Identifying AI Generated Scientific Abstracts using Quantum Machine Learning

Md Siam Ansary*[1,2]

[1] Ahsanullah University of Science and Technology, Dhaka 1208, Bangladesh
[2] Institute of Information Technology, University of Dhaka, Dhaka 1200, Bangladesh
*Corresponding Email: siamansary.cse@gmail.com

Abstract. Ever since artificial intelligence has gained popularity, it is being used and applied to many day-to-day things, which was quite unimaginable previously. Currently, AI tools are used to write many texts and documents. Even for preparing academic as well as business reports such platforms are being utilized. The ability of AI to produce texts and reports is tremendous, but it is also important to be able to identify which text is written by humans and which has been generated using AI platforms for the prevention of potential misuse and fraud. In this research, we have tried to address this issue for identification of AI Generated scientific abstracts because such abstracts are essentially fake and can lead to extremely dangerous and harmful scenarios. We have created a dataset for the task and fifty percent of it is AI generated while the rest are authentic and human written. On the dataset, we have employed state-of-the-art Quantum Machine Learning technique for classification and the approach achieved promising results. All research materials of this study will be made publicly available so that research community can utilize them.

Keywords: dataset, quantum machine learning, natural language processing, artificial intelligence

1 Introduction

Over the last few decades, research in different fields of artificial intelligence has been widely carried out. Nowadays, AI technologies are implemented in the areas that could not be even imagined several years ago, such as the generation of texts with the help of AI platforms.

Currently, there are many AI systems and applications like Gemini, ChatGPT, Microsoft Copilot, and QuillBot that can be used to create or paraphrase text. These platforms make many processes easier but overreliance on them may be very risky. Recent cases have demonstrated the use of AI by students to write academic reports without a human touch, and by workers to use such tools blindly to write business-related documentation.

This necessitates the need to detect AI-generated texts in order to differentiate them and those produced by humans in critical settings. It was in this regard that we have conducted this research.

M. S. Arefin et al. (eds.), *Proceedings of the International Conference on Intelligent Data Analysis and Applications (IDAA 2025)*, Advances in Intelligent Systems Research 206,
https://doi.org/10.2991/978-94-6239-664-7_70

Research papers are usually published by scientists when they have brought about some major contributions that are novel. The abstracts of these papers provide a good insight. Artificial intelligence can create writing that can be close to the authentic scientific abstracts, but the abstracts written by machines do not have any substantiation basis and, when published publicly, can be misused. That is why it is of utmost importance to detect fake, AI-created scientific abstracts properly.

In this regard, we have created a new dataset in this task. The sample is a collection of original research abstracts, based on various repositories, and a wide range of fields, as well as machine-generated similarities. It holds large amount of records of which half are the real and the other half are the fake abstracts.

We used a Quantum Machine Learning (QML) method to categorize the records and the results were extremely satisfactory, despite the experimental nature of the method. QML combines quantum computing and machine learning algorithms by using quantum mechanical principles to encode data into quantum states and execute operations that would be exponentially more efficient than classical models.

In the following sections, we refer to associated earlier experiments, methodology used to prepare the dataset, and the performance of QML on the curated dataset.

2 Literature Review

We have studied the existing researches on AI generated text detection to gain insights which can be very helpful.

Abburi et al. wrote an article on the capture of text produced by AI through an ensemble approach [1]. There were several datasets used to carry out evaluation. Under the adopted strategy, each input would go through five pre-trained models including DeBERTa Large, RoBERTa with XNLI, RoBERTa Large, RoBERTa Base OpenAI detector and XLM-RoBERTa NLI. In order to exploit the benefits of these models, independent generated class probabilities were concatenated and then made inputs to classifier in form of vote in a voting, which included Logistic Regression, Random Forest, Gaussian Naive Bayes and Support Vector machine. This experiment was able to deliver a maximum accuracy of 78.4%.

