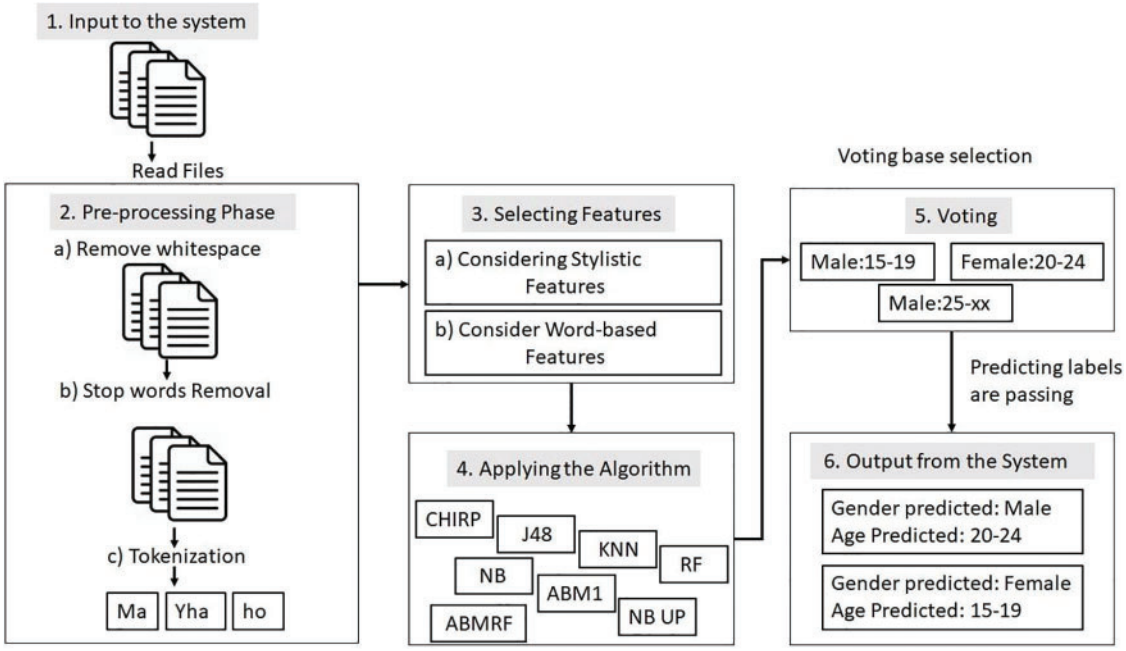

Figure 1: Proposed methodology

It is an ensemble learning method that combines multiple weak classifiers to create a strong classifier. It operates by iteratively training a sequence of weak classifiers, assigning higher weights to misclassified instances in each iteration. The weak classifiers are typically simple classifiers, such as decision stumps. During training, ABM1 adjusts the weights of the training instances based on their classification accuracy. The subsequent weak classifiers focus more on the previously misclassified instances, allowing them to correct the mistakes made by previous weak classifiers. The final prediction is determined by combining the predictions of all weak classifiers, weighted by their accuracy. The process is discussed in Algorithm 1.

Algorithm 1: Simple AdaBoost

 # Step 1: Initialize weights of training instances
InitializeWeights (W, N) # where N is the number of training instances
for i in range (N):
$W[i] = 1 / N$
 # Step 2: AdaBoost Iterations
for t in range (T): # where T is the number of iterations
 # Step 2a: Train a weak classifier
h_t = TrainWeakClassifier (X, Y, W) # X is the training data, Y represents the labels
 # Step 2b: Compute the weighted error
$epsilon_t$ = ComputeWeightedError (h_t, X, Y, W)
 # Step 2c: Calculate the weight of the weak classifier
$alpha_t = 0.5 * ln\ ((1 - epsilon_t) / epsilon_t)$
 # Step 2d: Update the weights of training instances
UpdateWeights $(W, h_t, X, Y, alpha_t)$

(Continued)

Algorithm 1 (continued)

```
    # Normalize the weights
    W = NormalizeWeights (W)
# Step 3: Combine weak classifiers into a strong classifier
def StrongClassifier (X, T, classifiers, alphas):
    H = 0
    for t in range(T):
        H += alphas[t] * classifiers[t] (X)
    return sign (H)
```

In the aforementioned equations, x_i represents the feature vector of instance i, and y_i represents its true label. $h_t(x_i)$ represents the prediction of weak classifier h_t, for instance, i. The sign function returns $+1$ for positive predictions and -1 for negative predictions. The proposed ensemble model ABMRF works flow as described in Algorithm 2.

Algorithm 2: AdaBoostM1

```
# Step 1: Initialize the weights of training instances
InitializeWeights (W, N)
    for i in range (N):
        W[i] = 1 / N
# Step 2: AdaBoost Iterations (ABM1)
for t in range (T):   # where T is the number of iterations
    # Step 2a: Train a weak classifier
    h_t = TrainWeakClassifier (X, Y, W)
    # Step 2b: Compute the weighted error
    epsilon_t = ComputeWeightedError (h_t, X, Y, W)
    # Step 2c: Calculate the weight of the weak classifier
    alpha_t = 0.5 * ln ((1 - epsilon_t)/epsilon_t)
    # Step 2d: Update the weights of the training instances based on misclassifications
    UpdateWeights (W, h_t, X, Y, alpha_t)
    # Step 2e: Normalize the weights
    W = NormalizeWeights (W)
# Step 3: Combine ABM1 with RF
for t in range (T):
    # Step 3a: Train a Random Forest classifier with the current weights
    RF_t = TrainRandomForest (X, Y, W)
    # Step 3b: Compute the weighted error
    epsilon_t = ComputeWeightedError (RF_t, X, Y, W)
    # Step 3c: Calculate the weight of the RF classifier
    alpha_t = 0.5 * ln ((1 - epsilon_t)/epsilon_t)
    # Step 3d: Update the weights of the training instances based on misclassifications
    UpdateWeights (W, RF_t, X, Y, alpha_t)
    # Step 3e: Normalize the weights
    W = NormalizeWeights (W)
```

