ASCILITE 2023

People, Partnerships and Pedagogies

Authorship Verification in software engineering education: Forget ChatGPT and focus on students' academic writing profiles

Shannon Rios, Yu Zhang and Eduardo Araujo Oliveira
The University of Melbourne

The prevalence of academic misconduct, specifically contract cheating, is a rising concern in higher education institutions globally. Among the recent advancements, Generative Artificial Intelligence (genAI) has exacerbated the situation by offering authentically generated writings, making detection through traditional plagiarism tools ineffective. This paper explores the development and application of students' academic writing profiles, using a combination of word embedding (Word2Vec) and stylistic feature extraction techniques. By leveraging a Siamese neural network, our method focuses on recognising distinctive writing styles, a concept rooted in Authorship Verification (AV). Our approach's efficacy evaluates favourably against other AV methods and is tested against AI-generated texts deliberately designed to mimic student writing. The study emphasises the importance of understanding individual academic writing styles to identify outsourcing or AI-generated work effectively.

Keywords: engineering education, authorship verification, academic misconduct, academic cheating

Introduction

Academic integrity issues continue to challenge higher education institutions, with increasing numbers of reported academic misconduct worldwide (Oliveira et al., 2020). A particularly popular form of academic misconduct is contract cheating, which refers to students outsourcing work to be written on their behalf and submitting it for academic credit (Lancaster and Clarke, 2008). Contract cheating is on the rise, with several websites advertising essay ghost-writing services as undetectable (Dawson et al., 2020). Recently, the use of new technologies like Generative Artificial Intelligence (genAI) also significantly increased concerns related to academic cheating (Kasneci et al., 2023). Contract cheating or works generated by Artificial Intelligence (AI) are difficult to detect because they involve the use of authentic (or original) texts; this means plagiarism tools that compare new academic work with existing ones wouldn’t be effective. Additionally, all forms of cheating have been found to be especially prevalent in engineering disciplines (Bretag et al., 2019; Marsden et al., 2005).

One approach to identify if students used genAI or outsourced work requires manual comparisons between new work submitted by students with their previous ones (Foltýnek et al., 2020), which is currently a time-consuming, complex, subjective, and not scalable task. Instead, we can apply machine learning techniques to create a model of student’s individual writing styles. Writing styles are used in Authorship verification (AV) to identify if two specific documents were written by the same author (Koppel and Schler, 2004). It provides a potential method for identifying authorship in the case of authentically generated writing, either from contract cheating or genAI (Oliveira & de Barba, 2022).

In this context, we aimed at developing students’ academic writing profiles and combining these with an existing AV method to perform innovative academic writing similarity detection. Here, we used a popular word embedding technique (Word2Vec) in combination with text stylistic feature extraction techniques to capture effective stylistic attributes from students to better support authorship verification. Further, we incorporated a Siamese neural network to learn the text embeddings derived from our created writing profiles. The performance of our suggested approach is compared to other AV methods. Moreover, we investigated how AV performed when given AI generated texts that were purposefully obfuscated to impersonate students’ writings.

Background literature

Detecting AI-authored text

The launch of ChatGPT received a rapid and controversial response early in 2023. Due to its ability to generate largely accurate, well-structured, original text, it is extremely easy for students to outsource their essays to ChatGPT and submit it as their own (Susnjak, 2022; OpenAI, 2023a). Confounding this is the fact that it is difficult to distinguish AI-generated text from human-written using traditional methods (Kasneci et al., 2023). This is further evidenced by the fact that OpenAI quietly removed its AI text classifier due to it's low rate of accuracy (OpenAI, 2023b). There have been several approaches in recent years to distinguish AI-generated text from human-authored text. Most notably, OpenAI released their own RoBERTa-based GPT-2 detector, which can be fine-tuned over newer Large Language Models (LLMs) (Kirchner et al., 2023). Some approaches involve zero-shot detection which avoids the overhead of training models (Ippolito et al., 2019; Mitchell et al., 2023). Another avenue to AI text generation is "watermarking" the text to ease detection by incorporating hidden patterns in the text that are unnoticeable and do not affect text quality, but algorithmically identify the text as synthetic (Kirchenbauer et al., 2023). Kirchenbauer et al. (2023) proposed a method of soft watermarking by creating a pseudo-random list of red and green tokens, and generating text whose tokens have a high probability of being from the green list. Non-watermarked text will, in contrast, have an equal probability of generating text with either red or green tokens. proposed a method of soft watermarking by creating a pseudo-random list of red and green tokens, and generating text whose tokens have a high probability of being from the green list. Non-watermarked text will, in contrast, have an equal probability of generating text with either red or green tokens.