Shah et al. used a hybrid method that entailed integration of style elements when detecting AI-generated texts (AGTs) [2]. Several lexical characteristics, readability, and vocabulary variety and richness were taken into account. The scientists took custom sets of data to carry out the study. The experimental process was started with the classification of the results with Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, and Gradient Boosting; the next step was the combination of the findings of the classifiers with the highest performance with two explainable AI (xAI) algorithms SHAP and LIME. The cross-breeding method brought positive outcomes.

Yan et al. [3] constructed their own dataset and used RoBERTa and Support Vector machine (SVM) to detect AGT. The RoBERTa model was further refined, and the SVM model used a set of features, which were produced using an internal scoring system. RoBERTa had 99.75% accuracy as compared to SVM with 96.

Mitchel et al. [4] used the probability curvature to the detection of texts generated by machines. The researchers hypothesized that big language model (LLM) produced texts that are negatively curved. They confirmed that with minimal rewrites of model-generated texts, the log-probability decreases under the model than with the original text whereas the opposite is true with human-written texts.

In the context of the work done by Yadagiri et al. [5], the emphasis was put on various linguistic characteristics when it came to identifying AGTs with the help of pre-trained models. The researchers created a sample that included about forty thousand queries and answers of ChatGPT and humans. The models used were CNN -BiLSTM, RNN, GPT -2, and RoBERTa. The best performing among them was RoBERTa with an accuracy of 99.73.

The Zhang et al. [6] article on AI-generated texts detection used a hybrid approach to the issue at hand. In the hybrid model, TF-IDF was then selected with a Bayesian classifier, Stochastic Gradient Descent (SGD) classifier, Light-GBM classifier, CatBoost classifier, and a combination of a Bayesian classifier and many instances of a Bayesian classifier, along with the use of the tokenisation method in the form of a byte-pair encoding (BPE). This pipeline was then run using twelve DeBERTa v3 Large models. The highest ROC-AUC of 97.5 percent was obtained in the course of the experimental approach.

Qazi et al. [7] created a new dataset and put forward a GPT-generated text detector in English in a model format. The researchers developed GPT Reddit Dataset (GRiD) that comprises 6513 samples; 1368 were created by the GPT-3.5-turbo model, and the rest of them were written by human participants. Human generated writings were taken out of Reddit via the Python training project PRAW and the posts taken were posted before the actual launch of the ChatGPT web application. The authors proposed a new approach, GPTEN, in which the analysis of the patterns is performed on the data using the decomposition of tensors. They compared BERT with SVM and Random Forest with the proposed model and discovered that BERT had the best performance.

Gryka et al [8] studied the status of AI email content detection on the basis of many different classifiers, such as Decision Tree, Random Forest (100 trees), MLP classifier, KNN (K=5), SVM with radial basis function kernel, AdaBoost (50 Decision Trees as estimators with a learning rate of 1), Gaussian Naive Bayes, and Quadratic Discriminant Analysis. The material of the study included more than 10950 emails, and the authors studied English and Polish corpora. The best performance was in the Random Forest classifier with F1 score of 98.8 92.1 burns English and Polish, respectively.

The article by Zhou and Wang [9] concentrated on at least five fields of AI-generated text detection. The authors created a corpus of their study by the combination of a number of datasets. They utilized RoBERTa -Ranker, which is

an adapted version of RoBERTa added with a margin ranking loss function and a mean pooling layer at the end of the encoding layer.

Guo et al. [10] have created a pipeline of AI-generated text detection which combines a multi-level contrastive learning strategy with multi-task learning (guo2024a). The samples were first pre-encoded and features extracted. In any particular text, the distance between the encoded features and every constituent feature vector in the total set of features was calculated. Lastly, a k -Nearest neighbours classification technique was used.

Wu et al. [11] experimented with the use of different detectors to detect text generated by large-language models. The existing datasets and platforms were used to find human-written texts, and AI-written texts were generated using GPT -3.5-turbo, PaLM -2-bison, Claude -instant, and Llama -2-70b.