(Continued)

Algorithm 2 (continued)

 # Step 4: Combine the weak classifiers (RFs) into a strong classifier
def StrongClassifier (X, T, classifiers, alphas):
    $H = 0$
    for t in range (T):
        $H \mathrel{+}= alphas[t] * classifiers[t]\ (X)$
return *sign (H)*

The proposed model relies on AdaboostM1 (ABM1) and Random Forest (RF) to enhance analysis compared to other employed models. ABM1, a widely-used machine learning model, is specifically designed for classification tasks. The quasi-code for ABMRF is shown Algorithm 3.

Algorithm 3: Ensemble ABMRF

Initialize $D_1\ (i)$
For $i = 1$ to m such that $D_1\ (i) = 1/m$
    For $t = 1$ to T:
        h_i = None
            For $K = 1$ to K:
                ${}^{\theta}k$ = GenerateVector ()
                $h(x, {}^{\theta}k)$ using any Decision Tree Algorithm
                tree = ConstructDecisionTree ($T, {}^{\theta}k$)
                if h_i is None:
                h_i = tree
                else:
                h_i = CombineHypotheses (h_i, tree)
    Return h_i
    Get back hypothesis $h_t: X \to Y$ error:
        $$\epsilon_t = \sum_{ih_t(X_t) \neq y_i} D_t\,(i)$$
If $\epsilon_t > 1/2$, then set $T = t\text{-}1$ and abort loop
    Set $\beta_t = \dfrac{1}{1 - \epsilon_t}$
    Update D_t :
        $$D_i + 1\ (i) = \frac{D_{ti}}{Z_t} * \begin{cases} \beta_t & if\ h_t = (x_i) = y_i \\ 1 & if\ h_t = (x_i) = y_i \end{cases}$$
Output: $H\ (x) = \arg\max_{y \in Y} + \sum_{t:\ h_t(x)=y}^{t} \left(\log \dfrac{1}{\beta}\right)$

Why did we select an ensemble model rather than a single model?

1. AdaBoostM1 assigns higher weights to misclassified samples in each iteration, forcing the model to focus more on challenging instances. In contrast, Random Forest builds trees on random subsets of the data without explicit emphasis on misclassifications. Combining these two approaches allows the ensemble to give extra attention to difficult-to-learn patterns, contributing to a more robust model.

2. Random Forest inherently helps reduce overfitting by building multiple trees and averaging their predictions. Combining it with AdaBoostM1, which sequentially fits models to the residuals, can further enhance the ability to generalize well to unseen data.
3. AdaBoostM1 effectively handles imbalanced datasets by assigning higher weights to misclassified samples. Combined with Random Forest, which inherently has mechanisms for handling imbalanced data, the ensemble model can perform tasks with imbalanced class distributions better.
4. Random Forest provides a measure of feature importance, which can be beneficial in understanding the relevance of different features in the NLP (Natural Language Processing) task. The combination with AdaBoostM1 may further refine the importance rankings and contribute to a more accurate feature selection process.
5. AdaBoostM1 and Random Forest are built on different principles, and combining them introduces diversity to the ensemble. Diversity is often crucial in improving the model's overall performance, as different models may capture different patterns or nuances in the data.

4 Experimental Setup

The app runs on Google Colaboratory (Colab), which provides free CPU cloud services via TensorFlow. The Google Colab has a 33 GB hard drive, 13 GB of RAM, and a 2-core Xeon 2.2 GHz processor. Python 3. x (3.6) runs and tests the system. The output produced using stylometric features is given as input to ML algorithms. For the experimental setup, our model uses 10-fold cross-validation to evaluate the performance of the model. We conducted our experiments using eight different ML algorithms: NB, J48, KNN, CHIRP, AdaBoostM1, NB Updatable, RF, and ABMRF.

4.1 Dataset Description

This study focuses on FIRE'18-MAPonSMS, a Roman Urdu dataset available at https://lahore.comsats.edu.pk/cs/MAPonSMS/de.html. The age and gender categories with their counts are presented in Table 1. FIRE'18 MAP on the SMS dataset consists of testing and training instances. Training consists of 350 instances and 1 truth file having mentioned gender as defined as male or female, while age is sub-categorized as the following groups: 15–19, 20–24, 25–XX. In training 350 instances, there are 210 male 18 files and 140 female files, while age-wise from 15–19, there are 108 records; from 20–24, there are 176 records 25–XX, there are 66 records. Testing consists of 150 instances, which will be used to test the proposed research model.

Table 1: Age and gender category and count

Category	Count (Training instances)
Gender	Male = 210, Female = 140
Age	15–19 = 108, 20–24 = 176, 25–XX = 66

4.2 Stylistic Features

The words or sentences are arranged in such a way that it gives us understanding. Narrative perspectives, the structure of stanzas, and position are stylistic features. The (14) Stylistic features are shown in Table 2.

