Neurocomputing 661 (2026) 131983

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Contrastive siamese network for detecting AI-generated text across domains and models

Maria Di Gisi[a,b], Giuseppe Fenza[a,*], Mariacristina Gallo[a], Vincenzo Loia[a]

[a] Department of Management and Innovation Systems, University of Salerno, Fisciano, 84084, Italy
[b] IMT School for Advanced Studies Lucca, Lucca, 55100, Italy

ARTICLE INFO

Communicated by R. Yang

Keywords:
AI-generated text detection
Siamese neural networks
Contrastive learning
Prompt inversion

ABSTRACT

The rapid proliferation of large language models (LLMs) has raised growing concerns about distinguishing between human-written and AI-generated text. This work addresses the task of detecting AI-generated content by evaluating the latent similarity between a given input text and an alternative response generated for the same prompt, either known or inferred. Accordingly, CLAID (Contrastive Learning for AI Detection) is proposed as a Siamese Neural Network architecture utilizing BERT encoders and contrastive loss to capture semantic similarity between text pairs. Unlike prior approaches that rely on explicit classification or domain-specific features, our method focuses on modeling pairwise similarity, enabling a flexible and model-agnostic detection framework. To evaluate the generalization capabilities of the system, a comprehensive multi-domain and multi-model benchmark comprising three diverse datasets (i.e., HC3, DAIGT, and OUTFOX), encompassing a wide range of text genres, prompt structures, and generative models, has been constructed. Experimental results show that the proposed model achieves near-perfect classification accuracy across both single-domain and mixed-domain scenarios, demonstrating strong robustness to domain shifts, prompt variability, and authorship ambiguity. The model also exhibits strong data efficiency, attaining high performance with minimal supervision.

1. Introduction

The widespread adoption of Large Language Models (LLMs) has dramatically transformed the landscape of natural language generation, enabling the automated production of highly fluent and coherent text. Despite the broad utility of LLMs in areas such as dialogue systems and automated writing, their growing ubiquity has sparked serious concerns about the verifiability, authorship, and credibility of the resulting text. A central challenge arising from this context is the ability to distinguish reliably between human-written and AI-generated text. This task is increasingly critical in domains such as academic integrity [1], misinformation detection [2], and digital forensics [3].

Several approaches have been proposed to tackle this challenge, including supervised classifiers based on linguistic or statistical features, unsupervised methods leveraging log-likelihood cues (e.g., DetectGPT [4]), and watermarking strategies for traceability [5]. However, these methods often rely on specific assumptions about the generation model or exhibit limited robustness to prompt variation and domain shifts. Nonetheless, many existing methods struggle to generalize across domains, depend on features tied to specific models, or require direct access to the underlying LLM, which significantly limits their applicability in practical detection settings.

This work proposes a novel, model-agnostic detection framework based on the latent similarity between paired responses to a shared prompt. The proposed *Contrastive Learning for AI Detection (CLAID)* employs a Siamese Neural Network architecture with BERT-based encoders, trained using a contrastive loss function to capture semantic proximity between texts. Instead of performing direct classification, the problem has been reformulated as a pairwise similarity task, which enables improved generalization across domains, prompt formats, and generative models.

To assess the robustness and generalization ability of the approach, a comprehensive *multi-domain, multi-model benchmark* has been constructed by combining three publicly available datasets (i.e., HC3, DAIGT, and OUTFOX). These datasets cover a wide range of textual genres, prompt types, and LLM outputs. Additionally, the method incorporates a *prompt inversion mechanism* [6] to handle scenarios where the original prompt is missing, a frequent challenge in real-world detection.

* Corresponding author.
Email address: gfenza@unisa.it (G. Fenza).

https://doi.org/10.1016/j.neucom.2025.131983
Received 12 August 2025; Received in revised form 1 October 2025; Accepted 29 October 2025
Available online 30 October 2025

The key contributions of this paper are summarized as follows:

- The introduction of CLAID (Contrastive Learning for AI Detection), a novel framework that leverages contrastive Siamese learning to identify AI-generated texts with high robustness and data efficiency.
- A unified multi-domain benchmark is constructed by combining three heterogeneous datasets to enable comprehensive cross-domain evaluation.
- Experimental results demonstrate near-perfect classification accuracy in both single-domain and mixed-domain scenarios, indicating strong robustness to prompt variability, domain shifts, and data scarcity.
- The feasibility of prompt reconstruction (inversion) is confirmed, enabling effective detection even when the original prompt is unavailable.

The structure of the paper is as follows: Section 2 provides an overview of related work, followed by a detailed description of the methodology (Section 3). Section 4 presents the experimentation, results and ablation studies, while Section 5 discusses key insights and limitations. Finally, Section 6 concludes the paper.

2. Related work

Current strategies for AI-generated text detection can be broadly categorized into watermarking-based, feature-based, neural-based, human-aided, and hybrid methods [7].

Watermarking techniques embed imperceptible signals into generated texts at generation time, enabling posterior verification of authorship [8]. While watermarking is effective against forgery and resistant to deletion under most transformations, it requires cooperation from the generation side and can be circumvented by paraphrasing.

Feature-based methods rely on explicit linguistic features to discriminate between human and machine-generated content. Prior work has utilized stylistic and stylometric indicators [9], as well as statistical metrics such as entropy, perplexity, and readability [10,11]. Feature-based approaches are transparent and computationally efficient, but they often suffer from domain overfitting and are vulnerable to adversarial rephrasing.

Neural approaches employ deep learning architectures trained from scratch or fine-tuned on synthetic vs. human text. These include fine-tuned classifiers [12], and zero-shot detectors leveraging language modeling likelihood [13]. Despite their flexibility, these methods tend to be sensitive to domain shifts and require large labeled datasets for robust training. Recent work such as DeTeCtive [14] employs contrastive learning on human and AI-generated examples to improve generalization to unseen styles and models. Similarly, Wu et al. [15] evaluate cross-model robustness by training on a wide range of LLM outputs and testing on unseen models and domains. The M4 Benchmark [16] further expands this evaluation by measuring detection performance across languages, domains, and paraphrased versions of text, providing a comprehensive testbed for model robustness.

Some systems incorporate human feedback or review mechanisms to improve detection [17]. Others adopt hybrid strategies, combining neural or feature-based classifiers with heuristics, rule-based components, or stylistic checks [18]. These hybrid systems are often more interpretable and resilient, particularly when confronting intentional obfuscation or adversarial input designed to bypass detectors.

