Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

The Use of Unsupervised Learning for Stylometric Feature Selection in Authorship Verification System

Lucia Dwi Krisnawati[1,*], Thomas Widiarya Budiman[1], Laurentius Kuncoro Probo Saputra[1], Haw Su Cheng[2]

[1] Department of Informatics, Faculty of Information Technology, Universitas Kristen Duta Wacana, 55224, Yogyakarta, Indonesia
[2] Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, Cyberjaya, Selangor, Malaysia

ARTICLE INFO

Keywords:
Author verification system; Stylometric features; Feature selection; Clustering; Classification

ABSTRACT

As one of the underlying problems of AV system is set on feature selection, this research focuses on finding the best combination of stylometric features in an Av system for Indonesian texts. To achieve this goal, 3 lexical features, 2 syntactic and 1 structural feature were combined into 20 feature combination sets. In discriminating these feature combinations, a clustering model, K-means was used and its outputs were measured with Purity score. To validate the robustness of feature combinations, they were experimented in an AV system using MKNN, KNN, and SVM classifiers in 5 experimental scenarios. It turns out that the most robust feature combination is the one containing both syntactic features plus the structural one, that is KF3. This best feature combination was applied to our AV system which was then tested with new datasets. The macro-average F-score of this test achieves 0.79, while the macro-average precision and macro-average sensitivity scores are 0.83 and 0.76 respectively.

1. Introduction

In Foucault's authorship concept, an author is associated with a text as his product bearing his personal styles, patterns, and characteristics [1,2]. The readers, on the other hand, will associate such writing traits to a specific writer which then stimulates texts to be the object of appropriation – "a property of its writer" [2,3]. Then it is closely related to pseudonym and anonymity which become a favored alternative in writing texts in a digital media. In a constructive perspective, pseudonym is used to liberate authors from pressure of readers' image on their works, to protect their real names and their privacy, or to have much freedom to express [2-6]. However, they function also as a mask of author's identity in writing texts containing twisted truth, fake news or even terror texts appearing on online alternative media.

Due to the increasing availability of massive news and messages, a need to identify text authorship arises. It cannot be simply addressed by taking the author ID, account or name since it is

* *Corresponding author.*
E-mail address: krisna@staff.ukdw.ac.id

https://doi.org/10.37934/araset.63.1.240254

very probable that they use the fake ID, account, or pseudonym [7]. Potthast *et al.,* [4] in noted that the advancement in the field of automatic Authorship Analysis has made it reliable enough to solve cases of disputed authorship, uncovering pseudonyms and unknown writers.

As a research area, Automatic Authorship Analysis (AA) includes Authorship Verification (AV) which attempts to determine whether a specific author wrote a questioned document [78]. Thus, given a set of documents of known authorship, the task of AV system is to determine whether a document with questioned authorship has been written by the same author. In the field of digital forensics and cybercrime, AV system is applied to disclose the authorship of black mailing, to detect online pornography posting, deceptive intent and fake news in e-commerce and social media, authors of child grooming texts, and to resolve copyright disputes [7,9-12]. In Humanities, AV system is useful for attributing anonymous or disputed literary works to known authorship and for revealing multiple aliases of the same users in social media [9].

One of the underlying problems in building an author verification system is set on the feature selection phase. For this reason, this research aims to solve this problem and focuses on finding the best combination of stylometric features which are reliable to represent authors' writing style on Indonesian texts. It assumes if the best combination of stylometric features could be obtained, then the authorship verification process -- which falls into a classification task -- will result in a high precision score and other evaluation measures. The feature combinations would be examined through an unsupervised learning model and measured with its purity rate, then they were cross-examined by applying them in three different classification models. This research hypothesizes that the purity rate of stylometric feature combinations is directly proportional to their accuracy, precision, and sensitivity scores.

2. Related Works

The earlier Author Verification (AV) systems have been built with profile-based approach which concatenated all training texts per author into a single file, and an aggregate representation of that author’s style is extracted from this file [13,14]. Meanwhile, the instance-based approach, which treats each training text as an individual instance of authorial style, is more preferable as it is mostly applied in Machine Learning models such as Latent Semantic Analysis and Latent Dirichlet Allocation, Siamese Neural Network, Naïve Bayes and Support Vector Machine [12,14-16]. The AV system reported in Kumar *et al.,* [17] applied IR-based method in which each training document is represented as vectors using Bag of Words model, while Hu *et al.,* [18] built a Topic Debiasing Representation Learning Model for stylometric representation learning in AV.

Beside models, features play an important role in an AV system. A previous study [17] experimented topical features taking form of non-uniform distributed term weight (NDTW). Other topical features such as n-grams are very robust in text classification but unsuitable for AV system due to its impotence in discriminating author's writing style [19]. The solution is to turn to the field of stylometry which is a study in Linguistics style adopting the use of mathematical-logical foundation and statistical analysis for identifying writers’ style of writing [20].

Identifying the best set of features are very challenging, therefore the majority of AV systems combine two or more stylometric features. Gunawan *et al*., [21] in combined the lexical and syntactic features taking form of character and word n-grams, type-token ratio, sentence, and paragraph lengths. The feature combination of bag of lexical n-gram, syntactic and topical modality were implemented in, while used character to word embeddings, word to sentence, and sentence to document embeddings [12,15].

Other components of AV pipeline that are interesting to survey are the similarity or distance metrices. In term of similarity metric, cosine similarity is still dominant in previous studies [9,17,22,23] compared to MinMax similarity or Standard Hausdorff Distance [9,24]. Using stylometric features in an Intrinsic Plagiarism Detection system, applied outlier analysis to uncover the sudden change of author's writing style which suggests an act of plagiarism [21]. The metrices used to evaluate AV systems are varied, i.e. accuracy, Area Under Receiver Operating Characteristic Curve, error rate, Precision, Recall, BCubed F-score [12,15,17,25].

Luyckx and Daelemans [26] reported that the majority of earlier AV systems focused on two or a few authors and used limited size of training data. However, the recent AV systems have dealt with both several authors and various sizes of training data. In term of training data usage, some AV systems relied on the available data collection such as the corpora of PAN 2023, PAN 2015, PAN 2014, while some systems preferred to build their own corpora [9,15,23]. Boenninghoff *et al*., [12] built a new large-scale corpus from short Amazon reviews with the size of ± 9+ Mio from 784,649 authors, whereas used 500 texts written by 100 authors for training data and 100 texts for test data [17]. Having no need of training data for their intrinsic plagiarism detection system, used only 31 test data for the evaluation of system performance [21].