Table 2: Stylistic features with descriptions

Features	Description
Average word length	First, it counts the total word used in a text document and then finds its average.
Average sentence length	First, it counts the total sentence used in a text document and then finds its average.
Percentage of words with six and more letters	It is the percentage of only those words which have six and more letters.
Percentage of words with two and three letters	It is the percentage of only those words which have two and three letters.
Percentage of question sentences	It is the percentage of question sentences in a text document.
Percentage of semicolons	It is the percentage of semicolons sentences in a text document.
Percentage of punctuations	First, it counts the total punctuation used in a text document and then finds its percentage.
Percentage of comma	It can be defined as the count of the total comma sentences used in a text document and then finding its percentage.
Percentage of short sentences	It counts only those sentences which have a length of a word less than eight and then finds the percentage of only those sentences.
Percentage of long sentences	It counts only those sentences which have a length of a word greater than 15 and then finds the percentage of only those sentences.
Percentage of capitals	It can be defined as the count of the total capital letter used in a text document and then finding its percentage.
Percentage of colons	It is the percentage of colons used in a text document.
Percentage of digits	It can be defined as the count of the total digit used in a text document and then finding its percentage.
Percentage of full stop	It can be defined as the count of the total full-stop sentences used in a text document and then finding its percentage.

4.3 Word-Based Features

The word-based features depend upon the length of the word. We can define these features based on the dataset we are using word-based features are based on the words used in a document [4]. The Word-Based features used in this study are presented in Table 3.

Table 3: Word-based features with descriptions

Features	Description
Mean number of characters per word	It can find the total number of characters in a document and the mean per word at the end. All these characters are divided by the total number of characters.
Stop words	Count the number of stop words in a document.
First-person pronoun	Count the number of first-person pronouns in a document.
Usage of words composed by two or three-character	Count only those words which have two and three letters.
Usage of words composed by six or more character	Count only those words that have six or more letters.
Total number of words	Count the total number of words in a document.

4.4 Pre-Processing Steps

Pre-processing comprises three phases, and traditionally, text pre-processing has held significant importance in NLP. It involves simplifying language to enhance the effectiveness of machine learning algorithms. The dataset underwent the following three preprocessing steps:

4.4.1 Remove Whitespaces

Almost all textual data in the world contains white space, the removal of which provides readability and ease of understanding [14]. To remove the starting and ending spaces of a text line, we have used the strip () function in Python. This is illustrated in Fig. 2. Part A shows white space before the word "Jaldi." This space is called the starting space of a sentence. Part B shows the preprocessed text; when the text is preprocessed, the white space is removed, so there is no space before the word "Jaldi".

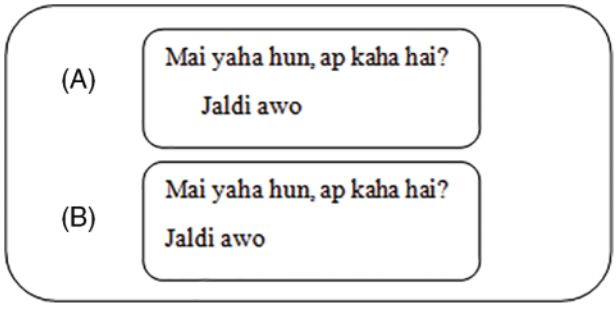

Figure 2: Remove whitespaces

4.4.2 Stop Words Removal

Stop words have no significant semantic relation to the context in which they exist [15]. In this phase, stop words are removed using Python code, e.g., in the example shown in Fig. 3, Part A has "Mai" and "hai," both considered stop words. In Part B, the stop words are removed when the text is preprocessed.

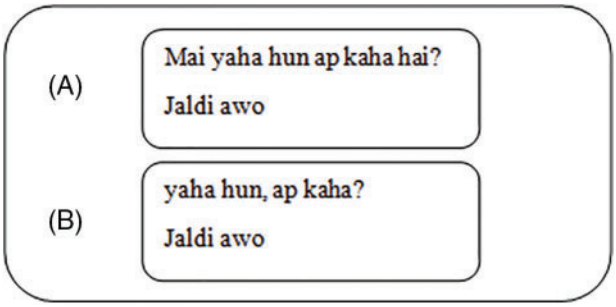

Figure 3: Stop word removal

4.4.3 Tokenization

It is the method by which text is broken down into smaller pieces known as tokens [18]. Words, numbers, and punctuation marks are supposed to be tokens. In this preprocessing phase, the sentences are tokenized using the split () function in Python, which separates the tokens inside the file by identifying the space between the tokens inside the sentence. In Part A of Fig. 4, the text is neither preprocessed nor tokenized. In Part B, each word is tokenized.

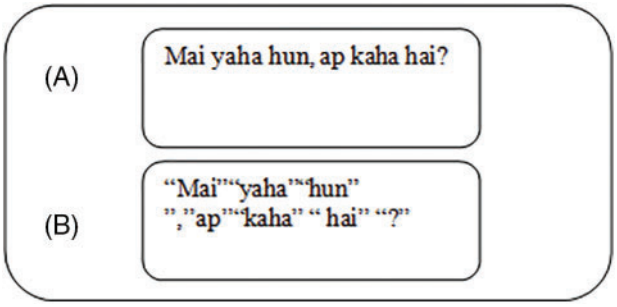

Figure 4: Tokenization

4.5 Training and Evaluation

Model training and testing are the core phases of any ML-based analysis. To this end, the study focuses on 10-fold cross-validation [19,23], which is a process for assessment that splits the complete data into ten subgroups of equal sizes; one subgroup is used for testing, whereas the rest of the subgroups are used for training, continuing until each subgroup has been used for testing [24–29]. The performance of the proposed and other ML models is evaluated using standard assessment measures, including accuracy, precision, recall, F-measure, and MCC (Matthews Correlation Coefficient) [20–22]. These can be calculated as shown in Eqs. (1)–(5).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{F–measure} = 2 \times \left[\frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}\right]. \tag{4}$$

$$\text{MCC} = \frac{\text{TP} * \text{TN} - \text{FP} * \text{FN}}{\sqrt{(\text{TP} + \text{FP})\,(\text{TP} + \text{FN})\,(\text{TN} + \text{FP})\,(\text{TN} + \text{FN})}} \tag{5}$$