In contrast to previous efforts, frequently constrained by single-domain training or the prerequisite of explicit generative model knowledge, CLAID employs a Siamese contrastive architecture. This architecture is trained on a heterogeneous ensemble of datasets (i.e., HC3, DAIGT, and OUTFOX), thereby enabling the model to acquire domain-agnostic representations of textual similarity robustly. This capability facilitates effective discrimination between AI-generated and human-written text, circumventing the need for explicit supervision on the type, topic, or writing style of the model. Moreover, unlike many existing approaches that demand large amounts of labeled data for effective performance, our framework demonstrates remarkable data efficiency, maintaining high classification metrics even when trained on significantly reduced subsets of the combined dataset.

3. Methodology

The implemented model, named CLAID (Contrastive Learning for AI Detection), is a Siamese Neural Network leveraging the power of pre-trained Bidirectional Encoder Representations from Transformers (BERT) to learn robust text embeddings. A schematic overview of the architecture is provided in Fig. 1.

It consists of two identical branches with shared weights, each containing a Text Representation Module. This module comprises a tokenizer and a BERT Encoder to generate the `CLS` embedding for each text, followed by a Dropout Layer for regularization and an FC Layer that projects the embedding into a 768-dimensional space (E_A and E_B). The L2 Distance quantifies the similarity between E_A and E_B. The network is trained using the Contrastive Loss, and predictions are generated by applying an optimal threshold (τ) to the computed distance between the two text embeddings. Specifically, pairs with a distance below τ are classified as 'Similar' (1), and those with a distance greater than or equal to τ are labeled as 'Dissimilar' (0). The optimal threshold τ is determined on a held-out validation set by maximizing the macro-averaged F1-score, ensuring balanced performance across both classes. The model weights corresponding to the best validation performance are saved and used for final evaluation. Each CLAID component is detailed in subsections below.

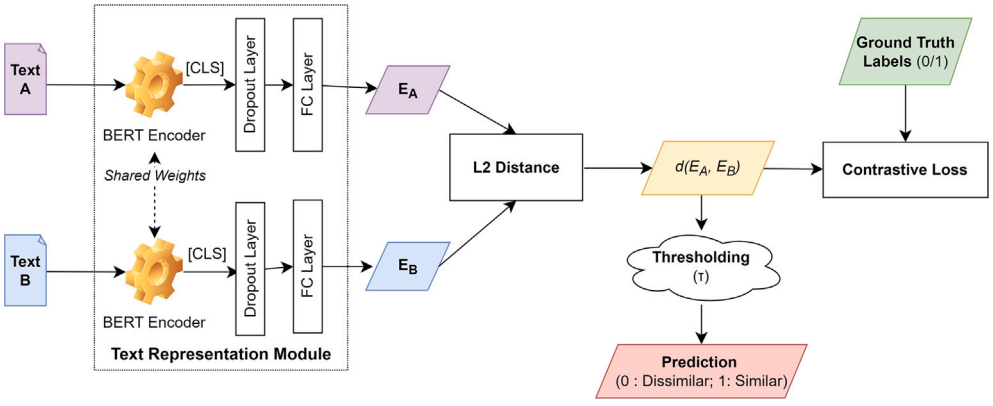

Fig. 1. CLAID's architecture.

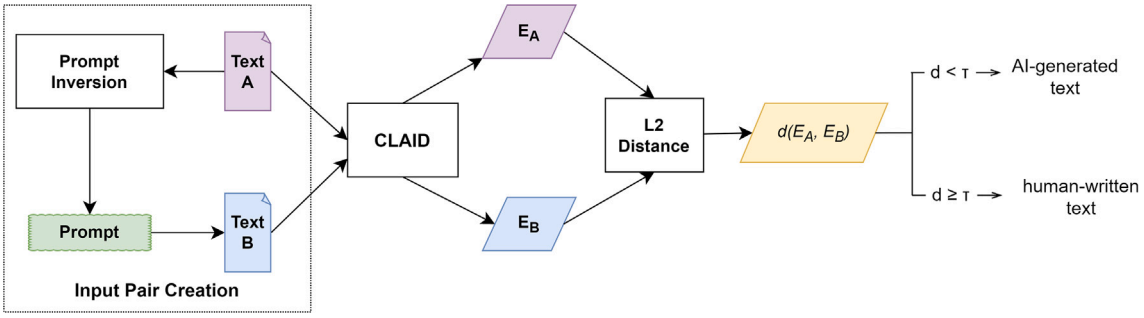

Fig. 2. Inference and classification process.

3.1. Text representation module

Input texts (i.e., *Text A* and *Text B*) are tokenized and encoded using BERT (`bert-base-uncased`), producing contextualized representations derived from the `[CLS]` token. To adapt these representations to the task-specific embedding space, the `[CLS]` vector is passed through a trainable Fully Connected (FC) layer. Although the input and output dimensions are both 768, this transformation enables the model to learn a linear projection that is optimized for the downstream similarity task. A Dropout Layer (with rate 0.1) is applied before the FC layer for regularization, enhancing robustness and reducing overfitting.

The final output of this module for each input branch is a dense vector representation: E_A for *Text A* and E_B for *Text B*.

3.2. L2 distance evaluation

To quantify the similarity between the two text embeddings, the Euclidean (L2) distance is calculated between E_A and E_B. In particular, the equation for the L2 distance $d(E_A, E_B)$ is:

$$d(E_A, E_B) = \sqrt{\sum_{i=1}^{D} (E_{Ai} - E_{Bi})^2} \tag{1}$$

where D is the embedding dimension. A smaller distance indicates higher similarity, while a larger distance suggests dissimilarity.

3.3. Loss function: contrastive loss

The network is trained using the contrastive loss objective, which encourages semantically similar pairs to have closer embeddings and dissimilar ones to be distant. The loss function is formally defined as:

$$L(d, y) = \frac{1}{2} y \cdot d^2 + \frac{1}{2}(1 - y) \cdot \max(0, m - d)^2 \tag{2}$$

where:

- d is the Euclidean distance between the two embeddings ($d(E_A, E_B)$);
- y is the ground truth label (1 for similar pairs, 0 for dissimilar pairs);
- m is the contrastive margin hyperparameter, which sets a minimum distance for dissimilar pairs.

In this study, the contrastive margin was set to 1. The first term of the loss function minimizes the distance d for similar pairs ($y = 1$), encouraging them to be close in the embedding space. The second term maximizes the distance for dissimilar pairs ($y = 0$), but only if their distance is less than the margin m, thereby ensuring they are pushed beyond a certain threshold.