These state-of-the-art methods are unreliable in practice. Sadasivan et al. (2023) discussed how each of these approaches can be easily evaded through paraphrasing attacks, which involves rephrasing of text either manually or through paraphrasing tools. These tools can erase watermarks and significantly lower the efficacy of other detection tools. The same authors also described spoofing attacks being a threat to watermarking approaches, involving an adversary learning the soft watermarking scheme by repeatedly querying it. Further, the detection tools themselves do not perform well enough to serve as a full solution to detecting AI-authored text. OpenAI's own classifier has only a 26% true positive rate, and a 9% false positive rate (Kirchner et al., 2023).

False positive cases in detection tools are extremely harmful in a sensitive area like academic integrity. If detection tools are treated as fully reliable, they have the potential to ruin the academic careers of falsely accused students. At best, given the present state of generative AI, detection tools should only be considered as a small part of a holistic approach to assessing and making decisions about the originality of a student's work. To profile and detect AI-generated text at this stage is a losing battle, especially as advancements even between GPT-3.5 and GPT-4 are gargantuan (OpenAI, 2023a). There are too many ways to evade even the most powerful detection strategies (Sadasivan et al., 2023) and as a result, approaching the problem by stylometric profiling and authorship analysis may be a better route than attempting to detect AI-authored text. It's worth noting that it is not yet clear how capable GPT-4 and other LLMs are at imitating specific writing styles, or how hard it will be. This is an open avenue of research that should be explored alongside authorship verification.

Authorship verification and academic students writing profiles

AV is the task of identifying an author's writing style, and then determining if a given text was or was not written by that same author (Koppel & Schler, 2004; Potha & Stamatatos, 2014). As a categorisation problem, authorship verification is complex because a single author may intentionally vary their style from text to text for many reasons or may unconsciously drift stylistically over time (Koppel & Schler, 2004). Stylometry is one approach to AV that involves the statistical analysis of a writer's work in order to generate a writing style to be used for comparison (Potthast et al., 2019). The use of stylometry for authorship verification assumes that an author's writing style is consistent and recognizable (Laramée, 2018).

The style of a text can be characterised by measuring a vast array of stylistic features, that includes lexical (e.g., word, sentence or character-based statistic variation such as vocabulary richness and word-length distributions), syntactic (e.g., function words, punctuation and part-of-speech), structural (e.g., text organization and layout, fonts, sizes and colors), content-specific (e.g., word n-grams), and idiosyncratic style markers (e.g., misspellings, grammatical mistakes and other usage anomalies) (Abbasi and Chen, 2008). Stylistic features are the attributes or writing-style markers that are the most effective discriminators of authorship. Over 1000 different style markers have been used in previous research on stylistic analysis, with no consensus on the best set (Rudman, 1997).

Attempts to solve authorship verification problems follow either the instance-based or the profile-based paradigm. The instance-based paradigm treats all available samples by one author separately; in this paradigm each text sample has its own representation. On the other hand, the profile-based paradigm treats all available text samples by one candidate author cumulatively. Text samples are concatenated into a single, often large representative document and then the profile of the author is extracted from that document (Potha and Stamatatos, 2014). Another profile is produced from the questioned document and the two profiles are compared using a dissimilarity function. Due to constant changes and improvements on students' vocabularies among higher education courses, the profile-based paradigm is best for educational settings and will be adopted in this study. Moreover, this paradigm is essential in the creation and maintenance of students' academic writing profiles across several years.

There is currently a lack of research investigating the use of style markers in the creation and management of academic writing profiles for authorship verification problems in software engineering education.

Current study

In the current study we designed and developed an approach to generate student academic writing profiles and combined these with an existing AV method. We then evaluated our method in three stages: (i) we compared the accuracy of this proposed method against PAN14 dataset; (ii) we evaluated how this method performed AV on data collected from software engineering students' assessments in The University of Melbourne; (iii) we expanded our dataset with answers generated by AI and evaluated results again.

Methods

Data Collection

Two main datasets were included in our investigations: PAN14 and SWEN datasets, as shown below in Table 1.