5 Experimental Results and Analysis

This section provides a detailed discussion of experimental results. The term "Classifier" denotes the machine learning algorithm, including NB, NB Updatable, J48, KNN, RF, CHIRP, AdaBoostM1, and ABMRF are employed to generate numeric scores. All the evaluations, including precision, recall, F-measure accuracy, and MCC, are derived from the Confusion Matrix (CM).

The "Classifier" indicates the ML algorithm we applied to produce NB, NB Updatable, J48, KNN, RF, CHIRP, AdaBoostM1, and ABMRF numeric scores. All the evaluations, including precision, recall, F-measure accuracy, and MCC, are derived from the CM. The analysis determines the superior performance of the proposed model compared to other models. Table 4 shows each technique's Correctly Classified Instances (CCI) and Incorrectly Classified Instances (ICI).

Table 4: CCI and ICI for age and gender using stylistics features

Technique	Age		Gender	
	CCI	ICI	CCI	ICI
J48	168	182	242	108
RF	188	162	246	104
NB	124	226	214	136
IBK	172	178	230	120
CHIRP	180	169	138	211
NB updatable	124	226	214	136
AdaBoostM1	185	165	237	113
ABMRF	190	160	249	104

The outcomes achieved via precision, F-measure, and recall for age are shown in Table 5. Some table rows, e.g., Table 5, contain a (?) sign. Here, the question mark sign is used instead of the message "DIV/0!" due to its value "0". In the confusion matrix, according to different equations, the division cannot be performed if some values need to be divided and that value becomes "0". Nevertheless, in the tables, we used "?" instead of "DIV/0!"

Table 5: Precision, recall, and F-measure for age using stylistics features

Algorithm	Age		
	Precision	Recall	F-measure
J48	0.563	0.537	0.41
RF	0.456	0.48	0.46

(Continued)

Table 5 (continued)

Algorithm	Age		
	Precision	Recall	F-measure
NB	0.483	0.491	**0.483**
IBK	?	0.529	?
CHIRP	0.409	0.354	0.344
NB updatable	0.562	0.516	0.368
AdaBoostM1	0.409	0.354	0.344
ABMRF	**0.720**	**0.543**	0.420

Here, we have just shown the average values of each attribute. When precision, recall, and F-measurement are evaluated using stylistic features, the suggested ABMRF surpasses existing approaches, attaining 0.723, 0.543 for precision and recall, while for F-measure RF produced 0.46, and, respectively. Conversely, CHIRP and AdaboostM1 produces the poorest results in precision, recall, and F-measure, which are 0.409, 0.354, and 0.232, respectively.

The outcomes achieved via precision, F-measure, and recall for gender are shown in Table 6. Here, we have just shown the average values for each attribute. When precision, recall, and F-measure are evaluated using stylistic features, the suggested ABMRF surpasses existing approaches, attaining 0.723, 0.711, and 0.687 for precision, recall, and F-measure, respectively. Conversely, CHIRP produces the poorest results in precision, recall, and F-measure, which are 0.309, 0.395, and 0.232, respectively.

Table 6: Precision, recall, and F-measure for *Gender* using stylistics features

Algorithm	Gender		
	Precision	Recall	F-measure
J48	0.716	0.703	0.675
RF	0.686	0.691	0.684
NB	0.309	0.395	0.232
IBK	0.67	0.677	0.666
CHIRP	0.59	0.611	0.554
NB updatable	0.654	0.657	0.655
AdaBoostM1	0.590	0.611	0.554
ABMRF	**0.723**	**0.711**	**0.687**

Moreover, Table 7 shows the outcomes of each technique using MCC for age and gender. These analyses show the better performance of ABMRF, with an accuracy of 54.29% and 71.14% for age and gender, respectively. However, using stylistic features, CHIRP shows the weakest performance.

Table 7: MCC for stylistic features

Algorithm	MCC	
	Age	Gender
J48	0.168	0.365
RF	0.108	0.34
NB	0.097	−0.077
IBK	?	0.305
CHIRP	0.189	0.117
NB updatable	0.146	0.279
AdaBoostM1	−0.003	0.117
ABMRF	**0.201**	**0.384**

Comparing ABMRF with other applied techniques, the best result was achieved by ABMRF, i.e., 54.29% for age. On the other hand, the worst performance of NB and NB Updatable is noted with an accuracy of 35.4%, while ABMRF achieved the best result of 71.14%, whereas the worst performance of NB is noted with an accuracy of 35% for age while CHIRP is noted with an accuracy of 39.54% for gender. The accuracy of stylistic features (age and gender) is illustrated in Table 8.