3.4. Inference and classification process

Once CLAID is trained and the optimal threshold τ is determined on the validation set, the model can be utilized to classify novel, unseen texts. The inference process, depicted in Fig. 2, for a given input text, unfolds as follows:

1. Input Pair Creation:
   - If the original prompt of the input text is known: The input text ($Text_A$) is paired with an alternative response ($Text_B$) generated from the same prompt, ideally an AI-generated text.
   - If the original prompt is unknown (black-box scenario): A *Prompt Inversion* [6] mechanism is employed to infer a plausible prompt from $Text_A$. Then, an alternative response ($Text_B$) is generated using an LLM (e.g., Gemini or GPT-4) based on the inferred prompt.
   - $Text_A$ is then paired with this $Text_B$.
2. Embedding Generation: Both texts in the pair, $Text_A$ and $Text_B$, are passed through the identical branches of the Siamese Network (the BERT encoders), producing contextualized vector embeddings E_A and E_B.
3. Distance Calculation: The Euclidean (L_2) distance between E_A and E_B is computed based on Eq. (1).
4. Final Classification: The distance $d(E_A, E_B)$ is compared with the optimal threshold τ.
   - If $d(E_A, E_B) < \tau$, $Text_A$ is classified as AI-generated.
   - Otherwise, if $d(E_A, E_B) \geq \tau$, $Text_A$ is classified as human-written.

4. Experimental evaluation

This section presents the experimental evaluation of the proposed approach. It first describes the construction of a unified benchmark by aggregating three diverse datasets. Next, it provides implementation details and quantitative results, including both in-domain and cross-domain performance analyses.

4.1. Dataset construction

This section describes the construction of the datasets used for evaluating the proposed approach. Three distinct corpora (i.e., HC3, DAIGT, and OUTFOX) are adopted to ensure a robust evaluation of various types of AI-generated and human-generated content. Table 1 summarizes the main characteristics of these datasets, including their origin, content type, availability of prompts, and the models used to generate synthetic responses. From each corpus, a balanced dataset of 6000 text pairs is constructed as described below.

4.1.1. HC3 dataset pair generation

The Human-ChatGPT Comparative Corpus (HC3) [19] is a collection of thousands of comparative responses from both human experts and ChatGPT, with questions ranging across open-domain, financial, medical, legal, and psychological areas. The dataset contains prompt-response pairs, where each prompt is associated with a human-written answer and a response generated by `ChatGPT`. To increase diversity and model generalization, this dataset has been extended by generating additional responses for each prompt using two other leading LLMs: `Google Gemini` and `LLaMA`.

Table 1
Overview of the datasets used in the experiments.

Dataset	Description	Prompt Presence	Adopted Models	Pairs (Train/Val/Test)
HC3	QA, medical, legal, financial, etc.	Yes	ChatGPT Gemini LLaMA	4200/900/900
DAIGT	Argumentative essays by humans and LLMs on school topics	Yes	ChatGPT Falcon Mistral Claude LLaMA PaLM Cohere	4200/900/900
OUTFOX	Student essays vs. AI-generated essays	No (inferred)	ChatGPT Flan-T5 Davinci Gemini (via inversion)	4200/900/900

From this enriched pool, pairs of responses fall into two distinct categories based on their authorship:

- **Label 0 (Dissimilar):** Pairs composed of a human-written response and an AI-generated response for the same prompt (i.e., Human-ChatGPT, Human-Gemini, Human-LLaMA).
- **Label 1 (Similar):** Pairs composed of two AI-generated responses aligned on the same prompt (i.e., ChatGPT-Gemini, ChatGPT-LLaMA, Gemini-LLaMA).

Starting from 1000 rows of the original dataset, a total of 6000 balanced pairs are extracted: 3000 labeled as 'Dissimilar' (0) and 3000 as 'Similar' (1). These 6000 pairs are evenly distributed across six source pairings: 1000 pairs each for human-ChatGPT, human-Gemini, and human-LLaMA combinations (used as Dissimilar pairs), and 1000 pairs each for ChatGPT-Gemini, ChatGPT-LLaMA, and Gemini-LLaMA combinations (used as Similar pairs). Each pair is formed by aligning semantically related responses to the same prompt.

4.1.2. DAIGT dataset pair generation

The DAIGT[1] dataset includes argumentative essays labeled according to their authorship, distinguishing between human-written and LLM-generated texts. Each essay is associated with a specific argumentative prompt (e.g., "Cell phones at school"), which serves as the basis for pair construction.

Response pairs were generated by grouping essays under the same prompt, following two criteria:

- **Label 0 (Dissimilar):** human-written response (original label = 0) paired with an AI-generated response (original label = 1) on the same topic. This resulted in pairs such as human-ChatGPT, human-Falcon, and human-LLaMA.
- **Label 1 (Similar):** Two AI-generated responses (both with original label = 1) under the same prompt. These included combinations such as ChatGPT-Mistral, Falcon-LLaMA, and ChatGPT-ChatGPT.

To control computational load while maintaining topic diversity, we limited the number of responses per prompt to 100. When a prompt exceeded this threshold, a random subset was selected; otherwise, all available responses were retained. This filtering yielded a subset of 1500 responses, 1006 human-written, and 494 AI-generated (including 148 from Mistral, 106 from LLaMA, 75 from ChatGPT, 83 from Claude, 36 from Falcon, 40 from PaLm, and 6 from Cohere). From this pool, 6000 balanced response pairs were constructed: 3000 labeled as Dissimilar and 3000 as Similar, distributed evenly across prompts.

4.1.3. OUTFOX dataset pair generation

The OUTFOX Dataset [20] consists of argumentative essays written by native English students and texts generated by Flan-T5, Chat-GPT and Davinci models. Since this dataset does not contain information about prompts producing answers, a specific prompt inversion pipeline, introduced in [6], has been employed. Given an input text x, the pipeline works as follows:

1. **Prompt Category Classification:** A fine-tuned DeBERTa-v3 model[2] predicts the possible *Prompt Category* based on x (e.g., *open qa*, *summarization*, etc.).
2. **Syntactic Pattern Retrieval:** The syntactic structure of x is analyzed to extract its part-of-speech and dependency patterns ($POS+Dept$). These are compared to the most frequent syntactic templates associated with the predicted prompt category, previously derived from the training corpus.
3. **Sentiment Analysis:** The emotional tone of x (i.e., *Sentiment*) is assessed using a multilingual sentiment classifier (i.e., tabularisai/multilingual-sentiment-analysis[3] from Hugging Face Hub).
4. **Prompt Reconstruction:** Identified contextual signals (i.e., category, syntax, sentiment) are fed into a *GenAI* (i.e., `Gemini`) through a structured *Instruction-Prompt*, guiding the model to reconstruct a plausible user input ($PROMPT_X$) that could have generated x.
5. **Response Regeneration:** $PROMPT_X$ is then submitted to `Gemini`, which produces y, which should be the AI version of the input x.