Table 1. Statistics of PAN14 English Essays and Software Engineering (SWEN) corpus

Dataset		Problems	Docs	Avg. of Known docs per problem	Avg. words per doc
PAN 2014 EE	Training	200	729	2.6	848
	Evaluation	200	718	2.6	833.2
SWEN Dataset	Training	12	40	3.3	962.6
	Evaluation	6	19	3.1	1124.7

PAN14

PAN14 dataset was retrieved from the PAN International Competition on Authorship Verification. PAN14 AV English Essays (EE) corpus were derived from the Uppsala Student English (USE) corpus compiled by Axelsson (2000). These are a set of essays collected from English as a Secondary Language (ESL) Swedish students in the year 1999, done over several terms. The corpus consists of 1,489 essays written by 440 Swedish university students of English at three different levels, the majority in their first term of full-time studies. The total number of words is 1,221,265, which means an average essay length of 820 words. A typical first-term essay is somewhat shorter, averaging 777 words. The essays cover set topics of different types. They were written out of class, against a deadline of two to three weeks, length limitations imposed (usually 700 – 800 words), and suitable text structure suggested. The essays included literature discussion, argumentation, reflection, and personal writing. The PAN14 dataset comprises samples that mimic the nature of academic writing. It thus provided an excellent platform for testing our approach, where the objective was to detect and deter authorship fraud within academic contexts. PAN corpora are subdivided into training dataset and test dataset. In each dataset, there are several problems (Table 1). Each problem is a verification task, and contains several documents of known authorship, all by the same author, and one questioned document of unknown authorship (Potha and Stamatatos, 2014). The training dataset is used to train verification models, and the test dataset is used for evaluating the actual performance of models.

The PAN14 competition required results to be a binary 'yes', 'no' or 'unanswered/I don't know' when questioned whether a document is likely to be written by the same author or not. The effectiveness of the model was evaluated using the area under the curve (AUC) and the c@1 measures (Peñas and Rodrigo, 2011). The final ranking of the performance was based on the product of the AUC and c@1 measures.

SWEN
SWEN is the dataset collected from Software Engineering students. All participant students were undertaking the yearlong Masters Advanced Software Project (SWEN90013) subject at The University of Melbourne in 2021. Participants were recruited via e-mail and provided informed consent (Ethics approval #24272). This sample included a total of 18 students. The aim of this subject is to give students the knowledge and skills required to carry out real life software engineering projects. Students work in large teams to develop a non-trivial software system for an external client using agile software engineering methods. The subject has 10 assessments (or deliverables), distributed across two teaching semesters: 6 of them are team-based (group assessment) and 4 of them are individual-based assessments.

As shown below in Figure 1, individual assessments involved writing (A1) personal objectives (10 marks), (A2) ethics report (2 marks), (A3) professional communication report (13 marks) and (A4) individual contribution statement (15 marks). During teaching week 5, students were asked to complete their personal objectives in the classroom (this activity was synchronous and supervised by lecturer) and to submit it to Canvas Learning Management System (LMS). Remaining individual assessments were asynchronous. This dataset was also organised into several training and testing problems as shown above in Table 1

Data preprocessing

After obtaining the answers from software engineering students, the dataset was examined and cleaned. Some students submitted assessments in different file formats: .pdf, .txt, .html, .docx. In some cases, we had to remove HTML tags, metadata, or special characters from these files. Additionally, in this phase, we converted all text to lowercase. We did not strip punctuation marks or numbers to students' texts to preserve specific individual characteristics in the writings.

Data analysis

Our AV investigation was organised in three studies, as shown below in Figure 1.

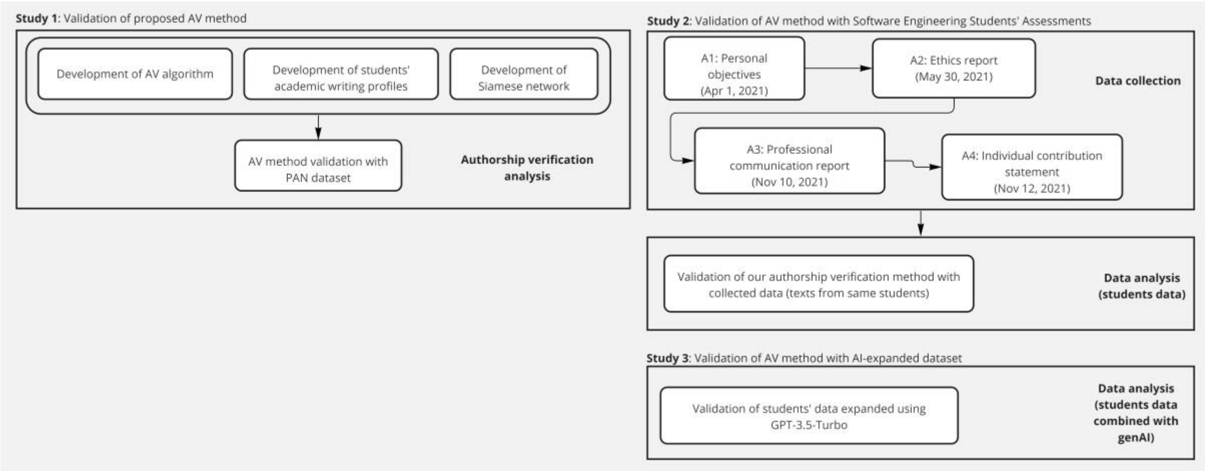
Study 1: Validation of proposed AV method
Development of AV algorithm
Development of students' academic writing profiles
Development of Siamese network
AV method validation with PAN dataset
Authorship verification analysis
Study 2: Validation of AV method with Software Engineering Students' Assessments
A1: Personal objectives (Apr 1, 2021)
A2: Ethics report (May 30, 2021)
Data collection
A3: Professional communication report (Nov 10, 2021)
A4: Individual contribution statement (Nov 12, 2021)
Validation of our authorship verification method with collected data (texts from same students)
Data analysis (students data)
Study 3: Validation of AV method with AI-expanded dataset
Validation of students' data expanded using GPT-3.5-Turbo
Data analysis (students data combined with genAI)