Guo et al. argued that, when exposed to human data, language model trained in a non-optimistic style has more memory on prior token information and lower predictive likelihoods on the subsequent token probabilities in the output logits compared to language model trained in an optimistic manner [12]. These researchers used a 2-way methodology of calculation that quantifies cross-entropy losses between the output logits of the model and its actual forward token, and between the output logits and the token immediately before it.

Wang et al. [13] have used GBERT to detect AI-generated text and they used a large personal set of 670 human-written texts and 708 AI-generated texts. The experiment obtained a final average training accuracy of 98.1 percent and a test accuracy of 97.71 percent.

Yu et al. [14] have created a dataset to identify Chat-GPT generated abstracts, and the dataset included 15,395 human created abstracts and 35,304 ChatGPT created abstracts. The abstracts were searched on IEEE Xplore where human-written abstracts were identified with the use of certain key words, the abstracts generated by AI were classified into three categories: Generation, Polish, and Mix. Under the Generation category, abstracts were solely generated by ChatGPT, in Polish, a human-written text was entered and later revised by the model, and with Mix, the abstracts consisted of ChatGPT-refined and human text. A number of BERT variants were tested on this dataset, with the highest score in AUC in Polish (99.56), Generation (100) and Mix (87.83).

The researchers identified the following AI-generated text recognition systems: ArguGPT, RADAR (Robust AI Text Detector via Adversarial Learning), and OpenAI Detector; they used the M4 and HC3 datasets in their studies [15].

The hypothesis of McGovern et al. [16] is that simple classifiers, with the right features, can identify AI-generated texts. They used GradientBoost classifier, but with three sets of features word n-grams, character n-grams and part-of-speech n-grams, which they demonstrated to yield promising results on several datasets.

Varadararajan et al. [17] argued that the outputs produced by LLM are rather inhumane and unproductive in terms of diversity. They put much stress on the human factors like demographics, personality, empathy and the behavioral linguistic features. Multiple datasets were provided in their study, with a spectral

clustering based on spectral data carried out with the help of a radial basis function to perform unsupervised classification.

Andric et al. [18] took texts as graph representations and integrated them with graph neural networks (GNNs) in detecting AI-generated texts. Other node-feature initialization schemes had been taken into account and co-occurrence graph was used. Pre-trained models, i.e., BERT-Base-Uncased and RoBERTa-Base, were used in the process of fine-tuning. The processed graph was then imported into a GNN using a GAT layer, and the resulting graph document embedding was inputted into a classifier, which was a dense neural network. Evaluation was done using the Autefication 2023 dataset.

Abbas [19] utilised a zero-shot prompt with Sentence BERT (SBERT) to identify AI-generated text identifications. Also, an implementation of a graph convolutional network model was added and contributed to the overall accuracy. The experimental corpus was PAN -AP -2019 dataset.

In the research article, the authors utilized several machine-learning and deep-learning classification models to detect the text produced by AI in a study published in 2025 [20]. They gathered a sample of 509 pairs of descriptive questions and answers, whose responses were provided by humans and by Chat-GPT engine. The model based on RoBERTa was better than the others.

Alhijawi et al. [21] developed 3000 recordings to detect the content generated by the LLCM with 3 categories: Human-written, AI-generated, and Mixed, with 1000 records each category. The authors tried several methods and one hybrid between MLP and CNN with the highest possible accuracy being 0.876 in the experiment.

3 Methodology

3.1 The Novel Dataset

The dataset comprises of a good number of records and fifty percent of them are authentic, collected from different research papers of scientific repositories.

The collected authentic abstracts are from research papers of various domains.

We have collected the abstracts from research papers on two very prominent databases, they are IEEE Xplore and ScienceDirect. 60% of the abstracts are from IEEE Xplore while the rest were are from ScienceDirect.

In the dataset, for creating the AI generated scientific abstracts, we have used ChatGPT. In its web interface prompt, humans have provided instructions and according to the input instructions, abstracts are generated. We have used prompt like: "Generate a scientific abstract (150–200 words) in the domain of <DOMAIN> describing a hypothetical research study with realistic methodology and results."