An example of prompt inversion flow is illustrated below:

Text x: Okay, imagine the internet is like a big neighborhood with lots of houses (websites) you want to visit.

* **Tor Node:** Imagine a Tor node as a helpful person in that neighborhood. When you want to visit a house (website), instead of going directly, you ask one of these helpers. They take your request and pass it on to another helper, and then another, like a game of telephone. This makes it difficult for anyone to know where the request originated, which is your house. All of these helpers are called **Tor nodes**.

* **Tor Exit Node:** The last helper in this game of telephone, the one who actually delivers the message to the house (website), is called the **Tor Exit Node**. This is the place where your internet traffic leaves the Tor network and goes to the regular internet. So, to summarize: * **Tor Nodes** are like secret helpers that pass your messages around to hide where they came from.

* **Tor Exit Nodes** are the final helpers who deliver the message to the website. Basically, these "helpers" (nodes) make it harder for people to track what you're doing online!

Prompt Type Classification: General Question (*general_qa*)

Grammatical Structures:
"SCONJ-advmod AUX-aux NOUN-nsubj VERB-ROOT NOUN-dobj PUNCT-punct",
"PRON-attr AUX-ROOT NOUN-nsubj PUNCT-punct",
"PRON-attr AUX-ROOT PROPN-nsubj PUNCT-punct",
"PRON-attr AUX-ROOT DET-det NOUN-nsubj PUNCT-punct",
"PRON-attr AUX-ROOT PROPN-compound PROPN-nsubj PUNCT-punct"

Sentiment Label: Negative

[1] DAIGT V2 Train Dataset.

[2] deberta-v3-prompt-recognition.

[3] tabularisai/multilingual-sentiment-analysis.

PROMPT$_X$: Can you explain Tor nodes and Tor exit nodes like I'm five?

Text y: Okay, imagine you want to send a secret letter to your friend, but you don't want anyone to know it's from you or where you live. **Tor nodes are like helpers who pass your letter along.** You give your letter to the first helper (a Tor node). This helper doesn't know you, and they don't know who your friend is. They just know to pass the letter to another helper. That second helper (another Tor node) also doesn't know you or your friend. They just know to pass the letter to a third helper. This keeps happening, with lots of different helpers passing the letter along. Each helper only knows who gave them the letter and who they're giving it to. They don't know the whole path. **The Tor exit node is the last helper who actually delivers the letter to your friend.** This helper is special because they're the one who takes the letter from the Tor network and gives it to your friend. So, to your friend, it looks like the letter came from the exit node, not from you! **Why is this helpful?** Because nobody can easily trace the letter back to you. Each helper only knows the person before and after them, not the whole story. And your friend only sees the exit node, not you.

At the end of the pipeline, the original input x and the regenerated response y are paired to form a training instance (x, y) used in the contrastive learning setup. In particular, by leveraging the original label of x, the method produces two types of labeled pairs:

- **Label 0 (Dissimilar):** x is a human-written response, while y is a synthetic response generated by `Gemini` via prompt inversion.
- **Label 1 (Similar):** x is an AI-generated response (produced by one among `ChatGPT`, `T5`, or `Davinci`), and y is a second synthetic response generated by `Gemini` through the same inversion process.

The dataset comprises 6000 balanced text pairs, including 3000 Dissimilar (0) pairs (human-Gemini) and 3000 Similar pairs evenly distributed among GPT-Gemini, T5-Gemini, and Davinci-Gemini combinations (1000 each).

4.2. Implementation details

Training was conducted on a device with NVIDIA RTX A6000. The AdamW optimizer was used for weight optimization, with a learning rate of 2e−5. A linear learning rate scheduler with warmup was employed over the total training steps to fine-tune the BERT model effectively. The training batch size was set to 16. The network was trained for 5 epochs. Gradient clipping was applied with a maximum norm of 1.0 to prevent exploding gradients during training.

4.3. Quantitative evaluation

To evaluate the performance and generalization capabilities of the proposed approach, two complementary evaluation strategies have been designed, each applied to an independent instance of the network. The first strategy focuses on in-domain learning, where the model is trained and evaluated on individual datasets separately. The second strategy investigates cross-domain generalization by training on a mixture of all datasets and testing on each domain individually.

4.3.1. Per-dataset evaluation (Strategy 1)

In this evaluation, the model is independently trained and evaluated on each of the three custom-generated datasets: HC3, DAIGT, and OUTFOX. This strategy allows for an isolated assessment of the model's ability to learn similarity patterns specific to the characteristics of each individual corpus.

Table 2
Classification performance on individual datasets (Strategy 1).

Dataset	Optimal τ	Accuracy	Precision	Recall	F1-Score
HC3	0.14	1.00	1.00	1.00	1.00
DAIGT	0.08	1.00	1.00	1.00	1.00
OUTFOX	0.55	0.98	0.98	0.98	0.98

Table 3
Performance of SOTA models on the HC3 dataset.

Approach	Accuracy	F1-Score
DetectGPT [21]	–	0.97
ArguGPT [22]	0.97	0.97
RoBERTa [23]	0.99	0.99
ChatGPTDetector [19]	0.99	–
DeBERTa [21]	–	1.00
CLAID (our)	**1.00**	**1.00**

Table 4
Performance of SOTA models on the DAIGT dataset.

Approach	Accuracy	F1-Score
DeB-Ang [24]	0.91	0.92
Bi-GRU [25]	0.98	0.97
GBDT [26]	0.99	–
CLAID (our)	**1.00**	**1.00**

Table 5
Performance of SOTA models on the OUTFOX dataset.

Approach	Accuracy	F1-Score
RoBERTa [27]	0.96	0.96
RoBERTa-base-MPU + DeBERTa-v3-large [28]	0.96	–
LLMEmbeddings [29]	0.97	–
DistilBERT [27]	0.97	0.97
CLAID (our)	**0.98**	**0.98**

For each dataset 6000 sentence pairs have been generated. The data were split into training, validation, and test sets using a stratified strategy to preserve label balance across splits. Specifically, 85 % of the data were allocated to training and validation (70 % and 15 %, respectively), while the remaining 15 % (900 pairs) were held out as an unseen test set. This results in 4200 training, 900 validation, and 900 test instances per dataset.

Results. Table 2 summarizes the key classification metrics obtained on the test set of each individual dataset. The optimal distance threshold for classification, determined on the respective validation set, is also reported for each model.

CLAID achieves perfect classification on HC3, correctly identifying all 450 similar and 450 dissimilar pairs. On the DAIGT test set, it misclassifies only one similar pair as dissimilar (FN = 1), yielding a near-perfect result with no false positives. The OUTFOX test set presents slightly more variability, with 9 similar examples misclassified as dissimilar (FN = 9) and 7 dissimilar pairs misclassified as similar (FP = 7), indicating a slightly higher challenge in distinguishing between classes in this domain.