Table 8: Accuracy using stylistic features

Algorithm	Accuracy (Age)	Accuracy (Gender)
J48	48%	69.14%
RF	53.71%	70.29%
NB	35.43%	61.14%
IBK	49.14%	65.71%
CHIRP	51.58%	39.54%
NB updatable	35.43%	61.14%
AdaBoostM1	52.86%	67.71%
ABMRF	**54.29%**	**71.14%**

5.1 Result Analysis Using Word-Based Features

The best result achieved by ABMRF for age was 177, and KNN misclassified 173 instances. For gender, AdaBoostM1 gained the best result, i.e., 211, while AdaBoostM1, i.e., 139, achieved the worst result among the ten classifiers. The outcomes achieved via CCI and ICI are illustrated in Table 9.

The outcomes achieved via precision, F-measure, and recall for age are shown in Table 10. Here, we have just shown the average values of each attribute. When precision, recall, and F-measure are evaluated using word-based features, the suggested NB and NB updatable surpass existing approaches, attaining 0.506 for recall, 0.489, and 0.484 for precision and f-measure, respectively. CHIRP and KNN, on the other hand, produce the poorest results in precision, recall, and F-measure, which are 0.417 and 0.362 using CHIRP and 0.489 using KNN, respectively.

Table 9: CCI and ICI using word-based features

Technique	Age		Gender	
	CCI	ICI	CCI	ICI
J48	174	176	209	141
RF	175	175	210	140
NB	176	174	199	151
IBK	146	204	205	145
CHIRP	175	174	154	195
NB Updatable	176	174	199	151
AdaBoostM1	175	175	211	139
ABMRF	177	173	210	140

Table 10: Precision, recall, and F-measure for age using word-based features

Technique	Age		
	Precision	Recall	F-measure
RF	?	0.5	?
J48	0.453	0.497	0.447
NB updatable	**0.489**	0.503	0.484
AdaBoastM1	?	0.5	?
NB	**0.489**	0.503	**0.484**
IBK	0.414	0.417	0.415
CHIRP	0.368	0.501	0.362
ABMRF	?	**0.506**	?

The outcomes achieved via precision, F-measure, and recall for gender are shown in Table 11. Here, we have just shown the average values of each attribute. When precision, recall, and F-measure are evaluated using word-based features, the suggested CHIRP, AdaBoostM1, and KNN surpass existing approaches, attaining 0.700 for recall, 0.603, and 0.588 for precision and F-measure, respectively. CHIRP, J48, and KNN, on the other hand, produce the poorest results in precision, recall, and F-measure, which are 0.571, and 0.441 using CHIRP and 0.319 using KNN, respectively.

Table 11: Precision, recall, and F-measure for gender using word-based features

Technique	Gender		
	Precision	Recall	F-measure
RF	?	0.6	?
J48	0.571	0.597	0.558
CHIRP	0.700	0.441	0.319

(Continued)

Table 11 (continued)

Technique	Gender		
	Precision	Recall	F-measure
AdaBoastM1	0.579	**0.603**	0.564
NB	**0.595**	0.569	0.573
IBK	0.591	0.586	**0.588**
NB updatable	**0.595**	0.569	0.573
ABMRF	?	0.600	?

For age, the best outcomes gained by NB and NB Updatable, i.e., 0.153 while the worse result gained by CHIRP was 0.028. For gender the best result achieved by NB Updatable, i.e., 56.8571% while the worse result achieved by J48, i.e., 0.094. The illustration of MCC are shown in Table 12.

Table 12: MCC for age using word-based features

Technique	MCC	
	Age	Gender
RF	?	?
J48	0.114	0.094
CHIRP	0.028	0.138
AdaBoastM1	?	0.108
NB	**0.153**	**0.152**
IBK	0.053	0.148
NB updatable	**0.153**	0.14
ABMRF	?	?

Comparing all applied eight techniques, the best result was achieved by ABMRF, i.e., 50.5714%, while the worst performance of KNN is noted with an accuracy of 41.71% for age. For gender, AdaBoostM1 gained 60.29%, and the worst performance of CHIRP is noted with an accuracy of 44.1261%. The accuracy of word-based features is illustrated in Table 13.

Table 13: Accuracy of word-based features

Technique	Accuracies (Age)	Accuracies (Gender)
J48	49.71%	59.71%
RF	50%	60.00%
NB	50.29%	56.86%
IBK	41.71%	58.57%

(Continued)

Table 13 (continued)

Technique	Accuracies (Age)	Accuracies (Gender)
NB updatable	50.2857%	56.8571%
CHIRP	50.1433%	44.1261%
AdaBoostM1	50.00%	**60.29%**
ABMRF	**50.5714%**	60%

Fig. 5 shows the percentage difference (PD) for the age of the better technique compared with the rest of the other employed techniques. PD is calculated as shown in Eq. (6).

$$\text{PD} = \frac{n_1 - n_2}{\frac{n_1 + n_2}{2}} * 100 \tag{6}$$

where n_1 represents the value of ABMRF and n_2 represents the value of other techniques. For Stylistic features, the illustration shows the minimal difference between ABMRF and RF is 1.07% and the highest difference is achieved using ABMRF with NB updatable is 42.04%. While for word based features the minimal difference between ABMRF and NB updatable is 0.57% and the highest difference between ABMRF and IBK is 19.21%.

