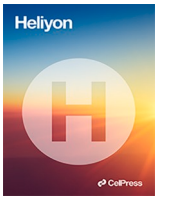
Heliyon 10 (2024) e32976

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

Unveiling ChatGPT text using writing style

Lamia Berriche *, Souad Larabi-Marie-Sainte

College of Computer & Information Sciences, Prince Sultan University, Saudi Arabia

ARTICLE INFO

Keywords:
ChatGPT
Stylometry
Plagiarism
Ensemble learning
Writing style

ABSTRACT

Extensive use of AI-generated texts culminated recently after the advent of large language models. Although the use of AI text generators, such as ChatGPT, is beneficial, it also threatens the academic level as students may resort to it. In this work, we propose a technique leveraging the intrinsic stylometric features of documents to detect ChatGPT-based plagiarism. The stylometric features were normalized and fed to classical classifiers, such as k-Nearest Neighbors, Decision Tree, and Naïve Bayes, as well as ensemble classifiers, such as XGBoost and Stacking. A thorough examination of the classifier was conducted by using Cross-Fold validation, hyperparameters tuning, and multiple training iterations. The results show the efficacy of both classical and ensemble learning classifiers in distinguishing between human and ChatGPT writing styles with a noteworthy performance of XGBoost where 100 % was achieved for accuracy, recall, and precision metrics. Moreover, the proposed XGBoost classifier outperformed the state-of-the-art result on the same dataset and same classifier highlighting the superiority of the proposed feature style extraction method over TF-IDF techniques. The ensemble learning classifiers were also applied to the generated dataset with mixed texts, where paragraphs are written by ChatGPT and humans. The results show that 98 % of the documents were classified correctly as either mixed or human. The last contribution consists in the authorship attribution of the paragraphs of a single document where the accuracy reached 92.3 %.

1. Introduction

Large Language Models (LLM) which are deep neural network models trained on large amounts of data have emerged recently. Even though humans are still better than these models [1], they are used in a variety of applications [2]. For example, they were exploited in tasks like automated content creation [3] such as in Twitter [4], stories generation [5,6], code generation [7,8], reports and academic papers generation [9,10], and text summarization [11]. On the other hand, they could be used inappropriately for plagiarism purposes. Plagiarism refers to using another person's ideas, or words without giving them proper credit and misrepresenting it as one's own. Plagiarism threatens the principles of originality, authenticity, and intellectual property rights. The increased use of content from ChatGPT also raises doubt about the originality of the author's work, especially in the educational domain. Consequently, we emphasize the need to detect plagiarism in texts written by AI generators, specifically ChatGPT. Various techniques such as text similarity analysis, machine learning, and natural language processing can be used to detect plagiarism in ChatGPT's texts.

Plagiarism detection techniques are classified into two main categories: intrinsic and extrinsic [12]. Whereas intrinsic detection

* Corresponding author.
E-mail addresses: lberriche@psu.edu.sa (L. Berriche), slarabi@psu.edu.sa (S. Larabi-Marie-Sainte).

https://doi.org/10.1016/j.heliyon.2024.e32976
Received 6 March 2024; Received in revised form 11 June 2024; Accepted 12 June 2024
Available online 15 June 2024

focuses on analyzing the text itself, extrinsic detection focuses on comparing the text to external sources. Intrinsic detection techniques use stylometric features to identify how the writing style of the writers differs. Intrinsic techniques include text similarity analysis [13], machine learning [14], and natural language processing.

Detecting AI-generated texts still deserves investigation specifically in the education domain. It is vital to recognize the student's homework among student's AI-generated homework. AI-generated texts could affect the student's performance in exams and lead to a lack of competencies.

This research study investigates the recognition of human writing texts among ChatGPT-generated texts. The stylometric technique was applied to detect the authorship 's writing style. The detection of a document's authorship using the feature style extraction method has shown its efficiency in many state-of-the-art studies [15], [16–18].

Recently, Machine and Deep learning using stylometric techniques were investigated for ChatGPT content detection. Some of the existing work focused on transformer-based learning approaches such as [19–21], and others emphasized ensemble learning techniques [22]. Transformer-based learning approaches give high accuracies, but they require enormous datasets. Also, they have weak generalization abilities and exhibit reduced robustness when dealing with noisy data. However, the ensemble learning approaches have proved their effectiveness in terms of result interpretability, noise resistance, and better generalization ability.

This article stresses the comparison between the classical classifiers and the ensemble learning classifiers to detect Human or ChatGPT writing style using the Feature Style extraction method. It also investigates the recognition of a document written only by humans (called simple documents) and a document written by both humans and ChatGPT (called mixed documents). Moreover, the article explores the detection of paragraphs' writing style (Human or ChatGPT) in one document. To our knowledge, this later is not yet examined by the researchers.

The contribution and research objectives of this work are as follows:

1. Detect human or ChatGPT writing style in documents using classical and ensemble learning classifiers
2. Study the accuracy of ensemble learning techniques in detecting ChatGPT text and human text.
3. Show the effectiveness of detecting human/ChatGPT documents using the Feature style extraction method.
4. Recognize mixed (written by humans and ChatGPT) and simple (written by Humans) documents using stylometric technique and ensemble learning classifiers.
5. Propose a new authorship attribution strategy based on paragraph's author detection in mixed documents.
6. Generate new datasets composed of simple and mixed documents.

To perform this study, three datasets were used. Dataset 1 is taken from Ref. [23] which is composed of human and ChatGPT documents. Dataset 2 is a newly generated dataset that contains simple and mixed documents. Dataset 3 is composed of only mixed documents in which paragraphs are written either by humans or ChatGPT. Datasets 2 and 3 can be found in Ref. [24].

The article is organized as follows. Section 2 presents the recent state-of-the-art studies. Section 3 explains the methodology used for the proposed strategies to fulfill the research objectives. Section 4 explores the data collection and preprocessing in addition to all the experiments and the comparison studies performed. Section 5 illustrates the discussion of the results found in the Experimentation section. Finally, Section 6 concludes the present study.

2. Literature review

[23] identified essays written by humans from writings produced by ChatGPT3. They gathered a dataset based on TOEFL essays, incorporating 126 essays authored by humans and 126 essays generated by ChatGPT, all addressing the same topics. They used two types of features: the term frequency-inverse document frequency (TF-IDF) and 244 hand-crafted stylometric features. Both features were used to assess the effectiveness of a classification model built on XGBoost. Their findings demonstrate that they can identify ChatGPT-generated text with an accuracy and F1 of 98 % when TF-IDF is used. When hand-crafted features are used, they reached an accuracy of 96 % and an F1 score of 96 % for the human class and 99 % for the ChatGPT class.

The goal of the study in Ref. [22] was to differentiate between academic scientific writing produced by ChatGPT and writing produced by scientists. They used 64 human articles and generated 128 ChatGPT papers for the training set with in total of 1276 paragraphs. They also used two test sets where each has 30 human papers and 60 derived ChatGPT papers with 1210 paragraphs. The study extracted 20 features within four categories including paragraph complexity, punctuation marks, diversity in sentence length, and popular words or numbers. An XGBoost classifier was used to evaluate how well these features distinguish between text produced by humans and text produced by ChatGPT. They also used the same classifier to distinguish paragraphs written by humans from ones written by ChatGPT. They reached a testing classification accuracy of 92 % at the paragraph level and 100 % accuracy at the document level. Their document-level classification is based on the majority of already classified paragraphs. Their model outperformed the RoBERTa AI deep learning model (85 % testing accuracy). In Ref. [25], the authors tested their AI text detector in the chemistry field. Overcoming the limitation of the previous paper which was only focusing on one scientific journal paper and only one ChatGPT prompt, they covered various Chemistry journals and a variety of prompts. Their accuracy at identifying AI-generated texts ranges from 98 % to 100 % with the XGBoost classifier.

In [26], authors conducted human evaluation, linguistic evaluation, and machine learning and deep learning evaluations to detect ChatGPT texts or sentences. Their human evaluation demonstrated that experts are often better, at differentiating, than amateurs. Additionally, they stated that ChatGPT's responses were considered as being more beneficial than those provided by people except in the medical domain. The linguistic analysis focuses on vocabulary features, part-of-speech (POS) usage, sentiment analysis, and

language model perplexity. They showed that human answers are relatively shorter in length but exhibit a large vocabulary. In addition, ChatGPT generally expresses more neutral sentiments and fewer negative emotions compared to humans. For the classification, they used RoBERTa and a logistic regression model which were trained on the Human ChatGPT Comparison Corpus (HC3) dataset. They found that their performance is affected by the presence of signaling words, with RoBERTa-based detectors being less impacted (99.82 % F1 score in full and 99.79 % when signaling words are filtered) whereas logistic regression gave 98.26 % in full and 98.30 in filtered. The overall results were better in English than in Chinese.