Though research on AV has flourished well, AV system for Indonesian texts has yet to be fully explored. The previous study on Indonesian text using stylometric features was aimed to detect the changing of writing style of passages within a document by making use of lexical and syntactic features [21]. The research on AV system for familial language, Malaysian, was conducted by Tarmizi *et al.,* [16] using character and word n-grams with n ranging from 1-5 as its stylometric features. While quantified the stylometric features and combine those all features to measure an outlier, simply used single feature of each character or word n-grams, compared their performances and reported that character 3-grams are the most relevant features in identifying the author of KadazanDusun messages, a dialect of Malaysian Language [16,21]. The summary and highlight of the relevant literary sources are displayed in Table 1.

Table 1
The summary of relevant literary sources for this study

Features	Models/Methods	Evaluation Metrices	Sources	Additional information
Bag of words (non-distributed term weight)	Information Retrieval model	accuracy	[17]	Applying cosine similarity for comparison measurement
Character, word, sentence & document embeddings	Siamese Neural Network	Error rate, correlation analysis	[12]	
Bag of n=grams	LDA, LSA, PVDBOW	Accuracy AUROC	[15]	Applying character, lexical, syntactic, & topical levels n-grams
Character & word n-grams, type token ratio, sentence & paragraph lengths	Outlier analysis	Accuracy, F1-score	[21]	Stylometric feature for intrinsic plagiarism detection
Character & word n-grams	SVM, Naïve Bayes	Accuracy	[16]	Dataset is in KadazanDusun, a dialect of Malaysian Language

3. Proposed Method

As the aim of this research is to get the most relevant feature combination in verifying authorship of Indonesian texts, we proposed to use clustering to discriminate the feature combinations, then classification was applied to verify the robustness of these features. The architecture of our system is shown in Figure 1. Its workflow starts with the training and test data collection which were crawled from an online non-mainstream news portal. The next step is to preprocess the texts followed by feature extraction, feature combination and feature vector normalization. The following steps are the tasks of clustering and author verification along with their evaluation measures.

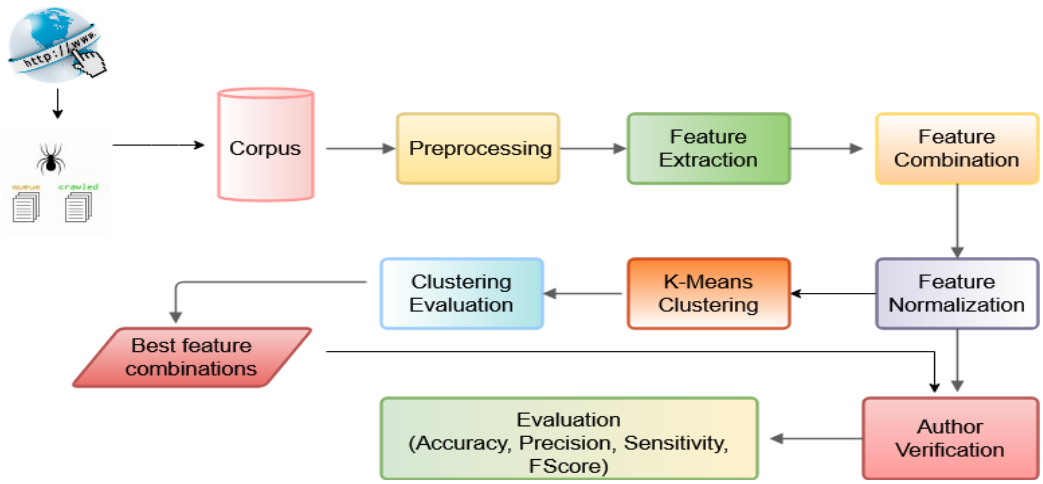


Fig. 1. The system architecture of the proposed method

3.1. Preprocessing

Though Indonesian language is formally written in Latin alphabet, texts or news written in online non-mainstream media may comprise a few Arabic words or sentences, loan words written with diacritics or special characters. For this reason, the first task in text normalization is to convert all character into ASCII format. Since the texts in the ASCII format comprise mixed cases, the case folding which converts all letters to lowercase was applied by using Python function *lower* from *string* library. The normalized news texts along with the information about their titles, authors, categories and date of creation were then saved in an Excell document as a basic csv database.

3.2. Feature Extraction and Combination

Before extracting features, it will be very useful to uncover the linguistic patterns capable of representing author's writing style. The previous study in Sarwar *et al.,* [24] compiled stylometric features on the lexical, syntactic and structural levels. Based on the work of Sarwar *et al.,* [24] and Gunawan *et al.,* [21], we decided to use three (3) lexical features, two (2) syntactic features and one (1) structural feature. These features are as follows:

i) The Relative Frequency of Punctuation Marks could be a reliable stylometric feature. To extract all punctuation marks in each news text, we used *string* library in Python and defined punctuation attribute to find their occurrences, then their relative frequencies were computed. This function has a default order of punctuations as displayed in Figure 2. It

returned a list of dictionaries whose values are the relative frequency of each punctuation in each document.

ii) The Relative Frequency of Stopword. Sastrawi stopword list which was used as it is available on Python library. The process of extracting stopword relative frequency is almost identical with the punctuation mark extraction. The difference is set on the preprocessing phase. Being a string, stopword extraction needs tokenization which was done by utilizing *word_tokenize* function in the NLTK library. The output of this function is a list of dictionary with stopword relative frequency as its values.

iii) The Relative Frequency of Alphabet. This feature is well known also as a character unigram and the same technique as in the relative frequency of punctuation marks was applied. Since the text has undergone case folding, the alphabet used here is simply the lowercase Latin alphabet from a-z and their relative frequencies.

iv) The Average of Sentence Length was extracted by means of *sent_tokenize* function of NLTK. After acquiring a list of sentences, each sentence was then parsed with *word_tokenize* function to get its tokens. The sentence length is simply measured by the total number of its tokens. From this list, the average sentence length could be calculated. The end output of this feature is a string with a value of the average sentence length.

v) The Average of Paragraph Length was extracted by splitting the text string based on newline (\n), then each of its elements was parsed using *sent_tokenize* function which returned a list of paragraph length. The count of sentences indicates the length of a paragraph, and the average paragraph length was computed from this list.

vi) Type-Token Ratio (TTR) becomes a stylometric feature since it measures the lexical richness of a text [27]. TTR gives the idea whether an author uses the same words over and over or various vocabulary for expressing the same thought. It is computed by dividing the number of types -- the unique words -- by the total number of tokens. In computing TTR, we discarded all tokens comprising of numeric characters, however tokens comprise alpha-numeric symbols were retained.

```
!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~
```

Fig. 2. The default order of punctuations in *string.punctuation*

The data structure of 6 features described earlier take into two forms, i.e. a list or array for the first three features and a string for the rest. To make things easier, we labelled the relative frequency of punctuation as Feature 1 or F1, in short, and so on till F6 which refers to the type-token ratio (TTR). Table 2 describes the feature labels and their references. The feature combination process was implemented by making use of *combinations* function of *itertools* module in Python. To combine features having different data structure, the *hstack* function from *numpy* was used. We combined 3 features in each iteration and thus the iteration tool resulted in totally 20 feature combinations as presented in Table 3.