In the GitHub repository `MdSiamAnsary\ClassificationTasks` [23], the dataset is available.

3.2 Application of Quantum Machine Learning for Classification

Table 1. Training Behavior of the Quantum ML Model

Training Aspect	Observation
Number of Qubits	4
Circuit Type	Variational PQC
Feature Dimension	4 (after PCA)
Optimization Method	Stochastic Gradient Descent (SGD)
Learning Rate	0.01
Number of Epochs	20
Loss Function	Mean Squared Error (MSE)
Loss Trend	Monotonically decreasing
Final Training Loss	0.1509

In Table 1, we use the quantum model of a parameterized quantum circuit (PQC) for classification. It is also referred to as a variational circuit. It consists of Input encoding gates, Trainable gates and Measurement. Input encoding gates are used to map classical data into quantum states; Trainable gates, which are variational layers, are to introduce adjustable parameters; Measurement produces output that is used to calculate loss. The model learns by adjusting parameters in the variational gates to minimize a loss function.

In our implementation, we have used a compact pre-trained sentence encoder for the vectorization of the texts. A BERT mode, MiniLM (all-MiniLM-L6-v2) [22], has been used for the task. The model transforms each sentence to a 384-dimensional dense vector.

It was needed to reduce the dimensionality as quantum circuits can handle less than or equal to five features. Hence, using Principal component analysis (PCA), the 384D vector is reduced to 4D.

Then, feature normalization has been performed. The 4D vectors are normalized into the range $[0, \pi]$ for input into rotation gates.

Next, label encoding has been performed and the dataset has been split for training and testing. We have done 75:25 split for training and testing purposes.

Subsequently, we have defined the quantum circuit. Firstly, a 4-qubit simulator has been set up. The classical features are encoded into qubit rotations which were scaled to $[0, \pi]$ previously. Later, between adjacent qubits, entanglement is introduced and a parameterized variational layer is applied for learning. In the implementation, a simple gradient descent optimizer has been used. The training is done for 20 epochs.

After the training phase is complete, the classification model is evaluated on the text set.

To summarize, In this study, we employ a 4-qubit parameterized quantum circuit (PQC) implemented using the PennyLane framework to perform binary classification. The architecture begins with an encoding layer, where classical input features are mapped to quantum states through Ry rotation gates, such that

each feature x_i is encoded as $Ry(x_i)$ on an individual qubit. This is followed by an entanglement layer designed to capture quantum correlations across qubits using a linear chain of CNOT gates arranged as $q0 \rightarrow q1 \rightarrow q2 \rightarrow q3$. The circuit then incorporates two variational layers comprising trainable single-qubit rotations Ry and Rz, resulting in a total of eight trainable parameters that are optimized during learning. For readout, we perform a Pauli-Z measurement on the first qubit, and the resulting expectation value is passed through a sigmoid activation function to produce a probabilistic output for binary classification. The PQC is trained using gradient descent optimization with a learning rate of 0.01 over 20 epochs. All experiments are executed on the PennyLane `default.qubit` simulator, running on a workstation equipped with 8 GB RAM and an Intel i7 CPU, offering a flexible and reproducible platform for hybrid quantum–classical machine learning research.

3.3 Evaluation Results

The research is connected with the problem of classification. In this way, we have measured the quantum model based on Accuracy, Precision, Recall and F1 score (see Table 2).

Accuracy the proportion of the correctly predicted cases over the total number of the samples; this gives an approximate concept of the classification capability of the model. Precision is the proportion of the actual predictions of positive to the sum of the predicted positives, and it shows how accurately the model is not false. Recall or sensitivity tells us the rate of the true positive cases which the model can identify successfully i.e. they are the one which is resistant to false negatives. The F1 score is the harmonic mean of Precision and Recall: which is an averaging of both complementary measures and is a more dependable measure when the classes are not equal.