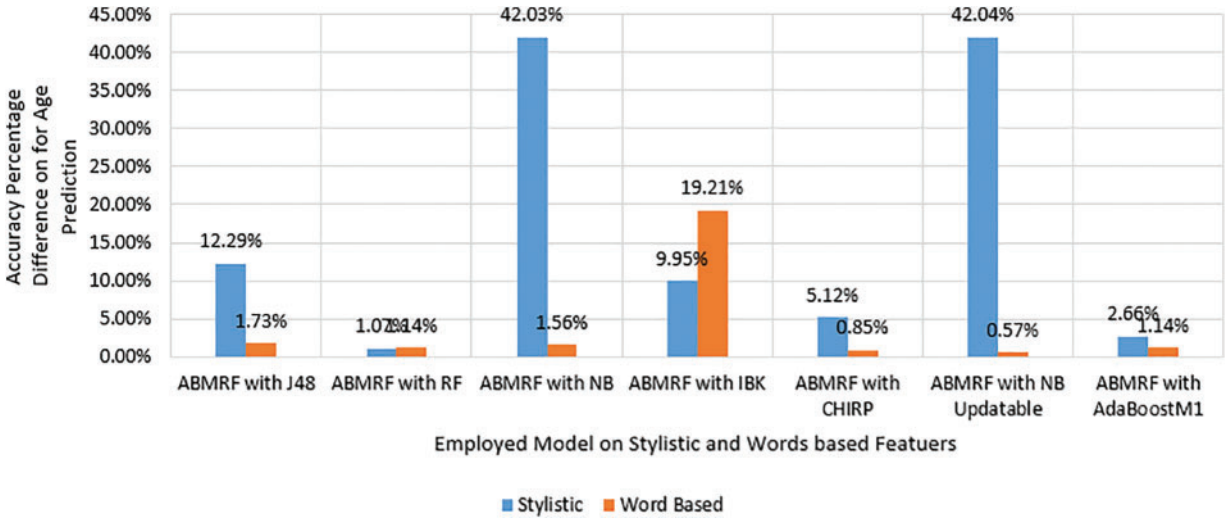

Figure 5: PD for age

Fig. 6 shows the PD for gender using Stylistic features. The illustration shows the minimal difference between ABMRF and RF is 1.21%, and the highest difference between ABMRF with CHIRP is 57.10% using stylistic features. While for word based features the minimal difference between ABMRF and RF is 0% and the highest difference between ABMRF with CHIRP is 30.49%.

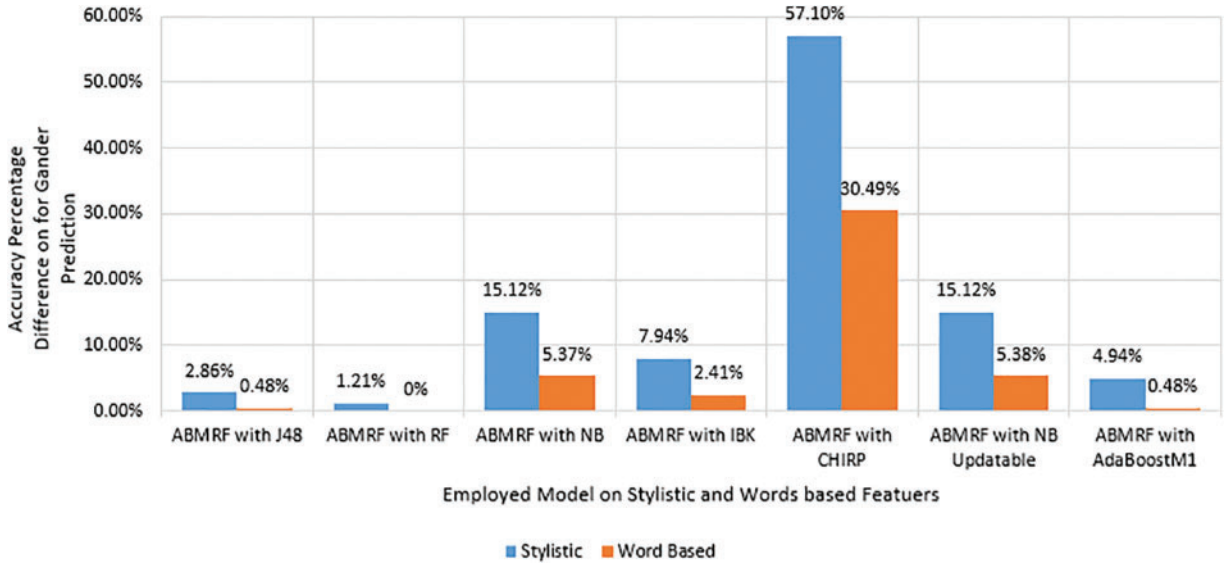

Figure 6: PD for gender

There are two primary reasons to prefer an ensemble model over one interconnected. An ensemble model can outperform any contributing model in terms of performance and accuracy. It may also stop the prevalence or dispersion of predictions and model performance. As a result, we focused on Boosting, an ensemble strategy for attempting to build a strong classifier from a set of weak classifiers. It may also improve the overall robustness or reliability of a model. With AdaBoostM1, we have ensemble RF. In an ML project, these are significant considerations, and we may sometimes choose one or both qualities from a model. The following are some of the reasons why we chose ensemble ABMRF. It gives variable weight, which helps identify the variable with a beneficial influence. ML models are frequently overfitted, but RF classifiers are not. We have different amounts of text in each file in this situation. Furthermore, when a class is rarer than other classes in the data, as we have, it may automatically balance datasets. Ensemble classifiers also outperform nonlinear classifiers on a wide range of tasks.