Table 2
Feature labels and their references

Labels	Features
F1	Relative frequency of punctuations
F2	Relative frequency of stopwords
F3	Relative frequency of alphabets
F4	The Average sentence length
F5	The Average paragraph length
F6	The type-token ratio

Table 3
Names, composition, and length of feature combinations

No	Labels	Features	Length	No	Labels	Features	Length
1.	KF1	F1, F2, F3	181	11.	KF11	F2, F3, F4	150
2.	KF2	F1, F2, F4	156	12.	KF12	F2, F3, F5	150
3.	KF3	F1, F2, F5	156	13.	KF13	F2, F3, F6	150
4.	KF4	F1, F2, F6	156	14.	KF14	F2, F4, F5	125
5.	KF5	F1, F3, F4	59	15.	KF15	F2, F4, F6	125
6.	KF6	F1, F3, F5	59	16.	KF16	F2, F5, F6	125
7.	KF7	F1, F3, F6	59	17.	KF17	F3, F4, F5	28
8.	KF8	F1, F4, F5	34	18.	KF18	F3, F4, F6	28
9.	KF9	F1, F4, F6	34	19.	KF19	F3, F5, F6	28
10.	KF10	F1, F5, F6	34	20.	KF20	F4, F5, F6	3

The list of 20 feature combinations is displayed in Table 3. The list of the first feature combination (KF1) comprises 181 elements, hence it is being the lengthiest feature array. The element values of these feature combination vary greatly. As clustering and classification algorithms work better when features have relatively similar scales, we applied *MinMaxScaler* to normalize their values. The *MinMaxScaler* was computed using the Eq. (1). In its implementation, we used MinMaxScaler object offered by SKLearn Library and made use of *fit_trasform* function with each combination feature as its parameter. This function returned the rescaled feature combination values into the range [0, 1] which was then fed to the clustering model.

$$x''_{ij} = \frac{x'_{ij} - min(X_j)}{\max(X_j) - min(X_j)} \tag{1}$$

where X is the original value of a feature combination taking form of a 2-dimensional array, x' s an element of X in a row i and column j.

3.3. Clustering Process

We relied on K-means as a clustering model. The rationale is that it falls into a hard clustering model in which each data point belongs to one specific cluster only. Its implementation was realized through the use of *K-means* class from SKlearn. The K-means parameter *random_state* was set up to 1, while n in parameter *n_clusters* was defined equal to the number of authors in our training data. The first step in this clustering is to create an object for K-means class. The following step is to call the *fit_predict* function on the object with stylometric feature combinations as its parameter values. The distance among texts as data points to form clusters was measured using the Euclidean distance that was computed with Eq. (2).

$$d_{ij} = \sqrt{\sum_{v=1}^{p}\left(X_{iv} - X_{jv}\right)^2} \quad (2)$$

where X_{iv} represents the feature vector of individual data point i, X_{jv} represents the feature vector of individual data point j, p is the total number of features in a data point, while v is an index in p.

3.4. Classification Process

For author verification process, we applied three classifiers, i.e. Modified K-Nearest Neighbours (MKNN), K-Nearest Neighbours (KNN), and Support Vector Machine (SVM). The MKNN classifier was firstly utilized to cross-check the validity of robust features in clustering process, while KNN and SVM were meant to conduct the author verification process. However, in our experiments we run all classifiers in an author verification system along with the clustering process to see the correlation of their outputs and the robustness of feature combinations. on the last experimental setup, the most robust feature combination would be used as features in Author verification process

MKNN Classifier, in this study, we applied Modified K-Nearest Neighbour (MKNN) algorithm explained in [28]. Its difference from traditional K-Nearest Neighbour (KNN) is set on its technique in predicting the label of the test data. This technique comprises two steps i.e. a validation and KNN weighting. The validation is akin to a training process whereas each text in training data should be validated with its neighbours. The validation process is computed by applying the Eq. (3).

$$val(x) = \frac{1}{H}\sum_{i=1}^{H} S(lbl(x), lbl(N_i(x))) \quad (3)$$

$$S(a,b) = \begin{cases} 1 & a = b \\ 0 & otherwise \end{cases} \quad (4)$$

In Eq. (3), H is the number of examined neighbours from the total training data. In our experiment, we set up H with the value of 0.1. *lbl(x)* refers to a class assigned to a data point or document x, while $N_i(x)$ is the number of nearest neighbours of data x. S is a function to compute the similarity between the label of a test data to the labels of its nearest neighbours. The S function is computed by voting as shown in Eq. (4).

The second step of MKNN is to weight the vote so that it does not use a simple majority or plurality voting rule [28]. Each vote is set to be equal to $1/(d_e + \alpha)$ where d_e refers to a Euclidean Distance of a data point to its neighbour and α is a smoothing threshold. In this study we set up the value of α into 0.5. The weighted vote is then calculated by multiplying it with the validity score of each point. The weighted vote for the nearest neighbour is then computed with Eq. (5) as in Parvin *et al.,* [28].

$$\mathrm{W}(i) = Val(i) * \frac{1}{d_e+\alpha} \quad (5)$$

where W(i) and Val(i) refer to the weight and the validity of ith nearest neighbours in the training data.

The KNN and SVM classifiers were implemented by making use of the *KNeighborsClassifier* and *SVC* modules provided by the *SKLearn* library. In KNN, we defined K to be equal to {1, 3, 5, 7, 9, 11, 13,15, 17, 19}. This K values were then assigned as an array which was used to iterate the verification process in the experiment phase. As for SVM, we used the non-linear one with rbf kernel and the

value of gamma parameter was set to *auto*. Both classifier's tasks are to verify whether an inputted text with unknown authorship is written by one of the writers available in the training data or not.

4. Experimental Setup

In this section we present the experimental scenarios for the proposed methods of feature selection in an author verification workflow system. Before that, we present the data collection.