In [27], authors collected a dataset of 509 question-answer pairs about computer science, artificial intelligence, and cybersecurity fields from ChatGPT and humans (dictionaries, encyclopedias). They applied Multinomial Naive Bayes, Random Forest, Support Vector Machines (SVM), and K-nearest neighbors (KNN) models with feature vectors extracted by a natural language processing pipeline. Also, they used the Long Short-Term Memory (LSTM) model as a starting point, DistilBERT, RoBERTa, and a customized model built on RoBERTa. They reported 99.1 % accuracy and 99.2 % F1-score for their customized model.

In [28], the authors performed two studies. The goal of the first study was to pinpoint stylometric variations between Japanese texts created in ChatGPT, GPT-4, and by humans. Their second study uses an RF classifier to distinguish AI from human written texts. Their dataset contains 72 Japanese academic publications and 144 Japanese texts generated using ChatGPT-3.5 and GPT-4. Bigrams of parts of speech, bigrams of postpositional particle words, comma placement, and the frequency of function words are four stylometric aspects that are examined. They showed that texts produced by AI and those written by humans often differ in terms of stylometric traits even with a large parameter model like GPT-4. They reached 100 % accuracy with the Random Forest (RF) classifier when all the features were used. Their study is limited to the Japanese language whose performance is ranked low, compared to English, with ChatGPT [29].

[30] used transformer-based DistilBERT model and SHAP (Shapley Additive Explanations) for explaining the model's decisions. Their dataset consists of human-generated restaurant reviews and two ChatGPT-generated datasets: ChatGPTquery (custom queries) and ChatGPTrephrase (rephrased human reviews). They compared the DistiliBERT model results to a perplexity-based approach. The ML-based approach achieved a high accuracy of 98 %, while the perplexity-based approach accuracy was 84 %. In the paraphrased reviews, the ML-based approach reached 79 % while the perplexity-based approach accuracy is 69 %. Also, their results from SHAP showed that ChatGPT uses more impersonal language, less personal pronouns and expressions of feelings, and often relies on a mixture of third-person and passive speech.

In [31], authors aimed to distinguish between human and AI-generated tweets and identify the point in a timeline where an author change from human to AI occurs, respectively. They used three categories of stylometric features: phraseology features, punctuation features, and linguistic diversity features. They used publicly available datasets on specific topics (anti-vaccine, climate change) and collected tweets on COVID-19 using the Twitter API as sources of human-authored tweets. For AI-authored tweets, they generated tweets using different language models such as gpt2, gpt2-medium, gpt2-large, and EleutherAI-gpt-neo-1.3B and fine-tuned these models on human-authored tweets. They reached 99 % accuracy with ROBERT-a and 95.8 % with XGBOOST on 20 timeline tweets on their in-house dataset and 91.1 % and 84.7 % on the TweepFake dataset. Their Stylometric Change Point Agreement (StyloCPA) style change detection technique reached an accuracy of 89.2 %.

In [32], a dataset of a total of 4400 samples was collected, with 2200 each in the medical abstract and radiology report datasets. The authors used four methods to detect ChatGPT-generated reports. In the perplexity-CLS method, they used text perplexity as a threshold to distinguish between human and ChatGPT-generated medical text. They applied CART (Classification and Regression Trees), XGBoost ensemble learning method with TF-IDF vectorizing samples using TF-IDF, and a fine-tuned BERT. The best performance was achieved by the BERT model, with an F1 score exceeding 95 % for both medical abstract and radiology report datasets in contrast to XGBoost reaching 89 % F1 score.

[33] introduced the Tunicate Swarm Algorithm with a Long Short-Term Memory Recurrent Neural Network (TSA-LSTMRNN) model to address this problem. The TSA-LSTMRNN technique uses TF-IDF with word embedding and counts vectorizers for feature extraction. The detection and classification processes utilize the LSTMRNN model with the TSA for parameter optimization. They conducted their simulations on the human and ChatGPT reviews dataset proposed by Ref. [34]. They achieved an accuracy of 93.17 %

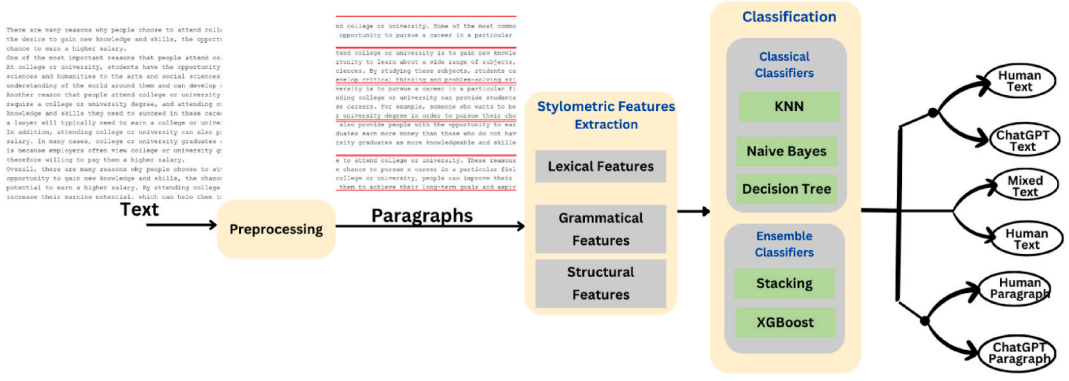


Fig. 1. Proposed technique.

for human-generated text and 93.83 % for ChatGPT-generated text.

3. Proposed strategy

In this section, we discuss the proposed technique which consists of three main phases: preprocessing, feature extraction, and classification, Fig. 1.

3.1. Preprocessing

Before feature extraction, we run a sequence of pre-processing steps. We divide each text into paragraphs. Then, we applied word tokenization followed by a removal of stop words. In addition, we applied a Part of Speech (POS) tagging, using NLTK's Perceptron Tagger, for syntactic feature extraction.

3.1.1. Feature extraction

In our work, we extracted lexical features, syntactic features, and structural features. These features were used in a variety of natural language processing applications such as text analysis [35], authorship attribution [36], and security [37]. Lexical features are statistical measures of word-based and character-based lexical variations in the text. We considered the following **12** lexical features. The selected lexical features are presented in Table 1. Lexical features, such as word frequencies, word lengths, and vocabulary richness capture the unique writing style of an author. Metrics such as the average number of words and average number of words after removal of stop words were used to measure the lexical variation which reflects the vocabulary size and demonstrated its importance in distinguishing between ChatGPT and human texts in Refs. [38,39].

In addition to the lexical features, we extracted **17** syntactic features. They capture the structure and grammatical patterns of an author's writing style. Syntactic features include the frequency of specific grammatical constructs, such as sentence length, and part-of-speech (POS) tags. Grammatical features include the use of various POS such as nouns, pronouns, adjectives, adverbs, and participles. Structural features refer to aspects of the text related to the organization, layout, and overall structure such as sentence length, and number of sentences. The selected syntactic features are presented in Table 1.

The listed features have different scales; some features are large numbers such as the number of lower-case letters and others are small numbers such as the average number of distinct words. Consequently, we applied feature normalization which consists of the adjustment of feature scales to a uniform range. Examples of feature normalization techniques are min-max scaling, vector normalization, and Z-score normalization. We used unit vector scaling for feature normalization [40,41].

a) Classification

In the classification phase, we utilized both classical classifiers and ensemble techniques. Classical techniques like k-Nearest Neighbors (k-NN), Naive Bayes, and Decision Tree served as a fundamental benchmark for our comparison. In k-NN, a data point is assigned to the class of its k-nearest neighbors based on the similarity of instances in the feature space. 'k', representing the number of nearest neighbors is the parameter that can impact the performance of k-NN classifier. Naive Bayes is a probabilistic classification

Table 1
Stylometric features.

Lexical Feature	
1-Total number of words = N	7-Average word length = C/N
2-Total number of distinct words = D	8-Frequency of upper-case letters = Number of Upper case/26
3-Average number of words = D/N	9-Frequency of lower-case letters = Number of Lower case/26
4-Total number of distinct words after removing stop words = Ds	10-Frequency of special characters = Number of special characters/
5-Average number of words after removal of stop words = Ds/N	11-Frequency of punctuation = Number of punctuations/9
6-Total number of characters = C	12-Frequency of numeric
Syntactic Grammatical Features	
1-Frequency of stop words	10- Frequency of modal auxiliary
2- Number of proper nouns	11- Frequency of conjunctions
3- Frequency of common nouns	12- Frequency of genitive case
4- Frequency of pronouns	13- Number of existential "There"
5- Frequency of adverbs	14- Number of existential "to"
6- Frequency of ordinal and cardinal adjectives	15- Frequency of past verbs
7- Frequency of determiner	16- Frequency of past participles
8- frequency of wh-words	17- Number of present participles
9- Frequency of preposition	18- Number of quotations
	19- Number of commas
Syntactic Structural features	
1- Total number of sentences	2- Character 3-g
3- Average number of words per sentence	

technique based on Bayes' theorem. It computes the likelihood of a class association to an instance, assuming independence among features. It is used in classification problems with a large number of features. Decision trees generate decision rules derived from relevant features. One main advantage of this technique is that it provides an easy-to-understand representation of decision rules. To reach a high performance, hyperparameter tuning is performed for each classifier.