The quantum machine learning pipeline has demonstrated that the classical-quantum model is learning successfully. In the more than 20 epochs, the loss value grew gradually downward between 0.3856 and 0.1509, which illustrates that the parameterized quantum circuit (PQC) is actually being trained on the training data by optimizing the variational weights, as seen in Table 3 and Table 4. The model in the test set had a perfect recall of 1.0, thereby causing the model to identify all the positive (AI-generated) samples, but a precision of 0.76, indicating that the model may have resulted in false positives in the predicted AI texts. The resulting F1-score of 0.864 indicates a good trade off between accuracy and recall, indicating that the quantum system, with just 4 qubits and one variational layer can compete reasonably with classical ML models to this binary text classification problem.

We observe that in terms of all the four metrics, the quantum model obtains scores which is very commendable.

Training QML can be more challenging compared to classical models as the model may face barren plateaus due to deep circuits and classical to quantum mapping causes encoding overhead. Despite the challenges, our model has gained very respectful results, being at early stages of an experimental approach

Table 2. Performance Metrics of the Proposed Quantum Machine Learning Model

Metric	Value
Accuracy	0.854
Precision	0.760
Recall	1.000
F1-score	0.864

Table 3. Detailed Epoch-wise Training Performance of the Proposed QML Model

Epoch	Training Loss	Learning Behavior
1	0.3856	Random initialization, high loss
2	0.3709	Early learning begins
3	0.3530	Rapid parameter updates
4	0.3327	Stable descent observed
5	0.3111	Strong early convergence
6	0.2897	Continued steady improvement
7	0.2696	Reduced loss variance
8	0.2514	Transition to fine learning
9	0.2353	Gradual refinement
10	0.2212	Mid-training stability
11	0.2089	Slower, consistent convergence
12	0.1982	Improved generalization behavior
13	0.1889	Fine-tuning phase
14	0.1807	Near-optimal region reached
15	0.1736	Diminishing returns
16	0.1675	Saturation trend begins
17	0.1623	Minor refinements only
18	0.1578	Near-converged regime
19	0.1540	Minimal improvement
20	0.1509	Final converged model

Table 4. Epoch-wise Training Summary of the Proposed QML Model

Epoch Range	Loss Range	Observation
1–5	$0.3856 \rightarrow 0.3111$	Rapid initial learning
6–10	$0.2897 \rightarrow 0.2212$	Stable convergence phase
11–15	$0.2089 \rightarrow 0.1736$	Fine-tuning of parameters
16–20	$0.1675 \rightarrow 0.1509$	Saturation and convergence

3.4 Baseline Models and Comparative Evaluation

Table 5. Performance Comparison of Baseline Models

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Logistic Regression	0.88	0.89	0.88	0.88	0.981
SVM (RBF Kernel)	0.88	0.89	0.88	0.88	0.974
Random Forest	0.90	0.91	0.90	0.90	0.983
XGBoost	0.93	0.93	0.93	0.93	0.983
BERT-base Fine-Tuning	0.976	0.955	1.00	0.977	–
RoBERTa-base Fine-Tuning	0.976	1.00	0.95	0.974	–

To provide a comprehensive comparison against the proposed quantum model, we evaluate a diverse set of classical and transformer-based baseline architectures. Classical baseline models include Logistic Regression, Support Vector Machine (SVM) with an RBF kernel, Random Forest with 300 decision trees, and XGBoost configured with a maximum tree depth of 6. These models are selected due to their strong performance and frequent use in traditional machine learning pipelines for structured data classification. Hyperparameters were tuned using grid search to ensure fair comparison, and all models were trained using 5-fold cross-validation to mitigate overfitting and improve robustness.

In addition to classical baselines, we incorporate Transformer-based baselines to evaluate performance against state-of-the-art deep learning approaches that leverage contextual representation learning. Specifically, we fine-tune *BERT-base* for 3 epochs using a learning rate of 2×10^{-5}, and similarly fine-tune *RoBERTa-base* for 3 epochs under identical training settings. Both transformer models utilize their pretrained tokenization schemes and are optimized using AdamW with linear learning-rate warm-up. These models represent strong modern baselines for natural language tasks and serve to highlight the performance difference between classical, quantum, and deep contextualized learning frameworks.