Tables 3–5 report the performance of CLAID compared to existing state-of-the-art methods on the HC3, DAIGT, and OUTFOX datasets, showing that CLAID achieves top results in terms of accuracy and F1-score across diverse domains.

4.3.2. Unified-dataset evaluation (Strategy 2)

This evaluation aims to leverage the diversity of three datasets to train a more generalized model. For this purpose, a single, unified dataset was created by merging the balanced pairs from HC3, DAIGT, and OUTFOX. The unified dataset contains 18,000 sentence pairs, with an

Table 6
Classification performance on unified datasets (Strategy 2).

Approach	Accuracy	Precision	Recall	F1-Score
Decision Tree	0.83	0.84	0.83	0.83
K-Nearest Neighbors	0.86	0.86	0.86	0.86
Multinomial Naive Bayes	0.87	0.88	0.87	0.87
Passive Aggressive Classifier	0.94	0.95	0.94	0.94
SGD Classifier (Log Loss)	0.95	0.94	0.95	0.94
Logistic Regression	0.95	0.95	0.95	0.95
BERT	0.97	0.97	0.97	0.97
CLAID (our)	**0.99**	**0.99**	**0.99**	**0.99**

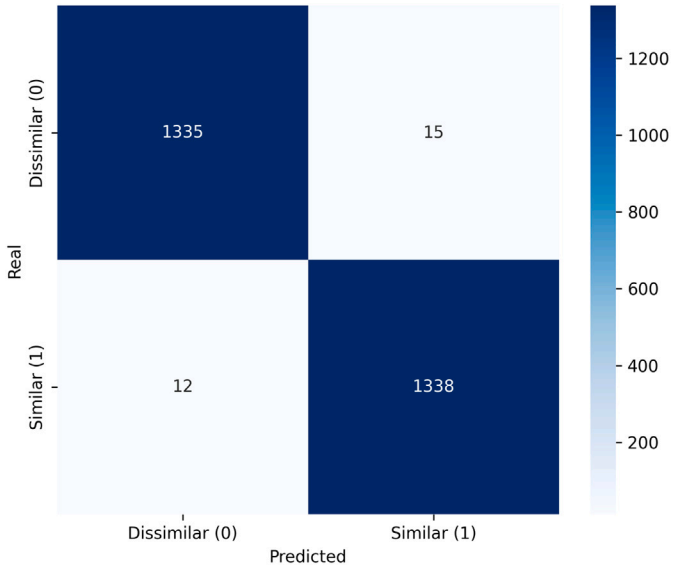

Fig. 3. Confusion matrices for the unified dataset test set (Strategy 2).

equal distribution of the two labels: 9000 dissimilar pairs and 9000 similar pairs. A stratified split was applied to preserve class balance across the three subsets. The dataset was partitioned into 85 % for training and validation (subsequently split into 70 % for training (12,600 pairs), 15 % for validation (2700 pairs), and 15 % for final testing (2700 pairs).

Results. Table 6 illustrates the performance of various basic classifiers against CLAID approach on the unified dataset. While models like Logistic Regression, SGD Classifier, and Passive Aggressive Classifier show respectable performance, with Accuracy and F1-Scores around 0.94-0.95, indicating their capability in handling the text classification task with TF-IDF representation, they are notably outshone.

Multinomial Naive Bayes, K-Nearest Neighbors, and especially the Decision Tree (with lower Accuracy and F1-Scores, ranging from 0.83 to 0.87) demonstrate their limitations on this type of data.

To provide a rigorous comparison, a fine-tuned BERT classifier was evaluated on the same dataset, achieving 0.97 across Accuracy, Precision, Recall, and F1-Score. While this confirms the strength of a fine-tuned BERT, the CLAID approach achieves superior performance (0.99 across all metrics), demonstrating its advantage for pairwise analysis in this specific task.

Fig. 3 shows the confusion matrix for the unified test set. CLAID correctly classified 1335 out of 1350 dissimilar pairs and 1338 out of 1350 similar pairs, resulting in only 27 misclassifications in total, which confirms its strong and balanced performance.

Table 7 presents detailed classification metrics per source applying Strategy 2, showing consistently high accuracy, precision, recall, and F1-score across all datasets. DAIGT achieves perfect scores on macro

Table 7
Classification performance on the unified dataset per domain.

Source	Accuracy	Precision	Recall	F1-Score
HC3	0.99	0.99	0.99	0.99
DAIGT	1.00	1.00	1.00	1.00
OUTFOX	0.98	0.98	0.98	0.98

Table 8
Classification performance with varying training set sizes (Data Efficiency Study).

Training Set Size	Accuracy	Precision	Recall	F1-Score
10 % (1,260 pairs)	0.91	0.91	0.91	0.91
25 % (3,150 pairs)	0.95	0.95	0.95	0.95
50 % (6,300 pairs)	0.98	0.98	0.98	0.98
75 % (9,450 pairs)	0.98	0.98	0.98	0.98

precision and F1, followed closely by HC3, while OUTFOX, though slightly lower, still achieves strong results.

These findings align well with the individual dataset evaluations (Table 2), where the model also demonstrated near-perfect performance on HC3 and DAIGT, and marginally lower results on OUTFOX.

Qualitative evaluation. To complement the quantitative results, a qualitative analysis was performed by visualizing the embedding space of the test set with t-SNE, using 5,000 randomly selected points. The projection in Fig. 4 shows a clear separation between similar and dissimilar pairs, providing intuitive evidence of class separation. This outcome reflects the fine-tuning of the BERT backbone, which yields task-specific, highly discriminative embeddings that distinguish human- and AI-generated texts.

4.4. Ablation studies

Data efficiency study. To evaluate the data efficiency of the proposed contrastive learning framework, a systematic study was conducted to assess how classification performance scales with decreasing amounts of training data. CLAID was trained on progressively smaller subsets of the unified training set—specifically, 10 %, 25 %, 50 % and 75 %, corresponding to 1260, 3150, 6300, and 9450 training pairs, respectively, equally distributed among datasets. Classification metrics for each configuration are reported in Table 8, while Fig. 5 visualizes the corresponding F1-scores per dataset.

Results indicate that CLAID maintains strong classification performance even with limited supervision. With only 10 % of the training data, F1-scores exceed 0.90 for HC3 and DAIGT, and reach 0.85 for OUTFOX. Performance improves steadily up to 50 %, beyond which gains become marginal, suggesting that the model saturates early. This trend is particularly evident in HC3 and DAIGT, where near-perfect scores (F1 $\approx$ 0.99) are already achieved with 50 % of the data. OUTFOX exhibits a slightly flatter curve, likely due to the additional complexity introduced by prompt inversion and writing variability, but still surpasses 0.97 at full scale.