Ensemble learning is a machine learning approach that combines the predictions of multiple models to improve the overall performance of the classification system. Ensemble methods include Bagging, Boosting, Majority Voting, Average Voting, and Stacking. In this work, we selected Stacking and XGBoost as ensemble techniques. Stacking relies on combining the learning results of base models used at the first level which are used as inputs for a meta-learner model. In our work, we leveraged classification algorithms in the first level namely (Logistic Regression (LR), K-Nearest Neighbor (kNN), Decision Tree (CART), Support Vector Machine (SVM), and Gaussian Naïve Bayes (Bayes)). We used Logistic Regression (LR) as a meta-learner. Logistic regression is a linear model that can be easily trained and optimized, making it an appropriate choice for combining the predictions of multiple base learners [42,43]. Our stacking model is presented in Fig. 2.

XGBoost (or eXtreme Gradient Boosting) is based on combining the gradient and decision trees techniques. It consists of creating a predictive model iteratively using several decision tree models as displayed in Fig. 3.

b) Performance Evaluation

We will compare the different techniques using the following performance criteria: accuracy, precision, recall, and F1.
Accuracy: It measures the proportion of correctly classified instances among all instances, see Eq. (1).

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

Recall (Sensitivity): It measures the proportion of correctly predicted positive instances among all actual positive instances, see Eq. (2).

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}} \tag{2}$$

Precision: It measures the proportion of correctly predicted positive instances among all instances predicted as positive, see Eq. (3).

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FP}} \tag{3}$$

F1 Score: It is the harmonic mean of precision and recall, providing a balance between the two metrics, see Eq. (4).

$$\boldsymbol{F1} = \frac{\boldsymbol{2} * \boldsymbol{Precision} * \boldsymbol{Recall}}{\boldsymbol{Precision} + \boldsymbol{Recall}} \tag{4}$$

4. Experimental results

First, the data collection and preprocessing are explained. Then, four experiments are performed to achieve and demonstrate the research objectives. Moreover, two comparison studies are presented to show the best classical classifier, among the k Nearest Neighbor, The Gaussian Naïve Bayes, and the Decision Tree, in addition to the comparison with the state-of-the-art studies. All experiments were carried out in a standard Colab environment. We used Python libraries such as NLTK, Scikit-learn, Pandas, and NumPy.

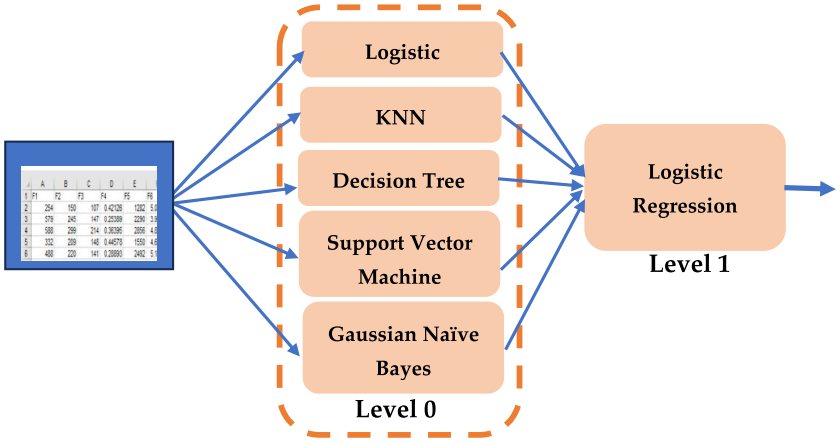


Fig. 2. Stacking ensemble classifier architecture.

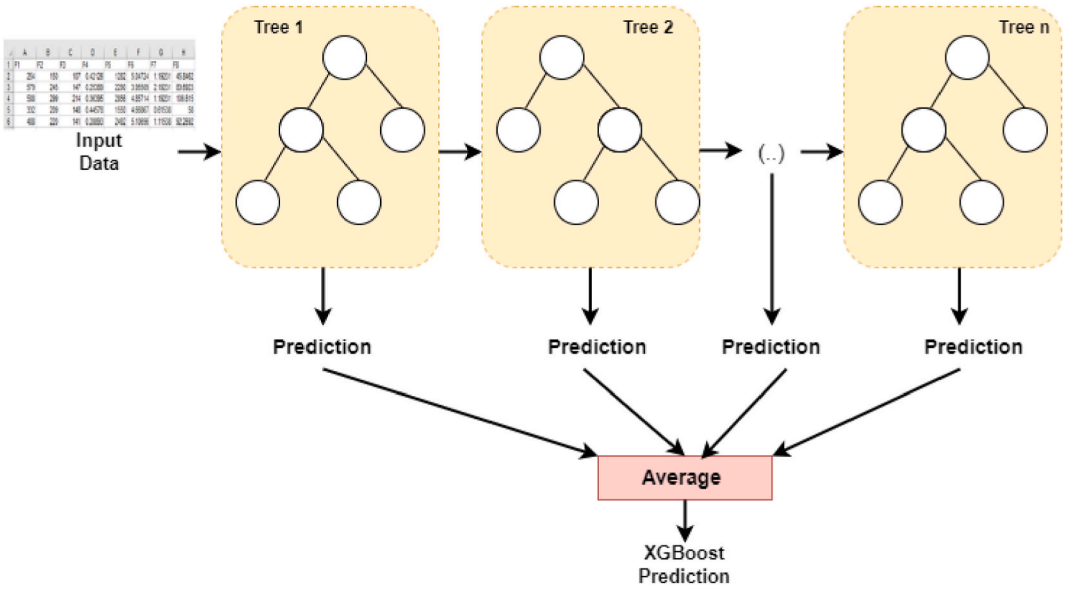


Fig. 3. XGBoost Classifier architecture.

4.1. Data collection

In this study, three datasets are used. The datasets are composed of documents written by humans and\or ChatGPT.

4.1.1. First dataset

The first dataset, collected from Ref. [23], is composed of training and testing sets. The training set contains two labeled sets, ChatGPT and Human, each one composed of 125 and 126 text files respectively. Table 2 displays more details about this dataset. The testing set has 46 text files, but they are not labeled. This dataset is used in Experiments 1 and 2 to classify the documents into Human or ChatGPT classes.

4.1.2. Second dataset

The second dataset used in this article was collected manually. It is composed of 100 text documents. Every document is composed of different paragraphs written by humans and ChatGPT. This document is called a "mixed document". Fig. 4 presents the structure of a mixed document and its different paragraphs. In this figure, the numbers represent the author of the paragraph, 0 for a paragraph written by a human and 1 for a paragraph written by ChatGPT. Only 81 documents were labeled. So, the 19 not-labeled documents were used in the testing stage.

To perform Experiment 3 which aimed at recognizing mixed documents among simple documents (written only by Humans), this dataset is arranged as follows. The 126 documents written by humans in Dataset 1 are taken and added to the 81 mixed documents in Dataset 2. Hence, two classes were formed as displayed in Table 3.

4.1.3. Third dataset

Dataset 3 was generated from dataset 2 to demonstrate the recognition of the writing style, human or ChatGPT, in paragraphs as explained in Experiment 4. Hence, dataset 3 is composed of 81 mixed and labeled documents from Dataset 2 in addition to 19 generated mixed and unlabeled documents as displayed in Table 4.

4.2. Data preprocessing

In our study, we conducted an automated feature extraction process using Python scripts executed within the Google Colaboratory (or Colab) environment. We used the NLTK library for the lexical features extraction. In addition, we used POS tagger to quantify the syntactic attributes. The feature style process was applied to each document of both datasets (dataset1 and dataset2, but for dataset1

Table 2
Details of dataset 1.

Training set	# of files	Class Label
ChatGPT	125	1
Human	126	0

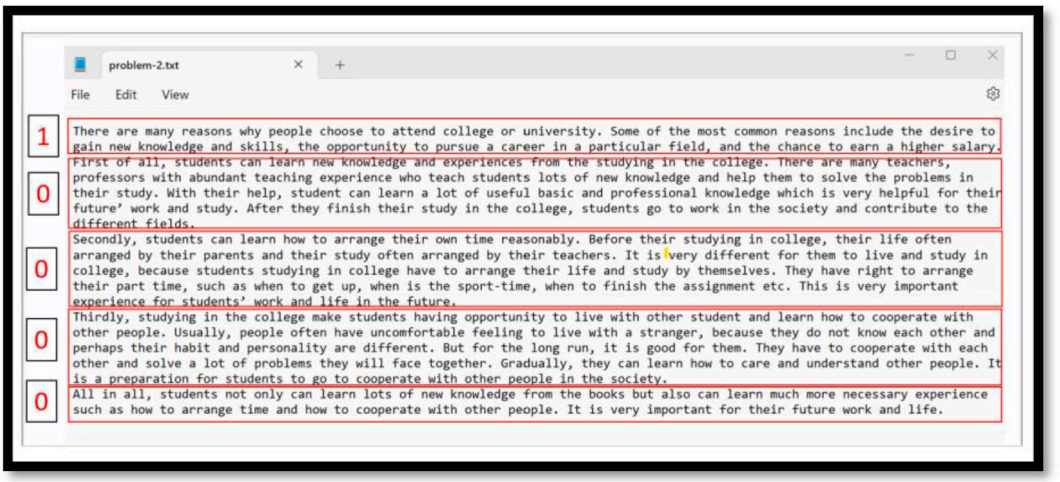


Fig. 4. A presentation of a mixed document containing paragraphs written by humans (label 0) and ChatGPT (label 1) in the collected dataset (Our_FS dataset).