ROC-AUC (Area Under the Receiver Operating Characteristic Curve) assesses the model's discriminative capability across all classification thresholds, where higher values indicate stronger separation between positive and negative classes.

According to the analysis of the baseline models, the traditional machine learning classifiers, Logistic Regression, SVM (RBF), Random Forest, and XGBoost, performed well in the dataset with the accuracy of up to 0.88 of the Logistic Regression, 0.93 of XGBoost models (see Table 5). Precision, recall and F1-score have shown that there is a balanced performance of the two classes with XGBoost performing slightly better (F1-score 0.93 between the two classes, ROC-AUC 0.9833). The other model that performed well was Random Forest (F1-score 0.90, ROC-AUC 0.9833). A further improvement was achieved using deep learning based on pre-trained transformer models, i.e., BERT-base, and RoBERTa-base, with a higher accuracy of 0.9756, an ideal recall of class 1 (100

per cent) and an F1-score (0.9767), and a higher precision (100 per cent) and F1-score (0.9744) respectively. On the whole, models based on transformers showed better results in comparison to traditional ML classifiers and proved to be more effective in analyzing the fine nuances of the data.

The presented Quantum ML (QML) model has shown good results in comparison with the classical and transformer-based baselines. Although classical models, such as the Logistic Regression and SVM, performed well (with an accuracy of about 0.88), and ensemble-based methods, such as the Random Forest and XGBoost had better performance (with up to 0.93), the QML approach has the best performance, as it is flawless and provides the complete performance in detecting sensitive work, not missing any positive samples. The overall accuracy (0.976) and the F1-score of the transformer models (BERT-base and RoBERTa-base) are slightly higher because of the deep understanding of the context, but the overall accuracy of quantum circuits is as low as 0.854, and the F1-score is also not so low (0.864), which is why quantum circuits can already find meaningful patterns in textual data even with a compact 4-qubit circuit. In general, it can be concluded that the proposed QML method offers a competitive and innovative solution, which combines interpretability, high recall, and further scalability in quantum resources.

3.5 Limitations of the Implementation Approach

The given parameterized quantum circuit (PQC) model, despite having competitive performance in binary text classification, has a number of weaknesses that need to be considered. To start with, the model is limited by the small number of qubits that are currently accessible on the quantum simulators and hardware. This caused high-dimensional sentence representations retrieved with the MiniLM (all-MiniLM-L6-v2) encoder to be downsampled to 4 principal components using PCA, which unavoidably causes loss of information and can decrease performance in classification compared to more expressive classical models. Moreover, the experiments were carried out on a classical quantum simulator fully because of hardware limitations, instead of on quantum hardware. Thus, the effect of quantum noise, gate errors, decoherence behavior, and finite circuit depth does not manifest itself in the results, and the performance of real-world devices can be vastly different.

The second weakness is due to the small feature space and the rather shallow variational circuit, which has only two trainable layers and eight parameters. Although this can increase the efficiency of computations, it limits representational power relative to more profound variational designs. Also, the training algorithm is based on a simple gradient descent optimizer, no hyperparameter optimization or elaborate optimization algorithms like Adam, SPSA, or natural gradients, which can compromise the quality of the convergence. They divided the data into two using a 75:25 ratio that was not stratified or analysed by variance, which could result in bias depending on the sample distribution. Lastly, the direct comparison between the PQC model and large-scale pretrained transformer models, including BERT and RoBERTa is biased in fairness because the

transformer models are trained on billions of parameters and large pretraining corpora, and the PQC is only trained using task-specific data. The future work ought to take into account analysis of bigger datasets, experimentation on actual quantum processors, examining more intricate architectures, enhancing optimization, and incorporating hybrid quantum-classical layers in order to obtain greater capacity to represent.