Figure 1. Research Procedure

Study 1: Validation of proposed AV method

In this study, we combined the Word2Vec method and stylometric metrics to create students' academic writing profiles. Word2Vec is a two-layer neural network that is designed to process text by converting words into numerical form, i.e., vectors (Mikolov et al., 2013). This model uses a large text corpus and produces a vector space, usually of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. It also captures the semantic meanings of words based on their context in the corpus, thereby creating meaningful representations of the text. This initial phase of the study happened as follows:

1. Training Word2Vec Model: we first trained a Word2Vec model on PAN14 training corpus. This step was crucial as it helped the model to learn the semantic relationships between words based on different contexts in the corpus. When training the Word2Vec model, we employed the Continuous Bag of Words (CBOW) method, with a vector size of 300, a window size of 5, and a minimum word appearance frequency set at 1.
2. Generating Vector Representations: Post-training, we used this model to derive vector representations for each word in a document. Word2Vec is known for its ability to establish semantic linear relationships in vector space, effectively capturing the nuances of semantic relations between words. This ability also allows it to express these relationships through vector operations to a certain extent.
3. Document Representation: To generate a comprehensive representation for a document, we then took all the individual word vectors within the document, added them, and then calculated the mean. The resulting vector served as the representation of the entire document. This process effectively condensed the information from all the words in a document into a single vector that captured the overall semantic content of the text.

Through this process, we were able to translate each document in PAN14 dataset into a single, coherent vector representation. This representation encapsulated the semantic characteristics of the text, thereby providing a rich, meaningful input for our Siamese network to process (further explained in more detail).

We then focused on deciding what stylometric features should be extracted to capture a comprehensive view of students' writing styles. Here, we focused on three main categories of stylistic attributes (Brizan et al., 2015): lexical features, sentence structure features, and punctuation usage features.

- Lexical Features: we analysed a variety of lexical characteristics to understand students' choice and usage of words. These included the frequency of rare words (number of words with frequency smaller than two in a document), the frequency of long words (number of words with more than six letters), the count of words longer than the average word length, the count of words shorter than the average word length, and the count of words that matched the average word length. Additionally, we calculated the Type-Token Ratio (TTR), a measure of lexical diversity defined as the ratio of unique words to the total number of words.
- Sentence Structure Features: we examined the average sentence length and counted the number of sentences that were longer, shorter, and equal to this average length. This gave us insights into students preferred sentence construction and complexity.
- Punctuation Usage Features: we recorded the frequency of commonly used punctuation marks. Punctuation can reflect an students' writing rhythm and stylistic preferences, providing another dimension to the writing style profile.

To prevent the loss of context and detail that could occur due to the varying sizes of text samples, we did not rely solely on frequency counts. For instance, larger text samples are likely to contain more instances of punctuation marks than smaller samples, so a simple frequency count might not accurately reflect a student's punctuation usage. To mitigate this, we normalised the extracted feature values to account for the text size. Specifically, we divided the counted values by the total word count in the text. This refinement provides a more accurate portrayal of a student's unique writing style, enriching the data set that feeds our method. Lastly, we concatenated the obtained Word2Vec and stylometric feature vectors into a single long vector, which became our representation of students' academic profiles.