4.1. Datasets

As it is described in Figure 1, the data was collected automatically by crawling an online non-mainstream news portal from Seword portal (Seword is browsable at https://www.seword.com/). Seword is an oppositional news portal on which everybody, unnecessarily a journalist, can write and send his/her article to be published online. The news texts were scraped by the help of the Seword's API. The data extracted from this portal are author names, the news texts, their titles, the news topical categories (politics, sports, etc) and the date of issue. Due to author varieties, we selected authors who wrote more than 20 news texts. The training data comprises 200 news documents which were written by 10 different authors. Thus, each author is represented by 20 samples of their texts. As test sets, we provided 75 news texts which were written by 12 authors. The composition is that 55 texts were written by 11 authors, so each author contributed to 5 of their writings, while the other 20 documents were written by a single author. The organization of these datasets are presented in Table 4.

Table 4
Information on the dataset used in the experiment

Dataset labels	# Doc	# unique Author	Function
Dataset 1	200	10	Training data
Dataset 2	55	11	Test data
Dataset 3	20	1	Training & test data

4.2. Experiment Scenario

We designed 5 experimental scenarios which varied in using the classifier models, datasets or parameters required by the classifiers. The goals of the first experiment are to obtain the best stylometric feature combinations and the value of K in MKNN on those feature combinations. It observes also the correlation between the purity score and some evaluation measures in author verification system. The flow of the experiment 1 took exactly the same steps described in Figure 1. Both K-means and MKKN got the normalized feature combinations with MinMaxScaler as inputs. However, their outputs would be assessed separately. In the training process of MKNN, we run k-stratified cross-validation with k equal to 5. The data used for both training and test set is the dataset 1. With 20 feature combinations, 5 folds validation and 10 values of K in MKNN, the total number of experimental cases in experiment 1 is equal to 1000.

In experiment 2, we tried to observe the performance consistency of the best feature combinations in the experiment 1 by increasing the number of documents and authors in the training data which has the unbalance number of samples for its authors. We would like to observe whether the fluctuation of the purity and the F-1 scores keeps showing the correlation. The difference between experiment 1 and 2 is set on the training process of MKNN which dismissed the

implementation of k-stratified cross-validation. In this experiment, datasets 1 and 3 were used as training data, while dataset 2 became the test data.

The experiments 3 and 4 dealt with author verification process to observe the robustness of each feature combination when they were run on different classifiers. The KNN classifier was applied to verify authors in experiment 3, while in experiment 4, we applied SVM classifier. The datasets used and the experiment flow in these experiments are similar to those in experiment 1.

The aim of experiment 5 is to observe the performance of the proposed author verification system with MKNN and to examine whether the new data added to the training data would increase its performance. To achieve this goal, dataset 1 and 3 were employed as training data, while dataset 2 was used as a test set. Unlike the previous experiments, the experiment 5 did not include the clustering process.

4.3. The Evaluation Scenario

For evaluating the performance of our proposed method, we split up our datasets into three (3) categories which were labelled as datasets 1, 2, and 3. Dataset 1 functions as the training data. Dataset 2 comprising of 55 documents functions as purely test dataset, while the dataset 3 was used as both test and additional training datasets. It comprises 20 documents written by a single author. Table 4 presents the information on our datasets

To evaluate the clustering output, the *purity* measure was opted. This metric assesses whether the data has been clustered well by counting the number of correctly assigned documents and dividing it by the total number of data points. To enable it, each cluster should be labelled and as the model identifies a group of data points having the same label as the ground truth, then it could be said that it has clustered the data points very well. The purity metric was computed using Eq. (6).

$$Purity = \frac{1}{N} \sum_{i=j}^{N} max_j \left| c_i \cap t_j \right| \tag{6}$$

where N refers to the total number of documents, k is the number of clusters, c_i is an individual cluster in C, while t_j is a class labelled in the ground truth for classification task. In this scheme, k is equal to author's number, different from [29] which uses Elbow and k-Medoids to determine k.

For assessing the author verification outputs, we used the widely known metrices, namely macro-average Precision, macro-average Sensitivity, and macro-averaged F1-score which is calculated from Precision and Sensitivity. Each of these metrics is computed using Eq. (7-9) as follows.

$$Prec_{macro} = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{L} \tag{7}$$

$$Sensi_{macro} = \frac{\sum_{i=1}^{l} \frac{tp_i}{fn_i + fp_i}}{L} \tag{8}$$

$$F{-}1_{macro} = \sum_{i=1}^{l} \frac{2 * Precision * Sensitivity}{Precision + Sensitivity} \tag{9}$$

5. Result and Discussion

To achieve the goals of experiments 1 - 4, we created 3 two-dimensional arrays. The first array was used to save the evaluation results of all metrices in each experiment. the second array was

created from the first array by sorting its purity score first, then by F-1 score. It saves only a single test case from one feature combination which achieves the best purity score followed by its F-1 score. Figure 3 displays the result of array 2 for experiment 1, while Figures 4 - 6 show the visualization of the second array values for experiments 1, 2, 3 and 4 respectively.

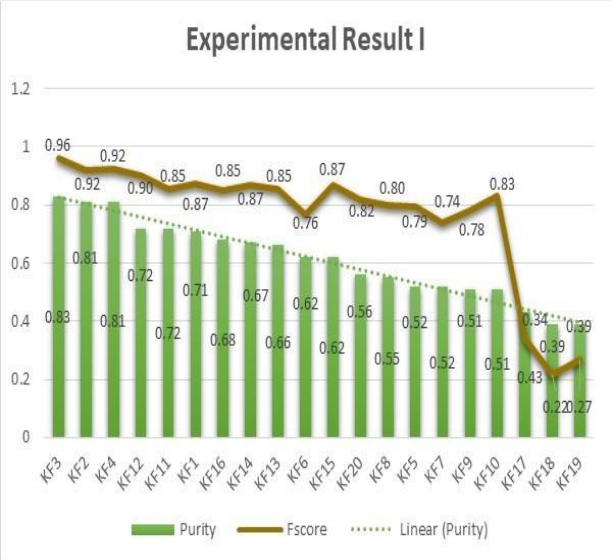
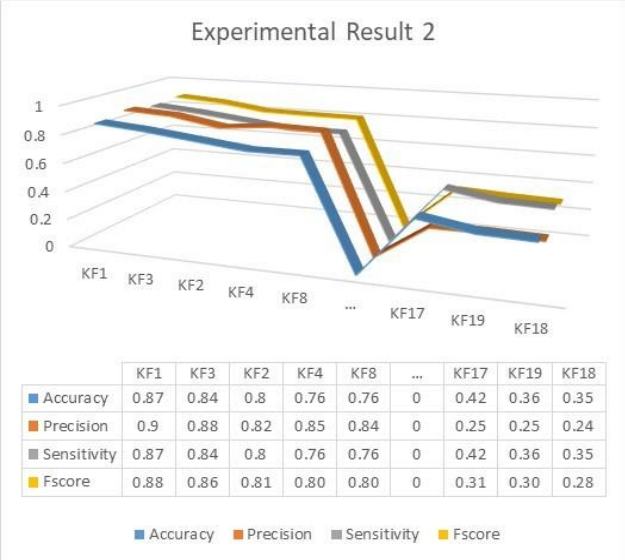