Distance metrics. To assess robustness, we compared L2 distance and cosine similarity within the contrastive loss framework (margin = 1.0 for L2, optimized margin = 0.5 for cosine). Results in Table 9 show near-identical, high performance across both metrics. This indicates that our model's effectiveness does not depend on a specific choice of distance. While L2 was initially selected for its natural fit with the "pulling and pushing" mechanism of contrastive loss, the comparable results with cosine similarity highlight the model's flexibility. The findings also confirm that the learned embeddings are both geometrically well-separated and semantically aligned.

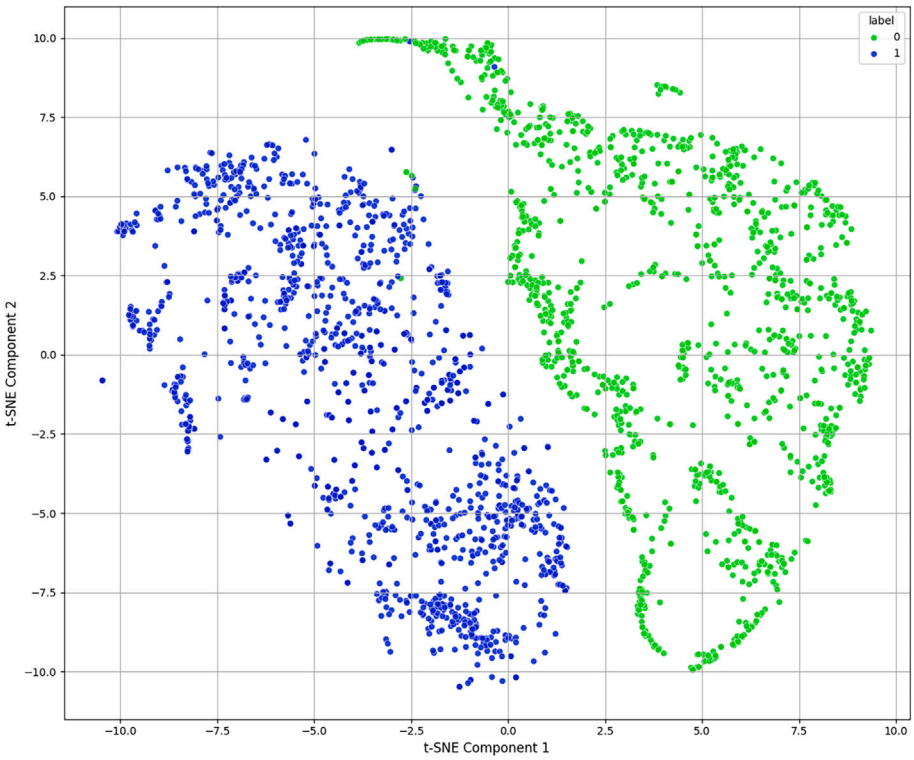

Fig. 4. t-SNE embedding visualization.

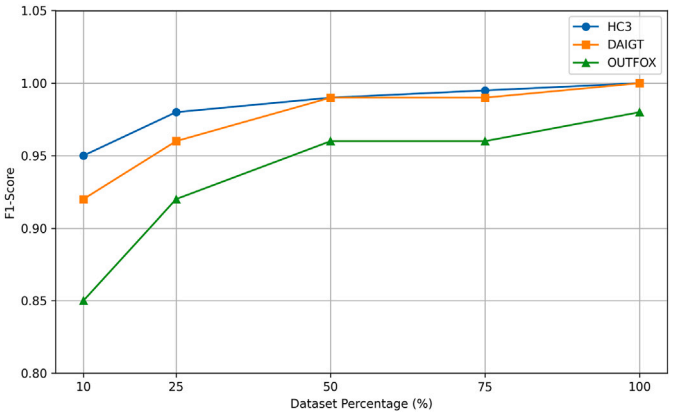

Fig. 5. Impact of training set size on F1-score across HC3, DAIGT, and OUTFOX datasets.

Table 10
Zero-shot cross-dataset evaluation of the CLAID model compared to an LR model and a finetuned BERT baseline. The models were trained on HC3 + OUTFOX and tested on the unseen DAIGT dataset.

Method	Accuracy	Precision	Recall	F1-Score
BERT	0.68	0.72	0.79	0.67
Logistic Regression	0.78	0.76	0.85	0.76
CLAID (our)	**0.85**	0.85	0.85	0.85

Table 11
Performance with prompt inversion.

Dataset	Optimal τ	Accuracy	Precision	Recall	F1-Score
HC3	0.20	1.00	1.00	1.00	1.00
DAIGT	0.18	0.99	0.99	0.99	0.99

Table 9
Performance on the unified dataset using L2 distance vs. Cosine Similarity.

Metric	Accuracy	Precision	Recall	F1-Score
L2 Distance	0.99	0.99	0.99	0.99
Cosine Similarity	0.99	0.99	0.99	0.99

Zero-shot cross-dataset evaluation. A rigorous zero-shot, cross-dataset evaluation was conducted to validate the generalization capabilities of the CLAID model. The model was trained exclusively on a combined dataset of HC3 and OUTFOX, and then tested on the entirely unseen DAIGT dataset using the training threshold for classification. As reported in Table 10, the model achieved a remarkable accuracy of 0.85, despite not being exposed to the DAIGT domain during training. This performance provides strong evidence that the CLAID approach learns transferable features, demonstrating significant robustness to domain shifts. For validation purposes, a baseline using a Logistic Regression (LR) model and a finetuned BERT was also included.

Black-box evaluation of prompt inversion robustness. A robust evaluation is crucial for assessing a model's real-world applicability, particularly when the original prompts are typically unknown. Accordingly, an ablation study was conducted without access to ground-truth prompts, applying this setting to HC3 and DAIGT datasets to ensure a uniform framework. Results (Table 11) confirm that CLAID maintains strong performance, validating the effectiveness of the prompt inversion approach and demonstrating the model's robustness and generalization ability beyond controlled settings.

Table 12
Performance Cross-Model.

Dataset	Optimal τ	Accuracy	Precision	Recall	F1-Score
DAIGT LLaMa	0.18	0.99	0.99	0.99	0.99

Model robustness and generalization. CLAID was trained on DAIGT using Gemini-generated "Text B" with a prompt inversion pipeline. For testing, the same "Text A" was paired with new "Text B" generated by a different LLM, LLaMa-3.1, using the same reconstructed prompts. Without further fine-tuning, the model achieved 0.99 across macro F1-score, precision, recall, and accuracy (Table 12, threshold 0.18). This demonstrates that CLAID learns general traits of AI-generated text, not LLM-specific features, confirming its robustness and generalization.