After creating representations of students' academic profiles, we also developed a Siamese Network structure to assist with our AV investigations trained using the Adam optimiser (Kingma and Ba, 2014). Siamese networks are a type of neural network architecture designed for similarity comparison tasks. They are commonly used in tasks such as image or text similarity, signature verification, face recognition, and information retrieval. The name "Siamese" comes from the idea that the network architecture consists of two identical subnetworks, which are referred to as "twins". These twin networks share the same architecture and weights. The inputs to the twin networks are typically two instances or samples that need to be compared. The key concept behind Siamese networks is to learn a similarity metric that can measure the similarity or dissimilarity between the inputs. The twin networks process each input independently and generate a fixed-length feature representation (embedding) for each input. These feature representations are then compared using a similarity metric such as cosine similarity or Euclidean distance.

In our study, we have made some improvements to the original Siamese Network structure to better serve our goals: (i) we utilised three identical sub-networks sharing the same weights to transform the input data; (ii) we replaced the final distance function in the traditional Siamese Network with two identical neural network classification modules, which also share the same weights. This modification enabled our network to capture more intricate relationships among stylistic features, enriching the network's representational capacity. We won't discuss these updates in technical details in this paper.

By applying these changes to the network, we ensured that samples from the same students are closely clustered together, while samples from different students are driven further apart in the network. This resulted in a more discernible separation between students' writing profiles, enhancing the accuracy of our AV model.

To train the model we partitioned the data into a 70% training set, a 15% validation set, and a 15% test set. This allowed us to not only use a substantial amount of data for training but also to ensure that our model was properly validated and tested on unseen data, thereby providing more accurate representation of its ability to generalise results. During training phases, we utilised early stopping to prevent overfitting. This mechanism monitored the validation loss and stopped the training when the validation loss ceased to decrease for several epochs, indicating that the model might start to overfit to the training data. Furthermore, the model state that yielded the lowest validation loss was saved and later used for evaluation and prediction, ensuring the best performance.

Study 2: Validation of AV method with software engineering students' assessments

After validating the accuracy of our AV method with PAN14 dataset in Study 1, we tested the accuracy of the same AV method against software engineering students' assessments. Given that the data is sourced from actual course material, it is rich in authentic academic texts, which enables us to evaluate our methodology under practical and real-world conditions. We used the same construction method as the PAN14 EE dataset to construct the software engineering dataset, both of which have 50% positive samples and 50% negative samples. Through the combination of these datasets, we aimed at providing a robust and comprehensive validation of our authorship verification method in higher education settings, ensuring its effectiveness across multiple scenarios, from controlled environments to real-world academic contexts.

Study 3: Validation of AV method with AI-expanded dataset

In our final study, we validated whether our AV method can distinguish between AI-generated content and students' own works. We employed gpt-3.5-turbo model to simulate students' writing styles on the software engineering dataset. The AI-generated contents served as unknown samples in our validation process. This means we first created students' academic writing profiles from their real collected samples and second, we provided students' samples to AI. We then asked AI to generate new samples impersonating the students' writing styles (Prompt 1). Our AV model from study 2 was then used to determine whether these samples originated from the same given student, thereby testing its proficiency in distinguishing AI-generated text from original human-authored text. Our prompt (or instructions) provided to gpt-3.5-turbo model to generate answers to similar questions answered by students during the course are shown in Prompt 1.

```
You are a university student majoring in Software Engineering.
Consider the exemplar text below:
{text}
Your task is to generate a similar text to the provided one.
While impersonating the same student as the input text, follow these steps:
1. **Understanding:** Comprehend the essential qualities of the given text, including its style, tone, and key themes.
2. **Emulation:** Reflect these qualities in your own response.
3. **Originality:** Produce unique output text that stays true to the essence of the given example.
4. **Precision:** Ensure that the detail and language quality of your text mirrors that of the provided text.
5. **Creativity:** Infuse your response with original perspectives, while aligning with the spirit of the given text.
Specifications for your response:
- Your response should closely mirror the information in the text example, without significant deviation.
- Structure your response using paragraphs and bullet points where appropriate (if necessary to match provided text).
- Ensure that your language usage, including capitalisation, matches that of the provided text.
- Ensure the length of the output text is similar to the provided text.
```

Figure 2. Prompt 1. Instructions Provided to gpt-3.5-turbo to Expand Students' Dataset

Results

In our study, documents from PAN14 and SWEN datasets were used to create a separate Siamese network (or model) for each dataset. Both models were trained with the same students' academic writing profiles.

Table 2, shown below, presents the evaluation scores of models trained on the PAN2014 EE dataset. We adopted the evaluation metrics c@1 and Area Under the ROC Curve (AUC), which were used in the PAN14 authorship verification competition, to gauge the performance of our models. When compared to the outcomes of these prior studies, our method surpasses their final scores and also outperforms the provided baseline scores. These results highlight the effectiveness of our method in model might start to overfit to the training data. Furthermore, the model state that yielded the lowest validation loss was saved and later used for evaluation and prediction, ensuring the best performance.