Despite the fact that the dataset employed in the present study offers a very wide background upon which the performance of the classification can be measured, one must admit that there are a number of limitations. To start with, the dataset is formed on two different sources: real scientific abstracts retrieved in IEEE Xplore and ScienceDirect and artificial abstracts generated by ChatGPT. Although this facilitates a controlled comparison of human-written and the machine-written content, two scholarly databases can perhaps restrict variety and representativeness of authentic scientific writing patterns, disciplinary framework, and publication criteria. Moreover, despite the attempt to cover a variety of fields, including Artificial Intelligence, Bioinformatics, Blockchain, Cryptography, Internet of Things, Robotics and Wireless Communication, the sample might still be skewed towards technologically oriented areas, limiting its use to other research areas, including social sciences, humanities, or medical research.

Another limitation arises from the creation process of AI-generated abstracts. Since all synthetic examples were produced using ChatGPT based on standardized prompting instructions, the generated text may exhibit stylistic similarities or structural patterns that simplify classification, potentially overestimating model performance. Furthermore, artificial data creation through a single language model introduces systematic bias, as linguistic characteristics are tied to the behavior and training corpus of ChatGPT. Finally, the dataset maintains a strictly balanced distribution between authentic and generated abstracts, which may not reflect real-world scenarios where class distributions are often imbalanced. Future work should explore incorporating real quantum-hardware noise, diversifying domain coverage, and expanding sources to include additional publishers and generative models to enhance dataset robustness and generalizability.

4 Conclusion and Future Works

In this work, we addressed the critical challenge of detecting AI-generated scientific abstracts. We developed a large, domain-diverse dataset of large amount of samples and evaluated a Quantum Machine Learning model based on a parameterized quantum circuit. Despite extreme feature compression and NISQ-era hardware constraints, the QML model achieved promising performance. Additionally, we provided thorough comparisons against classical machine learning and transformer-based baselines to contextualize QML's capabilities.

In future work, we aim to significantly expand and diversify the dataset by incorporating AI-generated abstracts from multiple large language models—including Gemini, Claude, Llama, Copilot, and emerging systems—as well as adding more human-written abstracts sourced from platforms such as ACM Digital Library

and PubMed. We also plan to enhance the Quantum Machine Learning (QML) architecture by integrating quantum feature maps, employing data re-uploading strategies, and exploring deeper variational circuits and quantum neural networks (QNNs). Additionally, we intend to investigate hybrid QML–transformer models to better leverage the strengths of both paradigms. Beyond architectural improvements, we will assess model robustness by evaluating performance on paraphrased or adversarially modified texts and by analyzing cross-domain and cross-LLM generalization. Finally, as quantum hardware matures and offers higher qubit counts, we plan to deploy and benchmark our model on real quantum devices to further validate its practical feasibility.