Table 3
Details of dataset 2.

Training set	# of files	Class Label
Mixed Documents	81	1
Human Documents	126	0

Table 4
Details of dataset 3.

Dataset3	# of files	# of Paragraphs
Training Set	81	391
Testing Set	1	105

only the training set is used). The result of each document is a Feature_Style, of length 33, of real values representing the feature styles. Fig. 5 displays 6 Feature_Style vectors associated with 6 documents from Dataset 1, the columns represent the features, and the rows are the documents.

For dataset 3, we extracted the feature style vector of each paragraph of the mixed documents. Hence, the Feature_Style vector (called Our_FS_Parag_Tr and Our_FS_Parag_Ts) indicates the feature values of one paragraph. For instance, if a document has 5 paragraphs, the results will be 5 Feature_Style vectors.

As explained in the Methodology section, some features are based on calculating the frequency (for example the frequency of special characters, frequency of proper nouns, etc.) and others are based on counting (for example the total number of words, the total number of characters, etc.). Therefore, the values in the Feature_Style vector are of varied ranges (see for example F1, F4, F14, and F33 in Fig. 5) which might affect the classification process. To overcome this issue, we normalized the features of dataset 1 by rescaling the vector values to the range between 0 and 1.

To show the efficiency of the normalization process using dataset1, the preprocessed dataset1 with (called FS_Norm dataset) and without (called FS) normalization are investigated. The results of both types were compared and discussed in Section 4.3.4.

4.3. Experiment 1: detection of writing style in documents written either by humans or ChatGPT using the classical classifiers

After performing data preparation and feature extraction, we used kNN, NB, and DT to detect the writing style (ChatGPT or Human) using both FS_Norm and FS datasets.

Each dataset is divided into 80 % training and 20 % testing and ten-fold cross-validation is applied to the training set (10-CV). Seeking high performance, we performed hyperparameter tuning for each classifier and the experiments were repeated 50 times, for each classifier.

4.3.1. K nearest neighbor

Based on the recommendation of [44], the 'k' parameter of the KNN is set between 1 and 25. After hyperparameter tuning, the best

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG
1	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30	F31	F32	F33
2	254	150	107	0.42126	1282	5.04724	1.19231	45.8462	0.056	6.33333	0	0.7095	7	0.00984	0.22222	0.005	0.0125	1.25581	0	0.21849	0.64	0.31818	4	1	8	0.01531	0.0297	0	9	17	16	15.875	1533
3	579	245	147	0.25389	2290	3.95509	2.19231	83.6923	0.04	6.22222	0	1.84916	6	0.01837	1	0.0375	0.1	2.32558	0.75	0.71429	1.44	1.07576	0	1	29	0.11386	0.05941	2	0	30	20	28.95	2866
4	588	299	214	0.36395	2856	4.85714	1.19231	106.615	0.064	5.55556	0	1.65922	13	0.0238	0.34722	0.0375	0.025	3.13953	0.25	0.63866	1.04	1.12121	3	0	22	0.08134	0.11765	0	0.5	19	24	24.5	3441
5	332	209	148	0.44578	1550	4.66867	0.61538	58	0.016	2.88889	0	0.83799	5	0.01264	0.20139	0.0225	0.025	1.39535	0.125	0.48739	0.4	0.4697	3	1	16	0.02525	0.0202	1	1	5	14	23.7143	1879
6	488	220	141	0.28893	2492	5.10656	1.11538	92.2692	0.032	7.11111	0	1.28492	10	0.01837	0.40278	0.0825	0	2	0.5	0.43697	2.28	0.89394	1	0	17	0.09091	0.0101	1	0	30	29	16.8276	2977

Fig. 5. Representation of Feature_Style vector.

model is generated with an accuracy of 0.905 (and 0.91 respectively) when k = 2 (resp. k = 4) using the FS (respectively FS_Norm) for the preprocessed dataset 1. This obtained model is then used for the testing set.

The scores of the 4-performance metrics for both types of preprocessed dataset1 using the testing set are presented in Table 5. For each metric, the best result of the 50 iterations is recorded in addition to the mean and the standard deviation.

It is noticed, for both types of preprocessed dataset1, that the accuracy and the F1 measures reached the highest value of 0.98 with an average of 0.87 and a standard deviation between 0.04 and 0.06. The recall reached 1 for both datasets with an average of 0.91 and a standard deviation around 0.05. The precision reached 1 for the FS dataset1 and 0.97 for the FS_Norm dataset1, with a mean of around 0.83 and a standard deviation achieving 0.07.

We conclude that kNN yields promising results in differentiating between ChatGPT and Human writing style using the Feature style concept. Moreover, the classification results of kNN are not affected by the normalization.

4.3.2. Gaussian Naïve Bayes

The Gaussian Naïve Bayes (NB) requires tuning the smoothing variable. The aim is to make the model stable by widening the curve. This is done by finding samples that are apart from the distribution mean. Based on the recommendations of [45], the logspace function in Python is used to yield values set apart on a log scale, starting from 0 to −9, producing 100 samples.

As previously done with kNN, the best model is found based on the largest accuracy. This obtained is used in the testing stage and the experiments with the same model were iterated 50 times using both datasets to demonstrate the consistency of the obtained results. The results of the testing set are displayed in Table 5.

The best accuracy value achieved is 0.96 using the FS_Norm dataset1 with an average equal to 0.86 and a standard deviation of 0.05. The Precision and Recall reached 1 with a standard deviation of 0.07, whereas F1 measure achieved 0.96 with a mean 0.86 and a standard deviation of 0.05.

The results obtained using the FS_Norm dataset1 outperformed the results yielded with FS dataset1. The normalization contributes to the improvement of the classifier results by 8 % for the accuracy results and 6 % for the precision and F1 measures. The recall reached 100 % for both types of the preprocessed dataset1.

4.3.3. Decision tree

DT has many parameters. However, the depth parameter was selected for optimization as large depth induces an overfitting and a small depth yields an underfitting.

The depth parameter is set between 0 and 7 based on the recommendations highlighted in Ref. [46]. After iterating the experiment 50 times, the best classification model is found with the highest training accuracy reaching 0.96 (resp. 0.90) for the FS (resp. the FS_Norm) dataset1 when the depth is equal to 5 (resp. 3).

This best model is used in the testing phase, and the results are shown in Table 5. As shown, the results obtained with the FS dataset1 are better than those yielded when using the FS_Norm dataset1. The difference is up to 4 % for the accuracy and precision, and 1 % for the F1 measure. The recall reached 100 % for both types of the preprocessed dataset1.

4.3.4. Comparison results

The results of the three classifiers are promising and comparable. Fig. 6 displays the best accuracy value for each technique using both types of the preprocessed dataset1. kNN outperforms the other classifiers using with and without normalization on dataset 1. The worst accuracy value is yielded by NB classifier when using a non-normalized feature vector. Fig. 6 also indicates that kNN is not affected by the data normalization. Contrary to kNN, the NB depicts an increase of 8 % when feature normalization is applied. On the other hand, the normalization slightly reduces the accuracy of DT Classifier by 2 %.

Even though the selected classifiers are known for their efficacy, the hyperparameter tuning applied to each classifier contributes to the enhancement of the results. Moreover, this demonstrates that our results are consistent. Figs. 7 and 9 display the best, the mean, and the standard deviation of the four metrics for the three classifiers using FS_Norm and FS respectively. It is noticed that, for all the classifiers, the highest value is not far from the mean for all the metrics using both types of the preprocessed dataset1. The difference between the highest value and the mean ranges between 0.09 and 0.22 for all the metrics of the three classifiers. This remark is proved by the standard deviation to which the values are very small and range between 0 and 0.1.

Fig. 8 presents the best values of the four classification metrics for the three classifiers using both types of preprocessed dataset1.

Table 5
Classification results of the three classifiers (kNN, NB, DT) using Feature Style Normalized (FS_Norm) and not normalized (FS) of the preprocessed dataset1.