Table 2: Evaluation results on PAN2014 EE dataset

	Final Score	AUC	C@1
Proposed method	0.542	0.786	0.69
Frery et al. (2014)	0.513	0.723	0.71
Satyam et al. (2014)	0.459	0.699	0.657
Moreau et al. (2014)	0.372	0.62	0.6
BASELINE	0.288	0.543	0.53

Our method achieved a c@1 score of 0.69 and an AUC of 0.786, demonstrating a substantial level of accuracy. For comparison purposes, Table 2 presents the performance evaluations of the top three teams from the PAN14 EE subset of the previous competition, along with the BASELINE. These results highlight the effectiveness of our method in distinguishing authorship in study 1, offering promising implications for its application in fields such as academic integrity and forensic linguistics.

Table 3: Evaluation results on SWEN dataset

	Final Score	AUC	C@1
Proposed method	0.506	0.7	0.722
BASELINE	0.262	0.525	0.5

Table 3 above showcases the results obtained on the SWEN dataset (study 2). It is observed that as the volume of the dataset samples decreases, there is a corresponding decline in the model's performance. Nevertheless, the model still demonstrates promising outcomes, indicating its robustness even in situations where data is limited. In this context, we employed a new BASELINE model, which operated on the premise of simple random guessing. This technique is often used to assess the effectiveness of classification models. A comparison of the final scores revealed that our method vastly outperformed this random guessing approach, reinforcing the potential of our method in discerning authorship even with constrained data availability.

Results from study 3 are shown below in Table 4. As can be seen, the model showed great results in attaining a c@1 score of 0.778, which is significantly greater than the baseline and even outperforms the study 2 results. Because in this study all the unknown samples were AI-generated negative samples, the AUC result was not relevant, so we have only calculated the c@1 scores.

Table 4: Evaluation results on SWEN dataset

	Final Score	AUC	C@1
Proposed method + GPT 3.5	-	-	0.778
BASELINE	-	-	0.489

These findings demonstrated that our model showed promising performance in discerning content generated by AI impersonating students, indicating its effectiveness and potential applicability in authorship verification tasks in higher education.

Discussions

The results presented above show that student writing profiles could be an effective way to evaluate authorship of a given text. As with all automated tools for identifying breaches of academic integrity, this evidence should only be used as an indication of possible student cheating. Alluqmani and Shamir (2018) noted in their research the dynamic nature of students' writing style. As students' progress through their studies, their vocabulary, style, and sentence construction inevitably evolve. This poses a significant challenge for models predicated on a fixed stylistic pattern. To address this, our model was designed within Siamese network, which can accommodate and learn the evolutionary trajectory of a student's writing style over time. We adopted a profile-based approach of continuous model updating through new texts (i.e., essays) submitted by students. Given that students in a learning environment typically submit new work on a regular basis, we leverage these submissions to continually update our model. This allows us to maintain an up-to-date profile of the student's writing style, ensuring our model's adaptability and effectiveness in authorship verification. By doing so, our approach remains flexible and adaptable, dynamically adjusting to accommodate any shifts in the student's writing style. Furthermore, as we receive more essays, our model benefits from a richer set of features, thereby enhancing the reliability of the student's writing profile. This ensures that our model remains accurate and relevant despite the inherent fluidity of a student's writing style. Though we did not incorporate certain relevant experiments into our present study, the construction method of word embeddings theoretically allows for this. It is worth noting that future research could endeavour to put this theory into practice through carefully designed experiments, thereby establishing its viability more concretely. Some related work that could be referred to for this purpose includes studies conducted by Can and Patton (2004) and Alluqmani and Shamir (2018). Moreover, future research could investigate how students' writing profiles change during their courses (i.e., increase and/or decrease in technical vocabulary, overall changes in vocabulary, correlations between writing profiles and academic performance), which is a contribution beyond the use of profiles in authorship verification problems.

Our study is also pioneer in applying AV methods to academic AI-generated content that explicitly aimed at impersonating students' styles. We used OpenAI's GPT-3.5-Turbo model to generate text that simulated students writing style and our model identified with a reasonably level of accuracy that these AI-produced samples were unrelated to the original author. This result reveals the model's potential to detect instances of AI-generated content, thereby enhancing the reliability of academic dishonesty detection. Our results are especially important for verification tasks in the face of the ever-growing capabilities of AI in text generation. This research can be further expanded to explore the influence of prompts on generating sample texts. Specifically, how do AI-generated texts that don't aim to replicate the source text compare to those that do? Additionally, are there more sophisticated prompts that can more accurately emulate a student's work? . With the rapid advancements in AI technologies and the increasing sophistication of AI-generated text, such models can become increasingly valuable.