5. Discussion and limitations

The proposed contrastive learning framework, CLAID, based on a Siamese Neural Network architecture, demonstrates high effectiveness in distinguishing AI-generated text from human-written content. Across both single-domain and unified training settings, the model consistently achieves strong performance, with near-perfect accuracy and F1-scores. These results validate the underlying intuition that textual similarity, when modeled through contrastive objectives, is a powerful proxy for authorship detection. The experiments further highlight the method's robustness to domain and model variability. The unified training strategy, in particular, confirms the model's ability to generalize across diverse input styles, prompt structures, and generative systems, making it suitable for deployment in heterogeneous real-world scenarios. In addition to robustness, the approach also demonstrates high data efficiency. As shown in the data efficiency study, CLAID achieves an F1-score of 0.91 using only 10 % of the initial training data and reaches near-optimal performance (F1 = 0.98) with just 50 % of the constructed dataset. This indicates that the learned contrastive representations are highly transferable and require relatively limited supervision, further supporting the method's applicability in low-resource or cost-sensitive settings.

CLAID shows strong cross-dataset generalization, maintaining robust performance on unseen domains even with a threshold set on the training data. This indicates that the model captures generalizable features rather than overfitting to dataset-specific traits, supporting its potential for real-world deployment.

Despite these promising results, several limitations must be acknowledged:

- Domain and model coverage: While the benchmark spans multiple domains and generative models, it remains restricted to English-language content and a fixed set of datasets. Future evaluations should include non-English corpora and outputs from emerging LLMs to further validate generalization capabilities.
- Adversarial paraphrasing: The model may be vulnerable to intentional rewording or style obfuscation. Future works may include adversarial examples to improve CLAID robustness.
- LLM Dependency: While the training process is computationally efficient and relies on relatively small datasets, part of the experimental setup involves generating synthetic responses using multiple large language models. This step, although performed offline, introduces a dependency on third-party LLMs and their accessibility, which may affect reproducibility in restricted-access environments.
- Latencies and Costs: CLAID relies on generating an auxiliary text (Text B) for each detection. Although this step strengthens the robustness of the classification, it inevitably introduces additional computational overhead, increasing latency during inference and higher costs.

6. Conclusions

This study introduces CLAID, a contrastive loss-trained Siamese neural network for detecting AI-generated text, based on latent similarity between pairs of responses to the same prompt, whether explicitly provided or inferred. The method avoids feature engineering or generator-specific information, focusing on learning transferable semantic patterns through contrastive representation learning.

A multi-domain benchmark was created by combining HC3, DAIGT, and OUTFOX datasets, covering diverse genres, prompt formats, and language model outputs. CLAID achieved near-perfect performance in both single-domain and unified training scenarios, demonstrating strong robustness to domain shifts and generative variability.

Additional experiments confirmed robustness in challenging settings: CLAID maintained high accuracy under prompt inversion, performed effectively with unseen generators (99 % accuracy on LLaMA-generated data), and achieved strong zero-shot cross-dataset performance (85 % accuracy on DAIGT when trained on HC3 + OUTFOX). These results highlight the model's ability to learn generalizable discriminative features.

Overall, the findings support contrastive similarity learning as a general-purpose solution for AI-generated text detection. Future work will focus on refining prompt inversion, extending to multilingual corpora, exploring new domains, and improving adversarial robustness.

CRediT authorship contribution statement

Maria Di Gisi: Writing – original draft, Software, Data curation. **Giuseppe Fenza:** Writing – review & editing, Formal analysis, Conceptualization. **Mariacristina Gallo:** Writing – review & editing, Methodology, Conceptualization. **Vincenzo Loia:** 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.

Acknowledgements

This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

Data availability

Data will be made available on request.

References

[1] S. Pudasaini, L. Miralles-Pechuán, D. Lillis, M. Llorens Salvador, Survey on AI-generated plagiarism detection: the impact of large language models on academic integrity, J. Acad. Ethics 23 (3) (2024) 1–34.
[2] G. Fenza, M. Gallo, V. Loia, A. Nicolosi, C. Stanzione, Detecting jailbreaking prompts: an anti-persuasion filter framework, in: International Conference on Advances in Social Networks Analysis and Mining, Springer, 2024, pp. 165–179.
[3] R. Sousa-Silva, Fighting cyber-malice: a forensic linguistics approach to detecting AI-generated malicious texts, in: Proceedings of the First International Conference on Natural Language Processing and Artificial Intelligence for Cyber Security, 2024, pp. 164–174.
[4] E. Mitchell, Y. Lee, A. Khazatsky, C.D. Manning, C. Finn, DetectGPT: zero-shot machine-generated text detection using probability curvature, in: International Conference on Machine Learning, PMLR, 2023, pp. 24950–24962.
[5] T. Munyer, A.A. Tanvir, A. Das, X. Zhong, Deeptextmark: a deep learning-driven text watermarking approach for identifying large language model generated text, Ieee Access 12 (2024) 40508–40520.
[6] M. Di Gisi, G. Fenza, M. Gallo, What prompted that? A structured approach to prompt inversion, in: Proceedings of the 18th International Conference on the Quality of Information and Communications Technology (QUATIC 2025), Lecture Notes in Computer Science, Springer, 2025, accepted, to appear.
[7] S. Fariello, G. Fenza, F. Forte, M. Gallo, M. Marotta, Distinguishing human from machine: a review of advances and challenges in AI-generated text detection, Int. J. Interact. Multimed. Artif. Intell. 9 (3) (2025) 6.