		Accuracy			Precision			Recall			F1		
		Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std
kNN	FS_Norm	0.9804	0.87	0.04	0.97	0.83	0.07	1.0	0.92	0.05	0.98	0.870	0.04
	FS	0.9803	0.87	0.05	1.0	0.84	0.07	1.0	0.91	0.07	0.98	0.87	0.06
NB	FS_Norm	0.9608	0.8620	0.05	1.0	0.8450	0.07	1.0	0.8893	0.07	0.9630	0.8640	0.05
	FS	0.8823	0.7502	0	0.9310	0.7022	0.07	1.0	0.8679	0.07	0.9	0.7726	0.05
DT	FS_Norm	0.9412	0.8106	0.05	1.0	0.810	0.08	1.0	0.8035	0.1	0.9524	0.8069	0.06
	FS	0.9608	0.82	0.06	0.9565	0.8109	0.07	1.0	0.8308	0.1	0.9667	0.8173	0.07

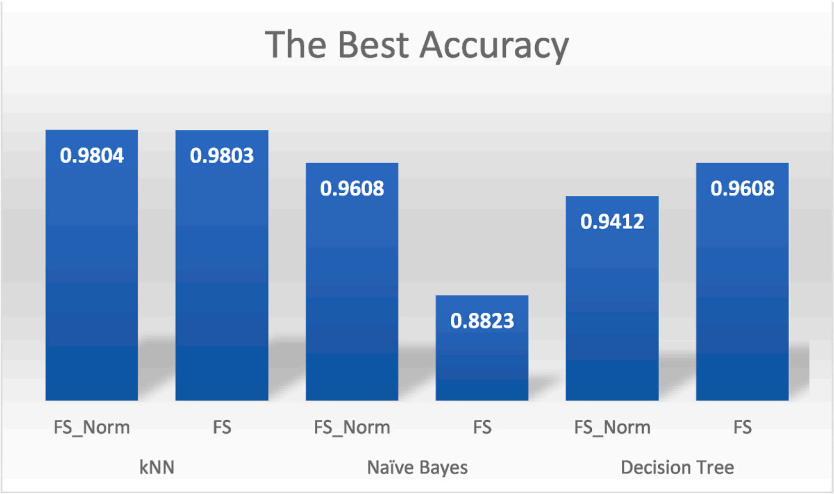


Fig. 6. The best accuracy values were obtained using the three classifiers with FS_Norm and FS dataset1.

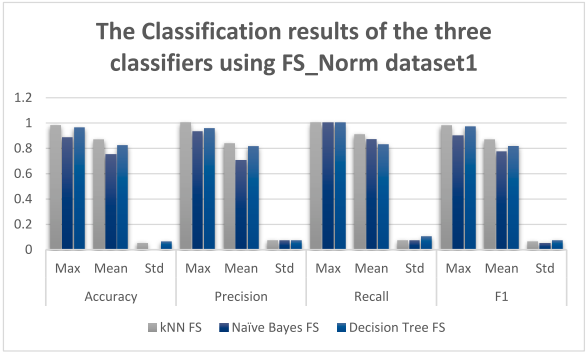


Fig. 7. The Classification results of the three classifiers using FS_Norm dataset1.

The Precision value reached 1 for the three classifiers with either FS or FS_Norm dataset1, while the lowest value is 0.93. The true positive rate (the recall) is 1 for all the classifiers with both types of the preprocessed dataset1. This result demonstrates that the proposed models have a high quality of positive prediction. Lastly, the F1 score ranges between 0.98 and 0.9. Hence, this result shows the reliability of the three models. To conclude, the classifiers efficiently detect the difference in the writing styles between the ChatGPT and Human.

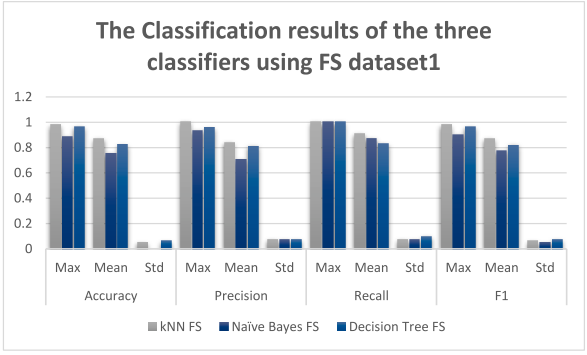


Fig. 8. The Classification results of the three classifiers using FS dataset1.

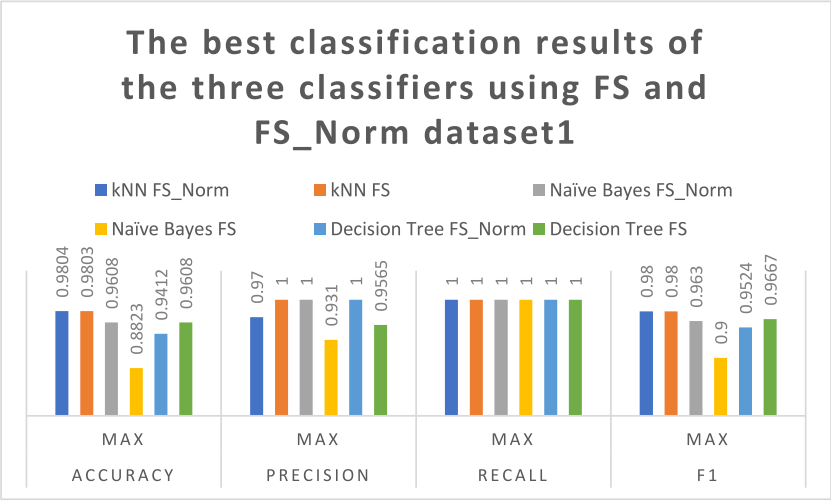


Fig. 9. The best classification results obtained using the three classifiers with both datasets.

4.4. Experiment 2: detection of writing style in documents written either by humans or ChatGPT using the ensemble learning classifiers

In this experiment, two ensemble learning classification algorithms were used, including the XGBoost and Stacking using both types of the preprocessed dataset1 utilized in experiment 1.

Both types of the preprocessed dataset1 are split using 10-fold validation. For the XGBoost, the experiment is initially repeated 50 times. However, the obtained results did not show consistency, especially when using the FS dataset1. Hence, the number of iterations was increased to 100 which leads to consistent results.

Table 7 shows the values of the four metrics after running the classification algorithm 100 times using the FS and FS_Norm dataset1. The results reached 100 % with an average exceeding 89 % and the standard deviation varying between 4 % and 7 % for all the metrics using FS and FS_Norm dataset1. Consequently, the normalization does not affect the XGBoost performance.

For the stacking classification algorithm, the first level (Level 0) of this technique involves five classification algorithms (Logistic Regression (LR), K-Neirest Neighbor (kNN), Decision Tree (CART), Support Vector Machine (SVM), and Gaussian Naïve Bayes (Bayes)) while the second level (Level 1) consists of applying Logistic Regression (LR). Table 6 displays the Best, Mean, and Standard deviation of the accuracy value for all the algorithms using both types of the preprocessed dataset1 after performing 50 iterations (note that the experiment showed constant results with only 50 iterations, the result could not be enhanced even though the number of iterations increases). The best result for each algorithm exceeded 92 % for Fs and FS_Norm dataset1 except for SVM where the best accuracy is 76 % for the FS dataset1. The standard deviation is very low which confirms that the accuracy values obtained throughout the 50 iterations are not very different from each other. The final result of this technique achieved an accuracy of 100 % for both types of the preprocessed dataset1. Again, the normalization has no effect on the results except for the SVM classifier where the normalization improved the classification accuracy by 20 %. Fig. 10 shows the accuracy of level 0 algorithms. For the FS dataset1 (Fig. 10-b), the best result is obtained by LR, kNN, CART, and Stacking (100 %), then Bayes reached 96 % and the worst accuracy is yielded by SVM at 76 %. For the FS_Norm dataset1 (Fig. 10-a), the best accuracy is achieved by kNN, CART, Bayes, and Stacking, then SVM reaches the second-best result (96 %), and the least accuracy is obtained by LR (92 %). To conclude, one can say that normalization affects positively SVM and negatively LR. While the other classification techniques are not sensitive.

Table 6 presents the four metrics results for both classifiers using FS and FS_Norm dataset1. All the metrics achieved 100 % with a representative mean and very low standard deviation. The ensemble learning classifiers demonstrate their effectiveness in differentiating between Human and ChatGPT writing styles.

4.5. Comparison results with existing work

In this section, the five classifiers previously used were compared with the state-of-the-art study [23]. Only one study was used in the comparison because it's the unique existing work that utilized dataset 1. As described in the Related work section [23], used two types of features, (TF-IDF) and hand-crafted features. The best result was found with the TF-IDF technique. Table 8 displays the accuracy and the results of this existing work and our work. The existing work [23] outperforms NB and DT [23] as the XGBoost is more powerful than the classical classifiers. Moreover, kNN demonstrated its effectiveness in achieving the results provided in Ref. [23]. On the other hand, our proposed Ensemble learning classifiers (XGBoost and Stacking) outperformed the XGBoost proposed in Ref. [23]. The reason behind this achievement is the utilization of the feature style and the optimization of the hyperparameters in kNN. This study demonstrates the performance of the feature style against the TF-IDF. In addition, the optimization of the hyperparameters of the classifiers increases the classifiers' accuracy.