This study also has a few limitations. Firstly, only a single prompt was used to generate spurious texts and only the GPT3.5 LLM was used. Future endeavors could explore the implementation of more robust and sophisticated prompts to generate samples, or even experiment with advanced or bespoke AI models such as GPT-4. Secondly, considering the limited number of participants in the data collection process, it remains an open question whether the AV method proposed in this study could be generalised and applied to a larger sample of academic writings. Although our model was subjected to extensive training and evaluation across various datasets, the range of authorship styles, topics, and genres represented in these datasets wasn't all-encompassing. Therefore, future studies should endeavor to incorporate a more diverse array of datasets to improve the model's applicability across different contexts. Broadening the scope to test the model's ability to discern authorship in non-English texts would also extend its utility across a wider range of situations. Such an expansion presents additional opportunities for examining the model's flexibility and adaptability across diverse scenarios. Finally, for this model to be used in practice, there needs to be a natural and secure way of gathering initial sample texts from students to generate accurate writing profiles that could be used in future AV investigations. Controlled experiments at early stages in students' courses could address this problem.

One of the main strengths of our approach, however, is its robustness against variations in the volume of data. Even with constrained data availability, our model yielded feasible outcomes, indicating its capacity to maintain a certain level of performance. This aspect of our approach extends the existing body of research, which often struggles with limited data scenarios. Our study offers substantial insights and pioneering techniques in the realm of authorship verification in higher education and the use of generative AI.

Conclusions and future work

The main goal of our investigation was to establish an efficient mechanism capable of distinguishing diverse writing styles of students with reasonable accuracy, while also discerning between student's original work from content generated by an AI model. To this end, we implemented an amalgamation of cutting-edge techniques to extract a rich feature-set from corpus, and a novel variant of the Siamese neural network, to push the boundaries of authorship verification accuracy.

The innovative aspect of our method was the fusion of word embeddings and stylometric analysis to create unique academic writing profiles from the corpus as a student's idiosyncratic writing style that involves more than superficial semantic examination. While word embeddings helped to encapsulate the semantic context, stylometric metrics allowed us to shed light on the unique characteristics of individual students writing styles. To effectively interpret these features, we developed a novel Siamese neural network. Our variant of this network introduced an additional neural network classifier layer into the traditional Siamese structure, intended to capture more sophisticated feature relationships and thus enhance the model's performance in authorship verification tasks.

Through our research, we observed that our proposed methodology could perform consistently across multiple datasets, thereby indicating its potential for reliable authorship verification. The potential implications of our research extend beyond software engineering courses and promise to be relevant in real-world scenarios, thereby justifying continued exploration and refinement of our methods in future investigations.

References

Abbasi, A., & Chen, H. (2008). Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems (TOIS)*, *26*(2), 1-29. https://doi.org/10.1145/1344411.1344413

Alluqmani, A., & Shamir, L. (2018). Writing styles in different scientific disciplines: a data science approach. Scientometrics, 115(2), 1071-1085. https://doi.org/10.1007/s11192-018-2688-8

Axelsson, M. W. (2000). USE-the Uppsala Student English corpus: an instrument for needs analysis. ICAME journal, 24, 155-157.

Bretag, T., Harper, R., Burton, M., Ellis, C., Newton, P., Rozenberg, P., ... & van Haeringen, K. (2019). Contract cheating: A survey of Australian university students. Studies in higher education, 44(11), 1837-1856. https://doi.org/10.1080/03075079.2018.1462788

Brizan, D. G., Goodkind, A., Koch, P., Balagani, K., Phoha, V. V., & Rosenberg, A. (2015). Utilizing linguistically enhanced keystroke dynamics to predict typist cognition and demographics. International Journal of Human-Computer Studies, 82, 57–68. https://doi.org/10.1016/j.ijhcs.2015.04.005

Can, F., & Patton, J. M. (2004). Change of writing style with time. Computers and the Humanities, 38, 61-82. https://doi.org/10.1023/B:CHUM.0000009225.28847.77

Dawson, P., Sutherland-Smith, W., & Ricksen, M. (2020). Can software improve marker accuracy at detecting contract cheating? A pilot study of the Turnitin authorship investigate alpha. Assessment & Evaluation in higher education, 45(4), 473-482. https://doi.org/10.1080/02602938.2019.1662884

Foltýnek, T., Meuschke, N., & Gipp, B. (2019). Academic plagiarism detection: a systematic literature review. ACM Computing Surveys (CSUR), 52(6), 1-42. https://doi.org/10.1145/3345317