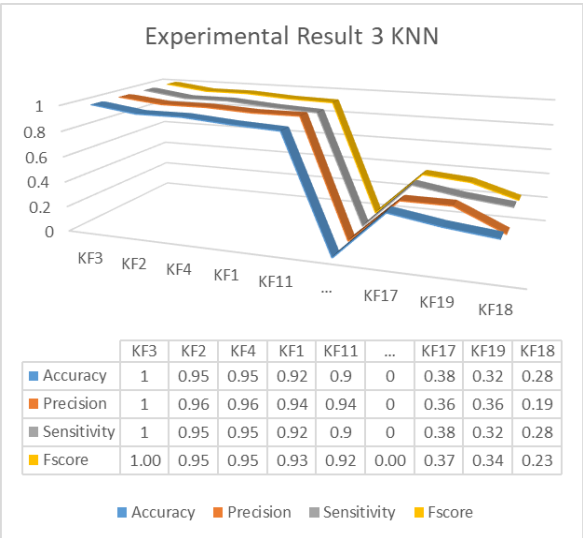
Fig. 3. The best scores of purities and F-1 of each feature combination in experiment 1



	KF1	KF3	KF2	KF4	KF8	...	KF17	KF19	KF18
Accuracy	0.87	0.84	0.8	0.76	0.76	0	0.42	0.36	0.35
Precision	0.9	0.88	0.82	0.85	0.84	0	0.25	0.25	0.24
Sensitivity	0.87	0.84	0.8	0.76	0.76	0	0.42	0.36	0.35
Fscore	0.88	0.86	0.81	0.80	0.80	0	0.31	0.30	0.28

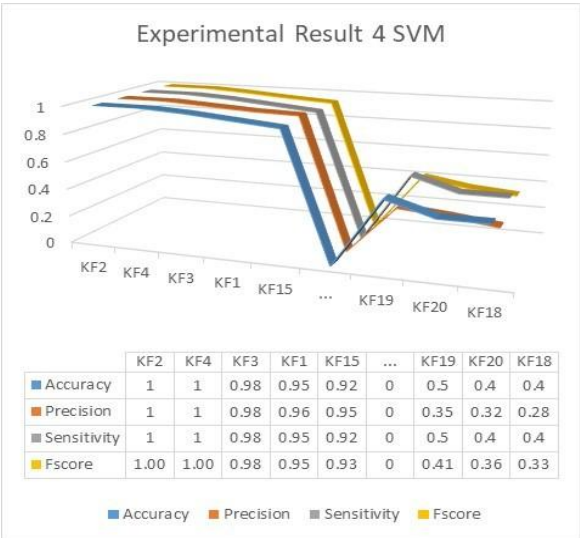
Fig. 4. Best scores of KFs in experiment 2

Figure 4 shows that the F-1 score, which is the harmonic mean of Precision and Sensitivity, is directly proportional to their purity values with exception on KF10, KF13, KF14, and KF15. In Experiment 1, we noticed that the purity scores of a single feature combination in its various test cases remain relatively stable. We assumed that this is caused by the measurement nature of the purity score. Unlike Figure 3, Figures 4–6 display the results of 4 measurements. These Figures present the top 5 and the least 3 feature combinations on the rank of its F-1 scores. It seems that KF3 takes the top position both in clustering and author verification with MKNN and KNN in experiment 2 with a perfect F-1 score -- 1.0, while KF2 takes a lead in author verification with SVM classifier with the same score.



	KF3	KF2	KF4	KF1	KF11	...	KF17	KF19	KF18
Accuracy	1	0.95	0.95	0.92	0.9	0	0.38	0.32	0.28
Precision	1	0.96	0.96	0.94	0.94	0	0.36	0.36	0.19
Sensitivity	1	0.95	0.95	0.92	0.9	0	0.38	0.32	0.28
Fscore	1.00	0.95	0.95	0.93	0.92	0.00	0.37	0.34	0.23

Fig. 5. Best scores of KFs in KNN model



	KF2	KF4	KF3	KF1	KF15	...	KF19	KF20	KF18
Accuracy	1	1	0.98	0.95	0.92	0	0.5	0.4	0.4
Precision	1	1	0.98	0.96	0.95	0	0.35	0.32	0.28
Sensitivity	1	1	0.98	0.95	0.92	0	0.5	0.4	0.4
Fscore	1.00	1.00	0.98	0.95	0.93	0	0.41	0.36	0.33

Fig. 6. KFs' best scores in SVM model

These Figures show also that KF1, KF2, and KF3 are among the top 5 feature combinations. It would be too bias if we concluded that KF3 is the best feature combination for its top position in both experiments 1 and 3. The rationale is that the data saved in Array 2 of each experiment shown in Figures 3, 6 have been unable to answer our research questions. Besides, they show only the best scores for each feature combination and yet cannot describe their robustness in author verification environment.

To solve this problem, a third array was created to trace which feature combinations show their robustness in different test cases. This was done by voting. The first step was to compute the Macro-Average F-1 (MAF) score for each feature combination and K-value both in MKNN and KNN classifiers. The values of MAF scores were saved in the array 3, then the *sort* function was applied to get the highest score for each K parameter of MKNN and KNN. The voting was done by incrementing its variable value when a feature combination (KF) hit the highest score. Due to space limitation, it is impossible for us to present the voting tables of four experimental scenarios here. Therefore, Table 4 is presented here to visualize the values saved in array 3 for voting system. The experiment 2 was chosen here to represent the results of voting system in experiments 1-4, since its results have much in common with experiment 1. Table 5 shows only the first 5 KFs with a reason that the best scores of each K belong to these KFs.