Table 6
The Classification accuracy (Maximum value, Mean, and Standard deviation (Std)) of the Stacking algorithm along with its involved classifiers using FS and FS_Norm dataset1.

	Level 0															Level 1		
	LR			CNN			CART			SVM			Bayes			Stacking		
	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std
FS_Norm	0.923	0.684	0.133	1.000	0.860	0.066	1.000	0.809	0.074	0.960	0.637	0.162	1.000	0.845	0.070	1.0	0.866	0.067
FS	1.000	0.916	0.055	1.000	0.792	0.080	1.000	0.814	0.073	0.760	0.564	0.062	0.960	0.744	0.081	1.0	0.916	0.055

Table 7
Classification results of the Ensemble Learning classifiers (XGBoost & Stacking) using Feature Style Normalized (FS_Norm) and not normalized (FS) dataset1.

		Accuracy			Precision			Recall			F1		
		Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std
XGBoost	FS_Norm	1.0	0.8978	0.04	1.0	0.8995	0.06	1.0	0.8963	0.06	1.0	0.8957	0.04
	FS	1.0	0.8990	0.04	1.0	0.9037	0.06	1.0	0.8971	0.07	1.0	0.8979	0.04
Stacking	FS_Norm	1.0	0.866	0.067	1.0	0.9375	0.05	1.0	0.9152	0.05	1.0	0.9246	0.03
	FS	1.0	0.916	0.054	1.0	0.9117	0.06	1.0	0.9122	0.06	1.0	0.9101	0.04

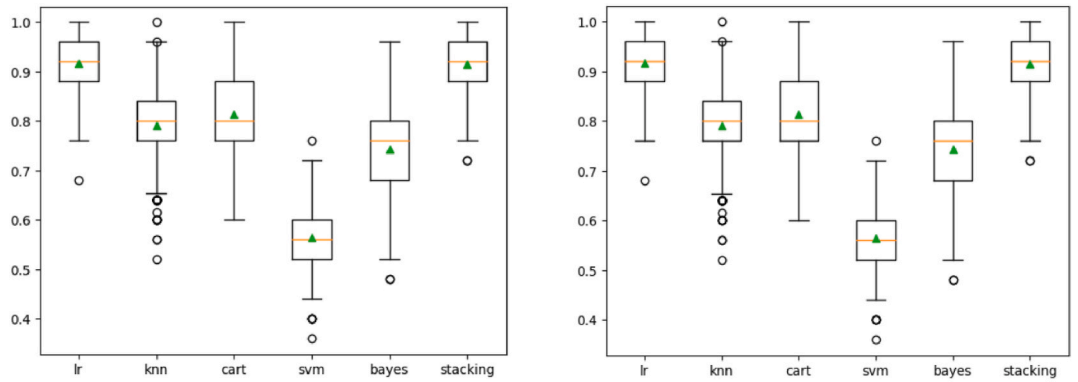


Fig. 10. Scatter plot displaying the accuracy of the classification algorithms involved in the stacking ensemble learning using the FS_Norm (b) and the FS (a) dataset1.

Table 8
Comparison of the Classification results provided by our proposals and the existing work [23] using dataset1.

	Accuracy		F1	
	Max	Mean	Max	Mean
XGBoost (Shijaku & Canhasi, 2023)	0.98	–	0.98	–
kNN	0.98	0.87	0.98	0.87
NB	0.96	0.88	0.96	0.86
DT	0.96	0.82	0.97	0.81
XGBoost	1.0	0.8990	1.0	0.8979
Stacking	1.0	0.916	1.0	0.9101

4.6. Experiment 3: detection of writing style in documents written together by humans and ChatGPT using the ensemble learning classifiers

In this experiment, we used the generated dataset 2. The goal of this experiment is to classify the document into either a Mixed document (Class 1) or a simple document (written by a Human only, Class 0). The feature style process (explained previously) is applied to this dataset (called Our_FS). Then, XGBoost and Stacked techniques are utilized for classification purposes. Normalization was not considered in this experiment since its contribution to enhancing the classification results in the previous experiments was not effective. The experiment is repeated 100 times for each classifier to ensure that the obtained results are consistent and cannot be enhanced.

Table 9 and Fig. 11 present the classification accuracy of the Stacking algorithm. The maximum accuracy values of all the techniques range between 0.6 and 1, the mean ranges between 0.6 and 0.87, and the standard deviation is between 0.07 and 0.1. The result achieved in the second level is about 1, with a mean of 0.87 and a standard deviation of 0.07. Moreover, the highest values of the precision, the recall and the F1 measures 1, with a mean between 0.85 and 0.89 for the four measures and a very small value of standard deviation (between 0.07 and 0.09). The obtained result signifies that the proposed algorithm successfully recognizes both classes “mixed” and “Simple”, which means that the writing style of a mixed document can be recognized from the writing style of a human.

The XGBoost classifier also effectively recognizes both classes (see Table 10) with an accuracy of 0.98, precision and recall equal to 1, and an F1 measure of 0.97. The difference between the highest value and the mean of each metric varies between 0.1 and 0.14. This means that all the values obtained throughout the 100 iterations are close to each other. This is also justified by the standard deviation

Table 9
The Classification accuracy (Maximum value, Mean, and Standard deviation (Std)) of the Stacking algorithm along with its involved classifiers using Our_FS dataset2.

	Level 0															Level 1		
	LR			kNN			CART			SVM			Bayes			Stacking		
	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std
Our_FS	1.0	0.769	0.1	0.87	0.607	0.1	1.0	0.864	0.07	0.591	0.559	0.02	0.957	0.748	0.09	1.0	0.869	0.07

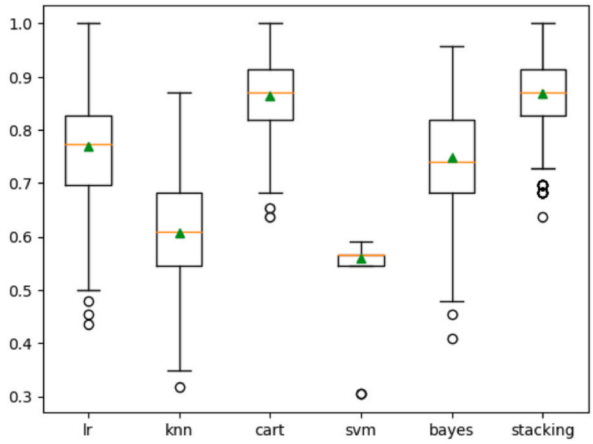


Fig. 11. Scatter plot displaying the accuracy of the classification algorithms involved in the stacking ensemble learning using the Our_FS dataset2.

which did not exceed 0.07.

Both Classifiers achieved promising results, but XGBoost is slightly better than Stacking when considering the mean values of the metrics. Consequently, documents written by ChatGPT and Human can be differentiated by the proposed strategy.

4.7. Experiment 4: detection of the writing style of paragraphs in a document written together by a Human and ChatGPT

This experiment aims to detect the writing style of the different paragraphs in one document. For this purpose, we used dataset 3.

To detect both classes (0: Human and 1: ChatGPT) for each paragraph in the training dataset3, 10-fold cross-validation was applied to the XGBoost classifier. Table 11 displays the classification results of this experiment after running the algorithm 150 times. The accuracy reached 92 % with a mean of 83 % and a standard deviation of 7 %. The positive prediction of the model reached 97 % with a mean of 83 % and a standard deviation of 6 %. The true positive rate was achieved at 100 % with a mean of 86 % and a standard deviation of 5 %. Finally, the F1 reached 93 % with a mean of 84 % and a standard deviation of 4 %. These results show the efficacy of the model to make a correct prediction using the whole training set.

The best model associated with the best accuracy found so far was used to predict the classes of the testing dataset 3. Indeed, the author of each paragraph contained in 19 documents (of the testing dataset3) was detected as indicated in Table 12. The proposed model identified the authorship. Each document contains distinct paragraphs written either by humans (Class 0) or by ChatGPT (Class 1). For example, document number 102 (the document numbers are mentioned as they are defined in dataset 3) has 6 paragraphs, 4 paragraphs are written by humans, and 2 paragraphs are written by ChatGPT. Moreover, based on the efficiency of the training model, we can say that classes were predicted with an error rate ranging between 8 % and 17 %. Hence, the proposed model successfully detected the writing style of humans and ChatGPT in one document.

5. Discussion

This study focuses on detecting the ChatGPT and Human writing style. The research objectives were demonstrated as follows.

The first research objective highlights the recognition of documents written either by Humans or ChatGPT using classical classifiers. To fulfill this objective, one experiment and one comparison study were performed. Experiment 1 demonstrated that the classical classifiers (kNN, DT, NB) can categorize the writing of a document into Human or ChatGPT with an accuracy reaching 98 %. This was achieved through the utilization of the feature style extracted from the texts. The setting of the classifiers' hyperparameters is investigated to enhance the classification accuracy. Moreover, the preprocessed dataset used in this classification was normalized. However, the normalization process was effective for the NB classifier and ineffective in DT and kNN. Overall, the comparison study showed that the results provided by these classifiers are prominent.