[8] X. Zhao, P.V. Ananth, L. Li, Y.-X. Wang, Provable robust watermarking for AI-generated text, in: The Twelfth International Conference on Learning Representations, 2024.
[9] T. Kumarage, H. Liu, Neural authorship attribution: stylometric analysis on large language models, in: 15th International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CYBERC 2023, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 51–54.
[10] Y. Lu, A. Liu, D. Yu, J. Li, I. King, An entropy-based text watermarking detection method, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 11724–11735.
[11] A. Hans, A. Schwarzschild, V. Cherepanova, H. Kazemi, A. Saha, M. Goldblum, J. Geiping, T. Goldstein, Spotting LLMs with binoculars: zero-shot detection of machine-generated text, in: Forty-First International Conference on Machine Learning, 2024.
[12] A. Alshawabkeh, A. Hassooni, F. Kharbat, S. Alouneh, Detecting academic misconduct: a BERT-based approach to identifying ChatGPT-generated content, in: 2024 2nd International Conference on Foundation and Large Language Models (FLLM), IEEE, 2024, pp. 450–455.
[13] J. Su, T.Y. Zhuo, D. Wang, P. Nakov, DetectLLM: leveraging log rank information for zero-shot detection of machine-generated text, in: The 2023 Conference on Empirical Methods in Natural Language Processing, 2023.
[14] X. Guo, Y. He, S. Zhang, T. Zhang, W. Feng, H. Huang, C. Ma, Detective: detecting AI-generated text via multi-level contrastive learning, Adv. Neural Inf. Process. Syst. 37 (2024) 88320–88347.
[15] J. Wu, R. Zhan, D. Wong, S. Yang, X. Yang, Y. Yuan, L. Chao, DetectRL: benchmarking LLM-generated text detection in real-world scenarios, Adv. Neural Inf. Process. Syst. 37 (2024) 100369–100401.
[16] Y. Wang, J. Mansurov, P. Ivanov, J. Su, A. Shelmanov, A. Tsvigun, C. Whitehouse, O.M. Afzal, T. Mahmoud, T. Sasaki, et al., M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection, in: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 1369–1407.
[17] X. Hu, P.-Y. Chen, T.-Y. Ho, Radar: Robust AI-text detection via adversarial learning, Adv. Neural Inf. Process. Syst. 36 (2023) 15077–15095.
[18] G. Mikros, A. Koursaris, D. Bilianos, G. Markopoulos, AI-writing detection using an ensemble of transformers and stylometric features, in: CEUR Workshop Proceedings, vol. 3496, CEUR-WS, 2023.
[19] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, Y. Wu, How close is ChatGPT to human experts? comparison corpus, evaluation, and detection, arXiv preprint arXiv:2301.07597, 2023.
[20] R. Koike, M. Kaneko, N. Okazaki, Outfox: LLM-generated essay detection through in-context learning with adversarially generated examples, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, 2024, pp. 21258–21266.
[21] G. Gritsai, A. Voznyuk, A. Grabovoy, Y. Chekhovich, Are AI detectors good enough? A survey on quality of datasets with machine-generated texts, in: AAAI 2025 Workshop on Preventing and Detecting LLM Misinformation (PDLM), 2025.
[22] Y. Liu, Z. Zhang, W. Zhang, S. Yue, X. Zhao, X. Cheng, Y. Zhang, H. Hu, ArguGPT: evaluating, understanding and identifying argumentative essays generated by GPT models, Comput. Linguist. 1 (1) (2016).
[23] A. Yadagiri, L. Shree, S. Parween, A. Raj, S. Maurya, P. Pakray, Detecting AI-generated text with pre-trained models using linguistic features, in: Proceedings of the 21st International Conference on Natural Language Processing (ICON), 2024, pp. 188–196.
[24] I. Zahid, Y. Chang, T. Madusanka, Y. Sun, R.T. Batista-Navarro, Multi-loss fusion: angular and contrastive integration for machine-generated text detection, in: Findings of the Association for Computational Linguistics: EMNLP 2024, 2024, pp. 7189–7202.
[25] A. Hireche, S. Al-Dabet, M. Mediani, A.N. Belkacem, Detecting AI-generated text: a Bi-GRU with linguistic features approach, in: 2025 IEEE Global Engineering Education Conference (EDUCON), IEEE, 2025, pp. 1–7.
[26] Z. Lai, X. Zhang, S. Chen, Adaptive ensembles of fine-tuned transformers for LLM-generated text detection, in: 2024 International Joint Conference on Neural Networks (IJCNN), IEEE, 2024, pp. 1–7.
[27] M. Abassy, K. Elozeiri, A. Aziz, M. Ta, R. Tomar, B. Adhikari, S. Ahmed, Y. Wang, O.M. Afzal, Z. Xie, et al., LLM-detectaive: a tool for fine-grained machine-generated text detection, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2024, pp. 336–343.
[28] X. Xu, X. Li, T. Wang, J. Tian, Y. Jiang, Team qust at semeval-2024 task 8: a comprehensive study of monolingual and multilingual approaches for detecting ai-generated text, arXiv preprint arXiv:2402.11934, 2024.
[29] Z. Guo, K. Jiao, X. Yao, Y. Wan, H. Li, B. Xu, L. Zhang, Q. Wang, Y. Zhang, Z. Mao, USTC-BUPT at SemEval-2024 task 8: enhancing machine-generated text detection via domain adversarial neural networks and LLM embeddings, in: Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), 2024, pp. 1511–1522.

Author biography

Maria Di Gisi received a Bachelor's Degree in Diplomatic, International and Global Security Studies and a Master's Degree in Data Science and Innovation Management, with a specialization in Cyber Risk Management for Advanced Defense Strategies, from the University of Salerno, Italy, in 2022 and 2024, respectively. She is currently pursuing the National Ph.D. in Cybersecurity at IMT School for Advanced Studies Lucca, Italy. Her research interests focus on the application of Artificial Intelligence to cybersecurity, with particular attention to the development of AI-driven solutions for cyber defense and resilience.

Giuseppe Fenza graduated and received a PhD in Computer Sciences from the University of Salerno, Italy, in 2004 and 2009, respectively. The research activity concerns Computational Intelligence methods to support semantic-enabled solutions and decision-making. He has over 60 publications in Fuzzy Decision Making, Knowledge Extraction and Management, Situation and Context Awareness, Semantic Information Retrieval, Service Oriented Architecture, and Ontology Learning. More recently, he has worked in Automating Open Source INTelligence and Big Data Analytics for counterfeiting extremism and supporting information disorder awareness. He is currently an Associate Professor in Computer Science at the University of Salerno.

Mariacristina Gallo earned a master's degree in computer science at the University of Salerno, Italy, in 2009. In 2021, she obtained a Ph.D. degree in Big Data Management at the same University. Research interests mainly focus on Computational Intelligence methods to support semantic-enabled solutions and decision making. Research activities regard Knowledge Extraction and Management, Context Awareness, Semantic Information Retrieval, Ontology Learning. She is currently a research fellow at the University of Salerno.

Vincenzo Loia graduated in Computer Science at the University of Salerno, Italy, in 1985 and received his Ph.D. in Computer Science in 1989 at the Université Pierre and Marie Curie Paris VI, France. He is currently a Computer Science Full Professor at the University of Salerno, where he served as a researcher from 1989 to 2000 and as an associate professor from 2000 to 2004. Dr. Loia is the Co-Editor-in-Chief of Soft Computing and the Editor-in-Chief of Ambient Intelligence and Humanized Computing. He serves as an Editor for 14 other international journals.