As mentioned in the analysis, ABMRF outperforms in contrast with other working classifiers to improve accuracy. This study focuses on eight diverse ML techniques for AP detection. The techniques are evaluated using multiple assessment measures on a Fire'18-MAPonSMS dataset.

ABMRF has been due to its better performance than other models employed in this study on the Fire'18-MAPonSMS dataset. We compared CHIRP, J48, RF, NB, IBK, AdaboostM1, NB Updatable, and ABMRF techniques and found that the Ensemble ABMRF is the most optimal age and gender classification for AP. Tables 8 and 13 present the accuracy analysis of age and gender, which shows the better performance of ABMRF, which achieved an accuracy of 54.2857% for age prediction and 71.1429% for gender prediction calculated on stylistic features. Regarding word-based features, age and gender were considered in 50.5714% and 60%, respectively. On the other hand, KNN and CHIRP show the weakest performance using all the linguistic features for age and gender prediction.

6 Conclusion

The focus of this research is to improve the accuracy of AP. The first and foremost approaches discussed in this research are ML approaches to achieve this goal. The dataset was taken from the web address "FIRE 18-MAPonSMS" in Roman Urdu, and accuracy, precision, recall, F-measure, and MCC are used as evaluation metrics. The accuracy results from the proposed model are compared with those of J48, NB, NB Updatable, CHIRP, KNN, RF, and AdaboostM1. The results show that ABMRF performs best compared to the other used models. This study focuses on proposing an ensemble model based on RF and AdaboostM1. The proposed model outperforms other employed models; however, there are also some limitations. The proposed model is less interpretable, and any wrong selection can lead to lower predictive accuracy than an individual. Features tuning is very hard while mapping with both ensemble models. Moreover, it is a bit expensive in terms of space and time-consuming.

However, it is important to acknowledge certain limitations of the proposed model. Firstly, the model's interpretability is reduced, and incorrect selections may lead to decreased predictive accuracy compared to individual models. Additionally, feature tuning proves challenging when integrating RF and AdaboostM1 models in an ensemble. Furthermore, the model requires considerable resources in terms of space and time, resulting in higher computational costs. Future research directions in AP may involve broadening the focus to incorporate more demographic characteristics, incorporating deep learning strategies, addressing privacy issues, investigating cross-lingual profiling, and adjusting to new digital communication platforms, allowing a more thorough and flexible approach to understanding people through their online content. Furthermore, further investigations should involve larger datasets to expand the scope of analysis. Furthermore, using deep learning (DL) techniques can lead to better results. By doing so, a more comprehensive understanding of author profiling in Roman Urdu can be achieved, ultimately improving accuracy and effectiveness.

7 Declarations

I declare that this research paper, titled "Predicting Age and Gender in Author Profiling: AMulti-Feature Exploration" is an original work conducted by Aiman et al. The study focuses on detecting authors' age and gender in Roman Urdu text messages through an ensemble model (AMBRF) based on AdaBoostM1 and Random Forest. The research utilizes the Fire18 Map on SMS dataset, proposing a model contrasted with established approaches such as ABMRF, J48, NB, KNN, CHIRP, NBUpdatable, RF, and AdaboostM1.

Acknowledgement: The authors thank Prince Sultan University and the EIAS: Data Science and Blockchain Laboratory for their valuable support.

Funding Statement: The authors would like to acknowledge the support of Prince Sultan University for the Article Processing Charges (APC) of this publication.

Author Contributions: The authors confirm their contribution to the paper as follows: Study conception and design: Aiman, Bilal Khan, Muhammad Arshad; data collection: Aiman, Muhammad Arshad; analysis and interpretation of results: Aiman, Bilal Khan; draft manuscript preparation. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used in this study is online available at https://lahore.comsats.edu.pk/cs/MAPonSMS/de.html. The code for this research is available on GitHub at https://github.com/aiman-syed/Features-Extraction.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] I. Ameer, G. Sidorov, and R. M. A. Nawab, "Author profiling for age and gender using combinations of features of various types," *J. Intell. Fuzzy Syst.*, vol. 36, no. 5, pp. 4833–4843, 2019. doi: 10.3233/JIFS-179031.

[2] F. C. Hsieh, R. F. S. Dias, and I. Paraboni, "Author profiling from Facebook corpora," in *Proc. Int. Conf. Lang. Resour. Eval.*, Miyazaki, Japan, 2018, pp. 2566–2570.

[3] G. Farnadi, J. Tang, M. de Cock, and M. F. Moens, "User profiling through deep multimodal fusion experimental results: User profiling," in *Proc. Int. Conf. Web Search Data Min.*, Marina Del Rey, CA, USA, 2018, pp. 171–179.

[4] D. Radha and P. C. Sekhar, "Author profiling using stylistic and N-gram features," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 1, pp. 3044–3049, 2019. doi: 10.35940/ijeat.A1621.109119.

[5] C. Peersman, W. Daelemans, and L. V. Vaerenbergh, "Predicting age and gender in online social networks," in *Proc. 3rd Int. Workshop Search Min. User-Gen. Contents*, Glasgow, Scotland, UK, 2011, pp. 37–44.

[6] A. Sboev, T. Litvinova, D. Gudovskikh, R. Rybka, and I. Moloshnikov, "Machine learning models of text categorization by author gender using topic-independent features," *Procedia Comput. Sci.*, vol. 101, pp. 135–142, 2016. doi: 10.1016/j.procs.2016.11.017.