Table 5
The third array in a form of a table for conducting the voting in Experiment 2

K/KF	KF1	KF2	KF3	KF4	KF5	Best KFs for voting	K/KF	KF1	KF2	KF3	KF4	KF5	Best KFs for voting
K1	.77	.79	.77	.77	.61	['KF2']	K11	.74	.73	.78	.72	.41	['KF3']
K3	.83	.76	.83	.76	.58	['KF1', 'KF3']	K13	.66	.73	.78	.72	.38	['KF3']
K5	.77	.75	.79	.74	.52	['KF3']	K15	.61	.74	.76	.74	.41	['KF3']
K7	.86	.71	.76	.75	.43	['KF1']	K17	.61	.67	.72	.66	.39	['KF3']
K9	.78	.73	.76	.72	.39	['KF1']	K19	.66	.67	.69	.67	.35	['KF3']

Note: KF stands for Feature Combinations, while K1 refers to the variable K in MKNN which is equal to 1, etc. The cell values show the macro-average F-1 scores.

The experiment 1 went on as its scenario described earlier. After reducing the dimensionality of array 2 by computing the Macro-Average Accuracy (MAA), Macro-Average Precision (MAP), Macro-Average Sensitivity (MAS) and Macro-Average F-1 score (MAF), we focused our examination on the correlation between MAF and Purity scores. Figure 7 shows the MAFs values of each KF in a bar chart, while the purity scores are described in a line chart. From this Table, we could see that MAF values fluctuate following the fluctuation of the Purity scores of the same feature combinations. After voting was done, it shows that KF3 got the highest voting by 8 scores. The most highest score is achieved by KF3 in K={7, 9, 11} with 0.86 point

In the experiment 2, we closely examined whether KF3, which achieved the highest and stable F-1 scores in the experiment 1, keeps its scores high when the new documents were added to the training data. Figure 8 shows the experimental result in this scenario. This figure shows that the fluctuation of the F1 -scores follows its purity scores. Besides, this Figure shows us that KF3 becomes the best stylometric feature combination. Table 5 shows also that KF3 gets the highest voting by 7 points. However, KF1 achieves the highest F-1 score by 0.86 on K equal to 7 but gets only 3 voting points (cf. Table 5).

The results of Experiments 3 & 4 are presented in Figures 9, 10 respectively. From these Figures, we could also see different classifiers, KNN and SVM, prove that the purity and the F-1 scores corelate proportionally. In both experiments, KF3 still dominates the voting points and proves its robustness. However, it achieves the highest F-1 scores of 0.91 on K equal to 15 in experiment 3. In Experiment

4, KF3 achieves the purity score of 0.83 and F-1 score of 0.88. The interesting result shown by SVM classifier is that some stylometric combinations such as KF14, KF12, KF16, KF13, KF11, and KF15 achieve F-1 scores that are close to KF1, KF2, and KF4 which are always on the top 5 rank, despite of their purity scores which range between 0.62-0.72. SVM prove to be more robust compared to k-NN in this experiment which corresponds to experiments conducted by Sofian *et al.,* [30] for Sentiment Analysis.

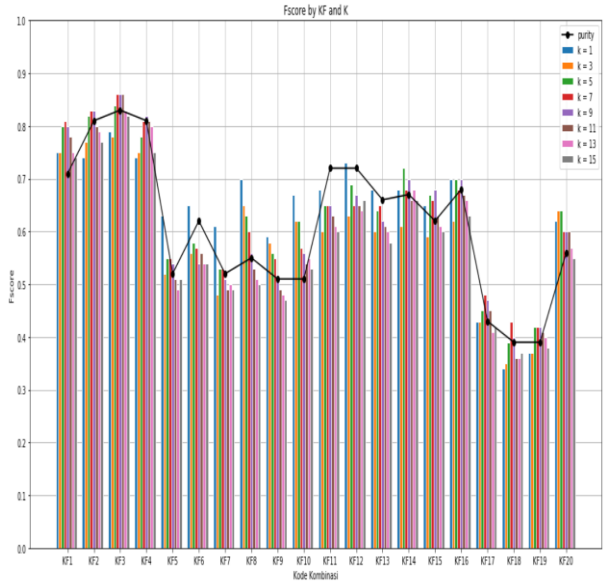


Fig. 7. MAF and Purity scores of experiments 1

Fig. 8. MAF and Purity scores of experiment 2

The experiment 5 evaluated the performance of MKNN classifier, when it was tested with some new authors whose document samples unavailable in the training data. For this reason, this experiment used the best K, which is equal to 7, and the best feature combination (KF3). The number of training data was increased, and the test set was written by 11 authors with 5 texts per author. Only half of the authors in this test set have their sample texts in the training data. Each of 55 test documents was fed to the system consecutively, and the MAA, MAP, MAS and MAF were computed. Thus, we only got a single output for each measurement. Figure 11 shows the result of experiment 5 where MAA achieves 0.76. However, MAP obtains the higher score by 0.83, while MAS achieves the lowest score by 0.76. Thus, the F1-score whose value is always in between MAP and MAS gets 0.79 score. In comparison to experiment 1 and 3 which used the same classifier, the MAF score on the experiment 5 is quite lower. We assumed that this is caused by the absence of some authors' sample texts in the training data. However, we took this result as a real performance of our author verification system built with MKNN

From all these experiments, it can be clearly seen that firstly, the scores of all measures i.e. Macro-Averaged Accuracy, Macro-Averaged Precision, Macro-Averaged Sensitivity, and Macro-Averaged F-1 correlate proportionally to the purity scores. Secondly, the stylometric feature combination achieving purity scores higher than 0.5 will achieve F-1 score greater than 0.5 too. These results lead to our conclusion that will be presented on the following section.

6. Conclusion

In this paper, we have presented the process of experimenting the stylometric feature combinations in a clustering system with K-means before they were applied in an author verification system built with different classifiers, namely MKNN, KNN, and SVM. The stylometric features used in these experiments comprise 3 lexical features, 2 syntactic features and 1 structural feature. Each of these features were combined into a set of features with a length of 3 features. Thus, it resulted