The second objective emphasizes the recognition of documents written either by Humans or ChatGPT using the ensemble learning classifiers. Experiment 2 proved that the XGBoost and Stacking classifiers successfully detected the writing style of human and ChatGPT documents. They outperformed the classical classifiers, and they are not affected by the normalization process. The proposed strategy successfully recognized documents written by humans or ChatGPT through the writing style.

The third objective is demonstrated through the comparison study, the results of the proposed strategy outperformed the results obtained in Ref. [23]using the dataset1. This is due to the performance of the feature style extraction method compared to the TF-IDF feature extraction technique.

The fourth objective consists of detecting documents written by humans and ChatGPT (called mixed documents) and documents written only by Humans (called simple documents) using the feature style. This objective was also demonstrated in experiment 3 where the ensemble learning classifiers (XGBoost and Stacking) achieved an accuracy of 100 % in categorizing the documents into simple or mixed. Up to one's knowledge, this research objective has not yet been investigated by the researchers.

Table 10
Classification results of the Stacking and XGBoost algorithms using Our_FS dataset2.

	Accuracy	Precision			Recall			F1			Accuracy		
		Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std
XGBoost	Our_FS	0.9777	0.8806	0.04	1.0	0.8591	0.07	1.0	0.8766	0.07	0.9714	0.8642	0.06
Stacking		1.0	0.869	0.07	1.0	0.8872	0.07	1.0	0.8490	0.09	1.0	0.8635	0.05

The fifth objective involves the recognition of paragraphs, written by Human or ChatGPT, in a mixed document. This objective is investigated in experiment 4 using our generated dataset. The XGBoost achieved a classification accuracy of 92 % to classify a paragraph into Human or ChatGPT. The proposed strategy can successfully designate the authors of a document based on the writing style of each paragraph.

The aforementioned experiments were performed using three datasets. Dataset2 and 3 were generated from simple and mixed documents. They are available to allow any researcher to produce this research and/or enhance its results. Thus, objective six is performed.

To conclude, the six research objectives were successfully demonstrated and achieved. It is demonstrated that the feature style extraction method is one of the best feature extraction techniques that contributes to the classification of text documents. Moreover, tuning the hyperparameters of the classical classifiers can boost the classification accuracy.

6. Conclusions

The present study is one of the scientific contributions to education. The aim is to detect plagiarism by recognizing documents written by humans or ChatGPT. It is extremely imperative to recognize students' work from the ChaptGPT-generated work. The article started by introducing the research objectives and the related work. Then, a new strategy based on the stylometric technique was proposed. Many experiments were performed to demonstrate the fulfillment of the research objectives. It has been demonstrated that the proposed strategy successfully detected documents written by humans and chatGPT using the classical classifiers (kNN, DT, and NB) and ensemble learning classifiers (XGBoost and Stacking algorithms). However, the ensemble learning classifiers outperformed the classical classifiers. It has also been proved that the stylometric method outperformed the TF-IDF technique. Moreover, the proposed strategy successfully recognized the simple and mixed documents. Furthermore, the proposed strategy for paragraph authorship (Human or ChatGPT) detection in mixed documents was effectively demonstrated.

The present study can serve the instructors in checking the students' work. It can be extended by implementing a user interface to make its usage more practical. The study can also be extended by investigating the sentence's author detection. One paragraph can be composed of different sentences written by humans and ChatGPT. This research direction will increase the classification accuracy. In future work, we aim to enhance the dataset content and compare the performance of transformer models as well as other deep learning models.

Funding statement

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

Data availability

Data is available on DOI: 10.13140/RG.2.2.31109.12009
https://www.researchgate.net/publication/378546762_ChatGPT-Human-NewGeneratedData?channel=doi&linkId=65df5338c3b52a1170fc70f8&showFulltext=true.

During the preparation of this work, the author(s) used generative AI to improve language. After using this tool/service, the authors reviewed and edited the content as needed and took full responsibility for the content of the publication.

CRediT authorship contribution statement

Lamia Berriche: Writing – review & editing, Writing – original draft, Validation, Resources, Methodology, Investigation, Funding acquisition, Conceptualization. **Souad Larabi-Marie-Sainte:** Writing – review & editing, Visualization, Validation, Supervision, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 11
Classification results of the XGBoost algorithm using Our_FS_Paragraph_Tr dataset.

		Accuracy			Precision			Recall			F1		
		Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std
XGBoost	Our_FS_Parag_Tr	0.9230	0.8325	0.07	0.975	0.8307	0.06	1.0	0.865	0.05	0.9333	0.8457	0.9230

Table 12
Predicted classes (0 for Human and 1 for ChatGPT) for each paragraph using the testing dataset3 (Our_FS_Paragraph_Ts).

Doc #	# of paragraphs	Predicted class	Doc #	# of paragraphs	Predicted class
102	6	0 0 1 0 0 1	19	6	1 1 1 0 1 1
114	6	0 0 1 1 0 0	41	5	0 1 0 0 0
116	5	0 1 0 1 0	42	5	0 0 0 1 1
118	6	1 0 1 0 1 1	63	5	0 1 0 1 1
121	6	0 0 0 1 1 0	79	5	1 0 0 1 1
12102	7	1 0 0 1 1 0 1	80	5	0 0 1 0 0
123	6	0 1 1 1 1 1	8002	6	1 0 1 0 1 1
14	6	0 0 0 1 1 1	85	5	1 1 1 0 1
152	5	0 1 0 1 1	902	5	0 0 0 0 1
97	5	0 0 0 0 0			

Acknowledgments

The authors would like to acknowledge the support of the Artificial Intelligence and Data Analytics Lab (AIDA), PSU, Riyadh, KSA.