Ippolito, D., Duckworth, D., Callison-Burch, C., & Eck, D. (2019). Automatic detection of generated text is easiest when humans are fooled. arXiv preprint https://doi.org/10.48550/arXiv.1911.00650 https://doi.org/10.18653/v1/2020.acl-main.164

Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., ... & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences, 103, 102274. https://doi.org/10.1016/j.lindif.2023.102274

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint https://doi.org/10.48550/arXiv.1412.6980

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., & Goldstein, T. (2023). A watermark for large language models. arXiv preprint https://doi.org/10.48550/arXiv.2301.10226

Kirchner, J. H., Ahmad, L., Aaronson, S., & Leike, J. (2023). New AI classifier for indicating AI-written text. OpenAI.

Koppel, M., & Schler, J. (2004, July). Authorship verification as a one-class classification problem. In Proceedings of the twenty-first international conference on Machine learning (p. 62). https://doi.org/10.1145/1015330.1015448

Lancaster, T., & Clarke, R. (2008). The phenomena of contract cheating. In Student plagiarism in an online world: Problems and solutions (pp. 144-159). IGI Global. https://doi.org/10.4018/978-1-59904-801-7.ch010

Laramée, F. D. (2018). Introduction to stylometry with Python. The Programming Historian. https://doi.org/10.46430/phen0078

Marsden, H., Carroll, M., & Neill, J. T. (2005). Who cheats at university? A self-report study of dishonest academic behaviours in a sample of Australian university students. Australian Journal of Psychology, 57(1), 1-10. https://doi.org/10.1080/00049530412331283426

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint https://doi.org/10.48550/arXiv.1301.3781

Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., & Finn, C. (2023). Detectgpt: Zero-shot machine-generated text detection using probability curvature. arXiv preprint https://doi.org/10.48550/arXiv.2301.11305

Oliveira, E., Conjin, R., de Barba, P., Trezise, K., Van Zaanen, M. & Kennedy, G. (2020). Writing Analytics Across Essay Tasks with Different Cognitive Load Demands. In S. Gregory, S. Warburton, & M. Parkes (Eds.), ASCILITE's First Virtual Conference. Proceedings ASCILITE 2020 in Armidale (pp. 60–70). https://doi.org/10.14742/apubs.2022.177

Oliveira, E., & de Barba, P. G. (2022). The impact of cognitive load on students' academic writing: an authorship verification investigation. In Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education 2022: Reconnecting relationships through technology (p. e22177). Australasian Society for Computers in Learning in Tertiary Education. https://doi.org/10.14742/apubs.2022.177

OpenAI (2023a). GPT-4 Technical Report. ArXiv:2303.08774.

OpenAI (2023b, September 15). New AI classifier for indicating AI-written text. https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text

Peñas, A., & Rodrigo, A. (2011). A simple measure to assess non-response.

Potha, N., & Stamatatos, E. (2014, May). A profile-based method for authorship verification. In Hellenic Conference on Artificial Intelligence (pp. 313-326). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-07064-3_25

Potthast, M., Rosso, P., Stamatatos, E., & Stein, B. (2019). A decade of shared tasks in digital text forensics at PAN. In Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part II 41 (pp. 291-300). Springer International Publishing. https://doi.org/10.1007/978-3-030-15719-7_39

Rudman, J. (1997). The state of authorship attribution studies: Some problems and solutions. *Computers and the Humanities*, *31*, 351-365. https://doi.org/10.1023/A:1001018624850

Sadasivan, V. S., Kumar, A., Balasubramanian, S., Wang, W., & Feizi, S. (2023). Can ai-generated text be reliably detected?. arXiv preprint https://doi.org/10.48550/arXiv.2303.11156

Susnjak, T. (2022). ChatGPT: The end of online exam integrity?. arXiv preprint https://doi.org/10.48550/arXiv.2212.09292

Zobel, J. (2004, January). " Uni cheats racket" a case study in plagiarism investigation. In Proceedings of the Sixth Australasian Conference on Computing Education-Volume 30 (pp. 357-365).

Rios, S., Zhang, Y. & Oliveira, E. A. (2023). Authorship Verification in software engineering education: Forget ChatGPT and focus on students' academic writing profiles. In T. Cochrane, V. Narayan, C. Brown, K. MacCallum, E. Bone, C. Deneen, R. Vanderburg, & B. Hurren (Eds.), *People, partnerships and pedagogies*. Proceedings ASCILITE 2023. Christchurch (pp. 195 - 204). https://doi.org/10.14742/apubs.2023.559