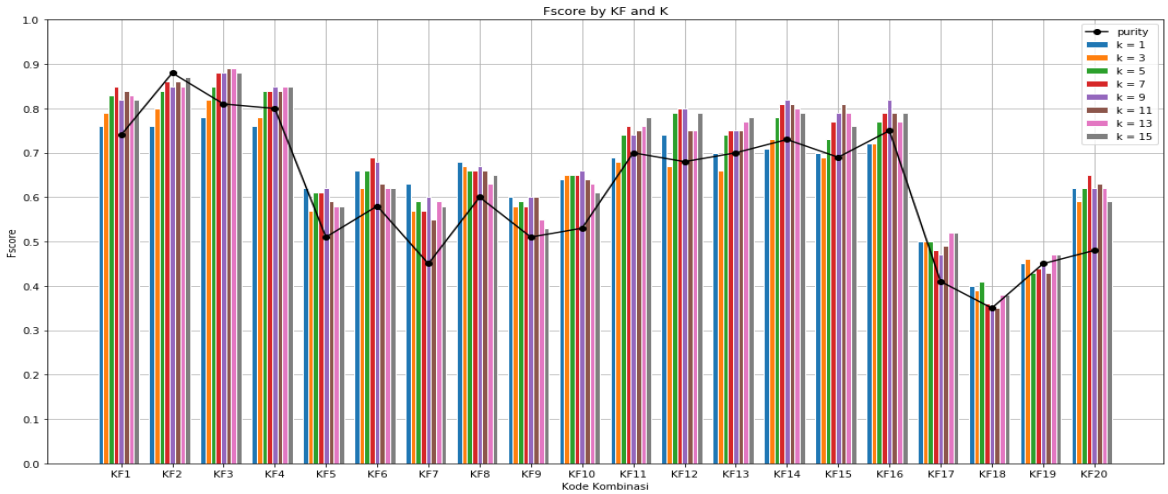


Fig. 9. MAF and Purity scores of experiment 3

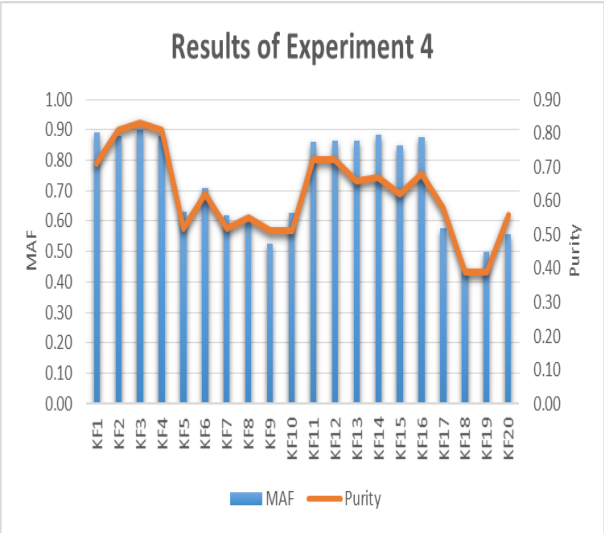


Fig. 10. MAF and Purity scores of experiment 4

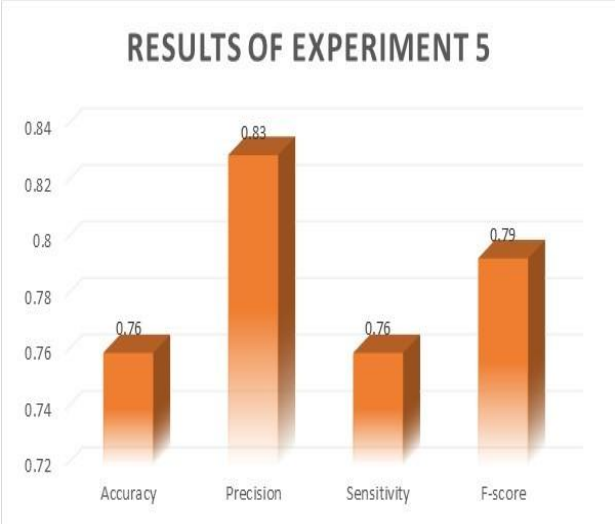


Fig. 11. The scores of MAA, MAP, MAS, and MAF in Experiment 5

in 20 subsets of feature combinations. These feature combinations were experimented in 5 different scenarios.

Examining the results of the 5 experiments described earlier, we came to conclusion that the purity scores in clustering process correlate proportionally to evaluation measures of author verification process, especially to MAF. We found out also that the stylometric feature combinations achieving purity score more than 0.5 achieve MAF scores higher than 0.5. Thus, the purity score in clustering process is reliable enough to discriminate stylometric feature combinations. Besides, any combination containing syntactic features, i.e. the relative frequency of punctuation and stopword

(KF1, KF2, KF3, KF4), proves to be robust as they got high voting points and the top 4 positions on both purity score and MAF. When these features are combined with the structural feature (KF3), it becomes the most robust feature combination with the highest points of voting in all experiment scenarios. Thus, it can be concluded that the combination between syntactic and structural features prove to be the most robust features.

In the future, we would like to explore more structural features combined with the syntactic ones. We perceive also that the lexical features would be a robust stylometric feature if both their form and quantification values are included as features, e.g. the top 50 word n-gram frequency. This suggests that the hybrid of profile-based author verification and instance-based one are worth conducting. Due to the hardware limitation, this study conducted experiments with only 3 features in each combination. It would be interesting to examine the lexical, syntactic and structural features combined into a set of features for an author verification system.

References

[1] Strausz, Erzsébet. "Writing with Foucault: Openings to transformational knowledge practices in and beyond the classroom." *Critical Studies on Security* 10, no. 3 (2022): 144-156. https://doi.org/10.1080/21624887.2022.2134698

[2] Heymann, Laura A. "The birth of the authornym: Authorship, pseudonymity, and trademark law." *Notre Dame L. Rev.* 80 (2004): 1377.

[3] Rouhvand, Hassan. "Author and Authorship: a Barthes-Foucauldian Plural Perspective." *American Academic & Scholarly Research Journal* 8, no. 6 (2016).

[4] Potthast, Martin, Matthias Hagen and Benno Stein. (2016). "Author Obfuscation: Attacking the State of the Art in Authorship Verification." in *Working Notes Papers of the CLEF 2016 Evaluation Labs*.

[5] Gerrard, Y. "What's in a (pseudo)name? Ethical conundrums for the principles of anonymisation in Social Media research." *Qualitative Research* 21, no. 5 (2021): 686–702. https://doi.org/10.1177/146879412092

[6] Nawwaf, Muhammad Nauval, Wini Indriani, Winda Maharani, and Devie Yundianto. "Analysis Of Self Disclosure On Users Of Pseudonym Accounts Which Display Toxic Disinhibition On Twitter Social Media: A Literature Study." In *International Conference Of Humanities And Social Science (ICHSS)*, pp. 402-409. 2022.

[7] Altamimi, A., S. Alotaibi, and A. Alruban. "Surveying the Development of Authorship Identification of Text Messages." *International Journal of Intelligent Computing Research (IJICR).* 10, no. 1 (2019): 953-966. https://doi.org/10.20533/ijicr.2042.4655.2019.0116

[8] Misini, A., and E. Canhasi. "A Survey on Authorship Analysis Tasks and Techniques." *SEEU Review* 17, no. 2 (2022): 153-157. https://doi.org/10.2478/seeur-2022-0100 .