[7] T. R. Reddy, B. V. Vardhan, and P. V. Reddy, "N-gram approach for gender prediction," in *Proc. IEEE 7th Int. Adv. Comput. Conf. (IACC)*, Hyderabad, India, 2017, pp. 860–865.

[8] E. R. D. Weren *et al.*, "Examining multiple features for author profiling," *J. Inf. Data Manage.*, vol. 5, no. 3, pp. 266, 2014.

[9] T. K. Koch, P. Romero, and C. Stachl, "Age and gender in language, emoji, and emoticon usage in instant messages," *Comput. Hum. Behav.*, vol. 126, pp. 106990, 2022. doi: 10.1016/j.chb.2021.106990.

[10] M. A. Ashraf, R. M. A. Nawab, and F. Nie, "Author profiling on bi-lingual tweets," *J. Intell. Fuzzy Syst.*, vol. 39, no. 2, pp. 2379–2389, 2020. doi: 10.3233/JIFS-179898.

[11] A. Alajmi, E. M. Saad, and R. R. Darwish, "Toward an Arabic stop-words list generation," *Int. J. Comput. Appl.*, vol. 46, no. 8, pp. 8–13, 2018.

[12] J. Soler-Company, and L. Wanner, "On the relevance of syntactic and discourse features for author profiling and identification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguist.*, Valencia, Spain, 2017, pp. 681–687.

[13] A. Sittar and I. Ameer, "Multi-lingual author profiling using stylistic features," in *Proc. Working Notes FIRE*, Gandhinagar, India, 2018, pp. 240–246.

[14] S. Ouni, F. Fkih, and M. N. Omri, "Toward a new approach to author profiling based on the extraction of statistical features," *Soc. Netw. Anal. Min.*, vol. 11, no. 1, pp. 1–16, 2021. doi: 10.1007/s13278-021-00768-6.

[15] M. Wiegmann, B. Stein, and M. Potthast, "Overview of the celebrity profiling task at PAN 2019," in *Proc. Working Notes Conf. Labs Eval. Forum*, Lugano, Switzerland, vol. 2380, 2019.

[16] J. Silva *et al.*, "A method for detecting the profile of an author," *Procedia Comput. Sci.*, vol. 170, no. 2020, pp. 959–964, 2020. doi: 10.1016/j.procs.2020.03.101.

[17] A. Nemati, "Gender and age prediction multilingual author profiles based on comments," in *Proc. Working Notes FIRE 2018-Forum Inf. Retr. Eval.*, Gandhinagar, India, 2018.

[18] G. Kovács *et al.*, "Author profiling using semantic and syntactic features: Notebook for PAN at CLEF 2019," in *Proc. Working Notes Conf. Labs Eval. Forum*, Lugano, Switzerland, vol. 2380, 2019.

[19] U. Sapkota, T. Solorio, M. Montes-Y-Gómez, and G. Ramírez-de-la-Rosa, "Author profiling for English and Spanish text: Notebook for PAN at CLEF 2013," in *Proc. Working Notes Conf. Labs Eval. Forum*, Valencia, Spain, vol. 1179, 2013.
[20] A. Grivas, A. Krithara, and G. Giannakopoulos, "Author profiling using stylometric and structural feature groupings notebook for PAN at CLEF 2015," in *Proc. Working Notes Conf. Labs Eval. Forum*, Toulouse, France, 2015.
[21] D. Estival, T. Gaustad, S. B. Pham, W. Radford, and B. Hutchinson, "Author profiling for English emails," in *Proc. Int. Conf. Pacific Assoc. Comput. Linguist.*, Melbourne, Australia, 2007, pp. 263–272.
[22] H. A. Nayel, "NAYEL @ APDA: Machine learning approach for author profiling and deception detection in Arabic texts," in *Proc. Working Notes FIRE 2019-Forum Inf. Retr. Eval.*, Kolkata, India, 2019, pp. 92–99.
[23] Z. Zhang and X. Xie, "Research on AdaBoost.M1 with random forest," in *Int. Conf. Adv. Emerg. Trends Comput. Technol.*, Chennai, India, 2010.
[24] S. Mamgain, R. C. Balabantaray, and A. K. Das, "Author profiling: Prediction of gender and language variety from document," in *2019 Int. Ceronf. Inf. Technol. (ICIT)*, 2019, pp. 473–477.
[25] B. Khan, M. Arshad, and S. S. Khan, "Comparative analysis of machine learning models for PDF malware detection: Evaluating different training and testing criteria," *J. Cybersecur.*, vol. 5, pp. 1–11, 2023.
[26] R. Asif *et al.*, "Hyper-tuned convolutional neural networks for authorship verification in digital forensic investigations," *Comput., Mat. Continu.*, vol. 76, no. 2, pp. 1947–1975, 2023.
[27] M. Nawaz *et al.*, "Single and multiple regions duplication detections in digital images with applications in image forensic," *J. Intell. Fuzzy Syst.*, vol. 40, no. 6, pp. 10351–10371, 2021.
[28] J. Abdul, I. Sajid, I. T. Manzoor, R. Amjad, A. B. Saeed and S. Tanzila, "Analytical analysis of text stemming methodologies in information retrieval and natural language processing systems," in *Inst. Electr. Electron. Eng. Inc.*, Nov. 2023. doi: 10.1109/ACCESS.2023.3332710.
[29] L. S. Zoya, L. Rabia, M. Hammad, and S. M. J. Nor, "Assessing urdu language processing tools via statistical and outlier detection methods on urdu tweets," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 22, no. 10, pp. 1–31, 2023.