References

[1] A. Koubaa, B. Qureshi, A. Ammar, Z. Khan, W. Boulila, L. Ghouti, Humans are still better than ChatGPT: case of the IEEEXtreme competition, Heliyon 9 (11) (Nov. 2023) e21624, https://doi.org/10.1016/j.heliyon.2023.e21624.
[2] A. Koubaa, W. Boulila, L. Ghouti, A. Alzahem, S. Latif, Exploring ChatGPT Capabilities and limitations: a survey, IEEE Access 11 (2023) 118698–118721, https://doi.org/10.1109/ACCESS.2023.3326474.
[3] D.A. Schweidel, M. Reisenbichler, T. Reutterer, K. Zhang, Leveraging AI FOR CONTENT generation: a customer equity perspective, Artificial Intelligence in Marketing Review of Marketing Research 20 (2023) 125–145, https://doi.org/10.1108/S1548-643520230000020006.
[4] Y. Feng, P. Poralla, S. Dash, K. Li, V. Desai, M. Qiu, The impact of ChatGPT on streaming media: a crowdsourced and data-driven analysis using twitter and reddit, in: 2023 IEEE 9th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), May 2023, pp. 222–227, https://doi.org/10.1109/BIGDATASECURITY-HPSC-IDS58521.2023.00046.
[5] Z. Jin, Z. Song, Generating Coherent Comic with Rich Story Using ChatGPT and Stable Diffusion, May 2023 [Online]. Available: https://arxiv.org/abs/2305.11067v2. (Accessed 18 June 2023).
[6] Z. Jin, Z. Song, Generating Coherent Comic with Rich Story Using ChatGPT and Stable Diffusion, May 2023 [Online]. Available: https://arxiv.org/abs/2305.11067v2. (Accessed 18 June 2023).
[7] M.A. Jaber, A. Beganović, A.A. Almisreb, A. Info, Methods and applications of ChatGPT in software development: a literature review, Southeast Europe Journal of Soft Computing 12 (1) (May 2023) 8–12, https://doi.org/10.21533/SCJOURNAL.V12I1.251.
[8] S. Biswas, Role of ChatGPT in computer programming, Mesopotamian Journal of Computer Science 2023 (Feb. 2023) 8–16, https://doi.org/10.58496/MJCSC/2023/002.
[9] M. Imran, N. Almusharraf, Analyzing the role of ChatGPT as a writing assistant at higher education level: a systematic review of the literature, Contemp Educ Technol 15 (4) (Oct. 2023) ep464, https://doi.org/10.30935/CEDTECH/13605.
[10] Md M. Rahman, H.J.R. Terano, M.N. Rahman, A. Salamzadeh, Md S. Rahaman, ChatGPT and academic research : a review and recommendations based on practical examples, Journal of Education, Management and Development Studies 3 (1) (2023) 1–12, https://doi.org/10.52631/JEMDS.V3I1.175.
[11] G. Lu, S.B. Larcher, T. Tran, Hybrid Long Document Summarization Using C2F-FAR and ChatGPT: A Practical Study, Jun. 2023 [Online]. Available: http://arxiv.org/abs/2306.01169. (Accessed 19 June 2023).
[12] M. Jiffriya, M.A. Jahan, R.G. Ragel, Plagiarism detection tools and techniques: a comprehensive survey, Journal of Science-FAS-SEUSL (2) (2021) 47–64.
[13] S.M. Alzahrani, N. Salim, A. Abraham, Understanding plagiarism linguistic patterns, textual features, and detection methods, IEEE Trans. Syst. Man Cybern. C Appl. Rev. 42 (2) (Mar. 2012) 133–149, https://doi.org/10.1109/TSMCC.2011.2134847.
[14] M. Davoodifard, Automatic detection of plagiarism in writing, Studies in Applied Linguistics and TESOL 21 (2) (Jan. 2022) 54–60, https://doi.org/10.52214/SALT.V21I2.9058.
[15] M. Arshad Aiman, B. Khan, S. Ahmad, M. Asim, Predicting age and gender in author profiling: a multi-feature exploration, Comput. Mater. Continua (CMC) 79 (2) (May 2024) 3333–3353, https://doi.org/10.32604/CMC.2024.049254.
[16] S. Yadav, S.S. Rathore, S.S. Chouhan, Authorship identification using stylometry and document fingerprinting, Lect. Notes Comput. Sci. 12581 (LNCS) (2020) 278–288, https://doi.org/10.1007/978-3-030-66665-1_18/COVER.
[17] M.A. Raafat, R.A.F. El-Wakil, A. Atia, Comparative study for Stylometric analysis techniques for authorship attribution, in: 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference, MIUCC, 2021, pp. 176–181, https://doi.org/10.1109/MIUCC52538.2021.9447600. May 2021.
[18] T. Neal, K. Sundararajan, A. Fatima, Y. Yan, Y. Xiang, D. Woodard, Surveying stylometry techniques and applications, ACM Comput. Surv. 50 (6) (Nov. 2017), https://doi.org/10.1145/3132039.
[19] S.M. Mitrović, D. Andreoletti, O. Ayoub, ChatGPT or Human? Detect and Explain. Explaining Decisions of Machine Learning Model for Detecting Short ChatGPT-Generated Text, Jan. 2023 [Online]. Available: https://arxiv.org/abs/2301.13852v1. (Accessed 20 June 2023).
[20] J.D. Rodriguez, T. Hay, D. Gros, Z. Shamsi, R. Srinivasan, Cross-domain detection of GPT-2-generated technical text, in: NAACL 2022 - 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 2022, pp. 1213–1233, https://doi.org/10.18653/V1/2022.NAACL-MAIN.88.
[21] T. Kumarage, J. Garland, A. Bhattacharjee, K. Trapeznikov, S. Ruston, and H. Liu, "Stylometric Detection of AI-Generated Text in Twitter Timelines", Accessed: June. 20, 2023. [Online]. Available: https://github.com/TSKumarage/Stylo-Det-AI-.
[22] H. Desaire, A.E. Chua, M. Isom, R. Jarosova, D. Hua, ChatGPT or academic scientist? Distinguishing authorship with over 99% accuracy using off-the-shelf machine learning tools [Online]. Available: https://arxiv.org/abs/2303.16352v1, Mar. 2023. (Accessed 20 June 2023).
[23] R. Shijaku and E. C. Canhasi, "ChatGPT Generated Text Detection," Preprint. Accessed: August. 23, 2023. [Online]. Available: https://www.researchgate.net/publication/366898047_ChatGPT_Generated_Text_Detection?channel=doi&linkId=63b76718097c7832ca932473&showFulltext=true.
[24] (1) (PDF) ChatGPT-Human-NewGeneratedData." Accessed: March. 1, 2024. [Online]. Available: https://www.researchgate.net/publication/378546762_ChatGPT-Human-NewGeneratedData.
[25] H. Desaire, A.E. Chua, M.G. Kim, D. Hua, Accurately detecting AI text when ChatGPT is told to write like a chemist, Cell Rep Phys Sci 4 (11) (Nov. 2023) 101672, https://doi.org/10.1016/J.XCRP.2023.101672.
[26] B. Guo, et al., "How close is ChatGPT to human experts? Comparison Corpus, Evaluation, and Detection," (Jan. 2023) [Online]. Available: https://arxiv.org/abs/2301.07597v1. (Accessed 24 August 2023).

[27] M. Maktab Dar Oghaz, K. Dhame, G. Singaram, L. Babu Saheer, Detection and Classification of ChatGPT Generated Contents Using Deep Transformer Models, Aug. 2023, https://doi.org/10.36227/TECHRXIV.23895951.V1.
[28] W. Zaitsu Id, M. Jin, Distinguishing ChatGPT(-3.5, -4)-generated and human-written papers through Japanese stylometric analysis, PLoS One 18 (8) (Apr. 2023) e0288453, https://doi.org/10.1371/JOURNAL.PONE.0288453.
[29] OpenAI, "GPT-4 technical report,", ArXiv (2023).
[30] S.M. Mitrović, D. Andreoletti, O. Ayoub, ChatGPT or Human? Detect and Explain. Explaining Decisions of Machine Learning Model for Detecting Short ChatGPT-Generated Text, Jan. 2023 [Online]. Available: https://arxiv.org/abs/2301.13852v1. (Accessed 17 September 2023).
[31] T. Kumarage, J. Garland, A. Bhattacharjee, K. Trapeznikov, S. Ruston, H. Liu, Stylometric detection of AI-generated text in twitter timelines [Online]. Available: https://arxiv.org/abs/2303.03697v1, Mar. 2023. (Accessed 23 September 2023).
[32] W. Liao, et al., Differentiate ChatGPT-generated and human-written medical texts [Online]. Available: https://arxiv.org/abs/2304.11567v1, Apr. 2023. (Accessed 23 September 2023).
33 I. Katib, F.Y. Assiri, H.A. Abdushkour, D. Hamed, M. Ragab, Differentiating chat generative pretrained transformer from humans: detecting ChatGPT-generated text and human text using machine learning, Mathematics (15) (Aug. 2023) 3400, https://doi.org/10.3390/MATH11153400.
[34] S.M. Mitrović, D. Andreoletti, O. Ayoub, ChatGPT or Human? Detect and Explain. Explaining Decisions of Machine Learning Model for Detecting Short ChatGPT-Generated Text, Jan. 2023 [Online]. Available: https://arxiv.org/abs/2301.13852v1. (Accessed 17 September 2023).
[35] M.R. Alrabiah, Linguistic features of legal language: a contrastive study of Saudi Arabia and Canada labour laws [Online]. Available: www.eajournals.org. (Accessed 1 March 2024).
[36] E. Stamatatos, A survey of modern authorship attribution methods, J. Am. Soc. Inf. Sci. Technol. 60 (3) (Mar. 2009) 538–556, https://doi.org/10.1002/ASI.21001.
[37] N. Alghamdi, L. Berriche, M. Alrabiah, Steganalysis of Markov chain-based statistical text steganography, International Journal of Computing and Digital Systems 12 (7) (Dec. 2022) 1553–1559, https://doi.org/10.12785/IJCDS/1201125.
[38] Y. Liu, et al., ArguGPT: Evaluating, Understanding and Identifying Argumentative Essays Generated by GPT Models, Apr. 2023 [Online]. Available: https://arxiv.org/abs/2304.07666v2. (Accessed 15 February 2024).
[39] P. Yu, J. Chen, X. Feng, Z. Xia, CHEAT: A Large-Scale Dataset for Detecting ChatGPT-writtEn AbsTracts, Apr. 2023 [Online]. Available: https://arxiv.org/abs/2304.12008v1. (Accessed 15 February 2024).
[40] D. Singh, B. Singh, Feature wise normalization: an effective way of normalizing data, Pattern Recognit 122 (Feb. 2022) 108307, https://doi.org/10.1016/J.PATCOG.2021.108307.
[41] I. Izonin, R. Tkachenko, N. Shakhovska, B. Ilchyshyn, K.K. Singh, A two-step data normalization approach for improving classification accuracy in the medical diagnosis domain, Mathematics 10 (11) (Jun. 2022) 1942, https://doi.org/10.3390/MATH10111942, 2022, Vol. 10, Page 1942.
[42] W. Wang, J. Zhang, B. Hu, Meta-learning with logistic regression for multi-classification, Smart Innovation, Systems and Technologies 270 (2022) 125–138, https://doi.org/10.1007/978-981-16-8558-3_9/COVER.
[43] B. Pavlyshenko, Using stacking approaches for machine learning models, in: Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018, Oct. 2018, pp. 255–258, https://doi.org/10.1109/DSMP.2018.8478522.
[44] S. Larabi-Marie-Sainte, R. Jan, A. Al-Matouq, S. Alabduhadi, The impact of timetable on student's absences and performance, PLoS One 16 (6) (Jun. 2021) e0253256, https://doi.org/10.1371/JOURNAL.PONE.0253256.
[45] S. Albahli, Efficient hyperparameter tuning for predicting student performance with Bayesian optimization, Multimed Tools Appl (Nov. 2023) 1–25, https://doi.org/10.1007/S11042-023-17525-W/METRICS.
[46] M. Azad, M. Moshkov, Applications of depth minimization of decision trees containing hypotheses for multiple-value decision tables, Entropy 25 (4) (Apr. 2023), https://doi.org/10.3390/E25040547.