[9] Potha, N., and E. Stamatatos. "Improved algorithms for extrinsic author verification." *Knowledge and Information Systems* 62, no. 1 (2022): 1903—1921. https://doi.org/10.1007/s10115-019-01408-4.

[10] Nini, Andrea, Oren Halvani, Lukas Graner, Valerio Gherardi, and Shunichi Ishihara. "Authorship verification based on the likelihood ratio of grammar models." *arXiv preprint arXiv:2403.08462* (2024).

[11] Manolache, Andrei, Florin Brad, Antonio Barbalau, Radu Tudor Ionescu, and Marius Popescu. "Veridark: A large-scale benchmark for authorship verification on the dark web." *Advances in Neural Information Processing Systems* 35 (2022): 15574-15588.

[12] Boenninghoff, Benedikt, Steffen Hessler, Dorothea Kolossa, and Robert M. Nickel. "Explainable authorship verification in social media via attention-based similarity learning." In *2019 IEEE international conference on big data (Big Data)*, pp. 36-45. IEEE, 2019. https://doi.org/10.48550/arXiv.1910.08144.

[13] Stamatatos, Efstathios. "A survey of modern authorship attribution methods." *Journal of the American Society for information Science and Technology* 60, no. 3 (2009): 538-556. https://doi.org/10.1002/asi.21001

[14] Potha, N., and E. Stamatatos. "Improving Author Verification Based on Topic Modeling." *Journal of the Association for Information Science and Technology,* 70, no. 10 (2019): 1074–1088. https://doi.org/10.1002/asi.24183 .

[15] Ding, Steven HH, Benjamin CM Fung, Farkhund Iqbal, and William K. Cheung. "Learning stylometric representations for authorship analysis." *IEEE transactions on cybernetics* 49, no. 1 (2017): 107-121. https://doi.org/10.1109/TCYB.2017.2766189

[16] Tarmizi, N., S. Saee, and D. H. B. Ibrahim. "Author Identification for under-resourced language Kadazandusun." *Indonesian Journal of Electrical Engineering and Computer Science* 17, no. 1 (2020): 248-255. https://doi.org/10.11591/ijeecs.v17.i1.pp248-255.

[17] Kumar, S., S. Rajeswari, S. Srikant, and T. R. Reddy. "A New Approach for Authorship Verification Using Information Retrieval Features," in *Innovations in Computer Science and Engineering*, Lecture Notes in Network and Systems 74 (2019): 118-145. DOI: 10.1007/978-981-13-7082-3_4

[18] Hu, X., W. Ou, S. Acharya, S. H. H. Ding, R. D'Gama, and H. Yu "TDRLM: Stylometric learning for authorship verification by Topic-Debiasing." *Expert System with Applications,* 233, no. C (. 2023): 120745. https://doi.org/10.1016/j.eswa.2023.120745.

[19] Halvani, Oren, Lukas Graner, and Roey Regev. "A step towards interpretable authorship verification." *arXiv preprint arXiv:2006.12418* (2020).

[20] Jean, Langlois. "When linguistics meets computer science: Stylometry and professional discourse." *Training, Language and Culture* 5, no. 2 (2021): 51-61.. https://doi.org/10.22363/2521-442X-2021-5-2-51-61f .

[21] Gunawan, Silvia P., Lucia D. Krisnawati, and Antonius R. Chrismanto. "Analisis Fitur Stilometri dan Strategi Segmentasi pada Sistem Deteksi Plagiasi Intrinsik Teks." *Jurnal Rekayasa Sistem dan Teknologi Informasi* 4, no. 5 (2019): 988-997. https://doi.org/10.29207/resti.v4i5.2486 .

[22] Hammoud, Khodor, Salima Benbernou, and Mourad Ouziri. "A Sentiment-Based Author Verification Model Against Social Media Fraud." *Atlantis Study in uncertainty Modelling* 3 (2021): 219-226. https://doi.org/10.2991/asum.k.210827.030 .

[23] Huang, Zhao H., Leilei Kong, and Mingjie Huang. "Authorship Verification Based on CoSENT," in *Conference and Labs of the Evaluation Forum*. Thessaloniki, Greece, 2023.

[24] Sarwar, Raheem and Saeed-Ul Hassan. "UrduAI: Writeprints for Urdu Authorship Identification," *ACM Transactions on Asian and Low-Resource Language Information Processing.* 21, no. 2 (2021): 1–18. https://doi.org/10.1145/3476467

[25] Mihaljević, Helena, and Lucia Santamaría. "Disambiguation of author entities in ADS using supervised learning and graph theory methods." *Scientometrics.* 126, no. 5 (2021): 3893–3917. https://doi.org/10.1007/s11192-021-03951-w .

[26] Luyckx, Kim and Walter Daelemans. "uthorship Attribution and Verification with Many Authors and Limited Data." In *Proceedings of the 22nd International Conference on Computational Linguistics* (2008.): 513-520. https://doi.org/10.3115/1599081.1599146

[27] Reviriego, Pedro, Javier Conde, Elena Merino-Gómez, Gonzalo Martínez and Jose H. Hernández. (2023). "Playing with Words: Comparing the Vocabulary and Lexical Richness of ChatGPT and Humans," *arXiv:2308.07462v2 [cs.CL]*. https://doi.org/10.48550/arXiv.2308.07462

[28] Parvin, Hamid, Hoseinali Elizadeh and Behrouz Minaei-Bidgoli. "MKNN: Modified K-Nearest Neighbor," in *Global Journal of Computer Science and Technology* 10, no. 11 (2010): 37-41.

[29] Maulana, Indra and Metta Mariam. "Analysis of Student Activities Based on Log Files in E-Learning Using Clustering Algorithm," in Journal of Advanced Research in Applied Sciences and Engineering Technology 53, no. 1 (2024): 1-15. https://doi.org/10.37934/araset.53.1.115

[30] Sofian, Muhhamad Adam Sani Mohd, Norlina Mohd Sabri, Ummu Fatihah Mohd Bahrin, Hrishvanthika N. and Norulhidayah Isa. "Sentiment Analysis on Acceptance of COVID-19 Vaccine for Children based on Support Vector Machine," in Journal of Advanced Research in Applied Sciences and Engineering Technology, 58, no. 2 (2024): 252-270. https://doi.org/10.37934/araset.58.2.252270