References

1. Abburi, H., Roy, K., Suesserman, M., Pudota, N., Veeramani, B., Bowen, E., Bhattacharya, S.: A simple yet efficient ensemble approach for AI-generated text detection. In: EMNLP 2023, p. 413 (2023).
2. Shah, A., Ranka, P., Dedhia, U., Prasad, S., Muni, S., Bhowmick, K.: Detecting and unmasking AI-generated texts through explainable artificial intelligence using stylistic features. Int. J. Adv. Comput. Sci. Appl. 14(10) (2023).
3. Yan, D., Fauss, M., Hao, J., Cui, W.: Detection of AI-generated essays in writing assessments. Psychol. Test Assess. Model. 65(1), 125–144 (2023).
4. Mitchell, E., Lee, Y., Khazatsky, A., Manning, C.D., Finn, C.: DetectGPT: zero-shot machine-generated text detection using probability curvature. In: ICML 2023, pp. 24950–24962 (2023).
5. Yadagiri, A., Shree, L., Parween, S., Raj, A., Maurya, S., Pakray, P.: Detecting AI-generated text with pre-trained models using linguistic features. In: ICON 2024, pp. 188–196 (2024).
6. Zhang, Y., Leng, Q., Zhu, M., Ding, R., Wu, Y., Song, J., Gong, Y.: Enhancing text authenticity: A novel hybrid approach for AI-generated text detection. In: ICETCI 2024, pp. 433–438. IEEE (2024).
7. Qazi, Z., Shiao, W., Papalexakis, E.E.: GPT-generated text detection: benchmark dataset and tensor-based detection method. In: The Web Conference 2024 Companion, pp. 842–846 (2024).
8. Gryka P., Gradon K., Kozlowski M., Kutyla M., Janicki A.: Detection of AI-generated emails: A case study. In: ARES 2024, pp. 1–8 (2024).
9. Zhou, Y., Wang, J.: Detecting AI-generated texts in cross-domains. In: DocEng 2024, pp. 1–4 (2024).
10. Guo, X., He, Y., Zhang, S., Zhang, T., Feng, W., Huang, H., Ma, C.: Detective: Detecting AI-generated text via multi-level contrastive learning. Adv. Neural Inf. Process. Syst. 37, 88320–88347 (2024).
11. Wu, J., Zhan, R., Wong, D., Yang, S., Yang, X., Yuan, Y., Chao, L.: DetectRL: Benchmarking LLM-generated text detection in real-world scenarios. Adv. Neural Inf. Process. Syst. 37, 100369–100401 (2024).
12. Guo, H., Cheng, S., Jin, X., Zhang, Z., Zhang, K., Tao, G., Shen, G., Zhang, X.: BiScope: AI-generated text detection by checking memorization of preceding tokens. Adv. Neural Inf. Process. Syst. 37, 104065–104090 (2024).
13. Wang, H., Li, J., Li, Z.: AI-generated text detection and classification based on BERT deep learning algorithm. Theor. Nat. Sci. 39, 311–316 (2024).

14. Yu, P., Chen, J., Feng, X., Xia, Z.: CHEAT: A large-scale dataset for detecting ChatGPT-written abstracts. IEEE Trans. Big Data (2025).
15. Pudasaini, S., Miralles, L., Lillis, D., Salvador, M.L.: Benchmarking AI text detection: assessing detectors against new datasets, evasion tactics, and enhanced LLMs. In: GenAIDetect Workshop 2025, pp. 68–77 (2025).
16. McGovern, H., Stureborg, R., Suhara, Y., Alikaniotis, D.: Your large language models are leaving fingerprints. In: GenAIDetect 2025, p. 85 (2025).
17. Varadarajan, V., Giorgi, S., Mangalik, S., Soni, N., Markowitz, D.M., Schwartz, H.A.: The consistent lack of variance of psychological factors expressed by LLMs and spambots. In: GenAIDetect 2025, pp. 111–119 (2025).
18. Valdez-Valenzuela, A., Gómez-Adorno, H., Montes-y-Gómez, M.: Text graph neural networks for detecting AI-generated content. In: GenAIDetect 2025, pp. 134–139 (2025).
19. Abbas, H.M.: A novel approach to automated detection of AI-generated text. J. Al-Qadisiyah Comput. Sci. Math. 17(1), 1–17 (2025).
20. Oghaz, M., Saheer, L.B., Dhame, K., Singaram, G.: Detection and classification of ChatGPT-generated content using deep transformer models. Front. Artif. Intell. 8, 1458707 (2025). https://doi.org/10.3389/frai.2025.1458707
21. Alhijawi, B., Jarrar, R., AbuAlRub, A., Bader, A.: Deep learning detection method for large language models-generated scientific content. Neural Comput. Appl. 37(1), 91–104 (2025).
22. Sentence-Transformers: all-MiniLM-L6-v2. https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2. Accessed 3 May 2025.
23. MdSiamAnsary/ClassificationTasks. https://github.com/MdSiamAnsary/ClassificationTasks. Accessed 25 November 2025.